

Shoreline Feature Extraction from Remotely-Sensed Imagery

by

Eduardo Araujo Loos


B.Sc., Universidade do Estado do Rio de Janeiro, 1997

A Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of


MASTER OF SCIENCE


in the Department of Geography

We accept this thesis as conforming
to the required standard


Dr. K. Olaf Niemann, Supervisor (Department of Geography)


Dr. Gail L. Kucera, Departmental Member (Department of Geography)


Dr. John R. Harper, Outside Member (School of Earth and Ocean Sciences)


Dr. Jens H. Jahnke, External Examiner (Department of Computer Science)

© Eduardo Araujo Loos, 2002

University of Victoria

All rights reserved. This thesis may not be reproduced in whole or in part, by
photocopying or other means, without the permission of the author.

Supervisor: Dr. K. Olaf Niemann

ABSTRACT

Different methods of delineation were developed and used to extract shoreline features from images with different spatial and spectral resolutions acquired by airborne and spaceborne sensors. The exact location of the shoreline is difficult to obtain from the images and therefore the definition of shoreline was based upon the geomorphological and oceanographical characteristics of the area of study. Extracted shoreline vectors were then compared to existing official shoreline vectors to assess their accuracy and software efficiency. The generation of shoreline vectors with a high accuracy will greatly improve the time of work and number of specialised personnel, and allow for the integration of the resulting shoreline vectors into cartographic databases.

Keywords: shoreline delineation, edge detection, feature extraction.

Examiners:

Dr. K. Olaf Niemann, Supervisor (Department of Geography)

Dr. Gail L. Kucera, Departmental Member (Department of Geography)

Dr. John R. Harper, Outside Member (School of Earth and Ocean Sciences)

Dr. Jens H. Jahnke, External Examiner (Department of Computer Science)

TABLE OF CONTENTS

Abstract	ii
Table of Contents	iii
List of Tables	vi
List of Figures	vii
List of Equations and Filter Templates	x
Acknowledgements	xii
Dedication	xv
1. INTRODUCTION	1
1.1 Goals and Objectives	2
1.2 Overview	6
2. BACKGROUND INFORMATION	8
2.1 Types of Coast	8
2.2 Historical Importance of Remote Sensing Studies along the Coastal Zone	10
2.3 Importance of Remote Sensing for Coastal Studies	12
2.4 Shoreline Determination	15
2.5 Difficulties Encountered in Shoreline Determination	16
2.6 Historical Importance of Boundary Tracing Research	19
2.6.1 Edge Detection in General Image Analysis Research	19
2.6.2 Road Detection	22
2.6.3 Boundary Delineation in Medical Research	24
2.6.4 Delineation of Coastline Contours	25
2.7 Digital Image Processing Methods Used in Feature Delineation and Extraction ..	28
2.7.1 Preprocessing Operations	29
2.7.1.1 Radiometric Calibration	29
2.7.1.2 Geometric Restoration	30
2.7.2 Geometric Correction	31
2.7.3 Image Enhancement	31

2.7.3.1 Grey-Level Thresholding.....	32
2.7.3.2 Density Slicing.....	33
2.7.3.3 Contrast Stretching.....	33
2.7.4 Filters/Spatial Feature Manipulation.....	36
2.7.4.1 Laplacian Filter	37
2.7.4.2 Sobel Filter.....	38
2.7.4.3 Roberts Filter	40
2.7.4.4 Prewitt Filter	41
2.7.4.5 Pixel Difference Filters	41
2.7.4.6 Frei-Chen Filter.....	43
2.7.4.7 Smoothing Filter	44
2.7.4.8 Median Filters	47
2.7.4.9 Canny Filter	49
2.7.5 Spatial Feature Extraction/Line Tracing.....	51
3. THE SEARCH FOR THE HOLY ALGORITHM	52
3.1 Methodology.....	52
3.2 Guidelines for Shoreline Determination.....	52
3.3 Area of Study.....	57
3.4 Datasets	58
3.4.1 Images	58
3.4.2 Vectors	59
3.5 Digital Image Processing.....	60
3.5.1 Preprocessing Operations.....	61
3.5.2 Geometric Correction.....	62
3.5.2.1 Transformation of Pixel Coordinates.....	62
3.5.2.2 Resampling	65
3.5.3 Regarding Spatial Resolution or Which Sensors Should Be Used? ..	66
3.5.4 Regarding Spectral Resolution or Which Bands Should Be Used?..	68
3.5.5 Enhancements and Filters	69
3.5.6 Feature Image.....	69

3.5.7 Shoreline Tracing.....	71
4. RESULTS	76
4.1 Control Images	76
4.2 Shoreline Feature Extraction	79
4.2.1 Low-Resolution Images	79
4.2.1.1 Coasts with Vegetation Line.....	80
4.2.1.2 Sandy Beaches	88
4.2.1.3 Mudflats/Shallow Areas.....	93
4.2.1.4 Rocky Coasts	99
4.2.2 High-Resolution Images	106
4.2.2.1 Coasts with Vegetation Line.....	108
4.2.2.2 Sandy Beaches	112
4.2.2.3 Mudflats/Shallow Areas.....	116
4.2.2.4 Rocky Coasts	119
5. DISCUSSION	123
5.1 Digital Image Processing Techniques	123
5.2 Low-Resolution Images.....	125
5.3 High-Resolution Images	126
5.4 Overall Considerations	127
6. CONCLUSION	131
6.1 Overall Assessment	131
6.2 Final Remarks.....	133
7. REFERENCES	135
APPENDIX A	142
APPENDIX B	152
APPENDIX C	156
APPENDIX D	157

LIST OF TABLES

Table 1 - Shoreline by Province and Territory (1972) (in kilometres)	4
Table 2 - Comparison between TMR and GMR methods	26
Table 3 - Acquisition Date and Resolutions of Imagery Used	60
Table 4 - Node Expansion and Final Node Costs of the Search in Figure 3.7	75
Table 5 - Performance of the Tracing Algorithm on Control Images.....	77
Table 6 - Characteristics of Control Images	78
Table 7 - FeIm Rationale for Low-resolution Images.....	80
Table 8 - FeIm Rationale for High-resolution Images	106

LIST OF FIGURES

Figure 2.1 - Edge Types.....	20
Figure 2.2 - (a) Satellite Imagery with low contrast; (b) GLV histogram	34
Figure 2.3 - (a) Satellite Imagery with linear enhancement; (b) GLV histogram.....	35
Figure 2.4 - (a) Satellite Imagery with equalisation enhancement; (b) GLV histogram ...	36
Figure 3.1 - Water Levels adopted by the Canadian Hydrographic Service.....	53
Figure 3.2 - Clayoquot Sound.....	58
Figure 3.3 - Geometric correction and nearest neighbour resampling.....	63
Figure 3.4 - Influence of polynomial order on the distortion of geocoded images.....	64
Figure 3.5 - Resolution comparison.....	67
Figure 3.6 - Edge Element Expansion Rationale.....	73
Figure 3.7 - Martelli search algorithm on a subset of an image dataset	74
Figure 4.1 - Control Images	76
Figure 4.2 - Enlarged control image	78
Figure 4.3 - SPOT 3 (10 m). Coasts with vegetation line. Extracted shoreline.....	81
Figure 4.4 - Landsat 7 (15 m). Coasts with vegetation line. Extracted shoreline	82
Figure 4.5 - Spot 1 (20 m). Coasts with vegetation line. Extracted shoreline	82
Figure 4.6 - Landsat 7 (30 m). Coasts with vegetation line. Extracted shoreline	83
Figure 4.7 - Landsat 7 (15 m). Coasts with vegetation line. Vector comparison	84
Figure 4.8 - SPOT 1 (20 m). Coasts with vegetation line. Vector comparison	85
Figure 4.9 - Landsat 7 (30 m). Coasts with vegetation line. Vector comparison	86
Figure 4.10 - Shoreline GLV profiles for coasts with vegetation line.....	87
Figure 4.11 - SPOT 3 (10 m). Sandy beaches. Extracted shoreline.....	88
Figure 4.12 - Landsat 7 (15 m). Sandy beaches. Extracted shoreline.....	88
Figure 4.13 - Spot 1 (20 m). Sandy beaches. Extracted shoreline	89
Figure 4.14 - Landsat 7 (30 m). Sandy beaches. Extracted shoreline.....	89
Figure 4.15 - Landsat 7 (15 m). Sandy beaches. Vector comparison	90

Figure 4.16 - SPOT 1 (20 m). Sandy beaches. Vector comparison	90
Figure 4.17 - Landsat 7 (30 m). Sandy beaches. Vector comparison	91
Figure 4.18 - Shoreline GLV profiles for sandy beaches	92
Figure 4.19 - SPOT 3 (10 m). Mudflats/shallow areas. Extracted shoreline	93
Figure 4.20 - Landsat 7 (15 m). Mudflats/shallow areas. Extracted shoreline	93
Figure 4.21 - Spot 1 (20 m). Mudflats/shallow areas. Extracted shoreline.....	94
Figure 4.22 - Landsat 7 (30 m). Mudflats/shallow areas. Extracted shoreline	94
Figure 4.23 - SPOT 3 (10 m). Mudflats/shallow areas. Vector comparison.....	95
Figure 4.24 - Landsat 7 (15 m). Mudflats/shallow areas. Vector comparison.....	95
Figure 4.25 - SPOT 1 (20 m). Mudflats/shallow areas. Vector comparison.....	96
Figure 4.26 - Landsat 7 (30 m). Mudflats/shallow areas. Vector comparison.....	97
Figure 4.27 - Shoreline GLV profiles for mudflats/shallow areas.....	98
Figure 4.28 - SPOT 3 (10 m). Rocky coasts. Extracted shoreline	99
Figure 4.29 - Landsat 7 (15 m). Rocky coasts. Extracted shoreline	99
Figure 4.30 - Spot 1 (20 m). Rocky coasts. Extracted shoreline	99
Figure 4.31 - Landsat 7 (30 m). Rocky coasts. Extracted shoreline	100
Figure 4.32 - SPOT 3 (10 m). Rocky coasts. Vector comparison.....	101
Figure 4.33 - Landsat 7 (15 m). Rocky coasts. Vector comparison.....	102
Figure 4.34 - SPOT 1 (20 m). Rocky coasts. Vector comparison.....	103
Figure 4.35 - Landsat 7 (30 m). Rocky coasts. Vector comparison.....	104
Figure 4.36 - Shoreline GLV profiles for rocky coasts.	105
Figure 4.37 - Aerial orthophoto. Example of shadows caused by trees.....	107
Figure 4.38 - KFA-1000. Coasts with vegetation line. Extracted shoreline	108
Figure 4.39 - KFA-1000. Coasts with vegetation line. Vector comparison	109
Figure 4.40 - KFA-1000. Shoreline GLV profile for coast with vegetation line.....	110
Figure 4.41 - Aerial orthophoto. Coasts with vegetation line. Extracted shoreline.....	111
Figure 4.42 - Aerial orthophoto. Coasts with vegetation line. Vector comparison	111
Figure 4.43 - Aerial orthophoto. Shoreline GLV profile for coast with vegetation line .	112
Figure 4.44 - KFA-1000. Sandy beaches. Extracted shoreline.....	113
Figure 4.45 - KFA-1000. Sandy beaches. Vector comparison	113
Figure 4.46 - KFA-1000. Shoreline GLV profile for sandy beaches.....	114

Figure 4.47 - Aerial orthophoto. Sandy beaches. Vector comparison	115
Figure 4.48 - Aerial orthophoto. Shoreline GLV profile for sandy beaches	115
Figure 4.49 - KFA-1000. Mudflats/shallow areas. Extracted shoreline	116
Figure 4.50 - KFA-1000. Mudflats/shallow areas. Vector comparison.....	117
Figure 4.51 - KFA-1000. Shoreline GLV profiles for mudflats/shallow areas.	117
Figure 4.52 - Aerial orthophoto. Mudflats/shallow areas. Vector comparison.....	118
Figure 4.53 - Aerial orthophoto. Shoreline GLV profile for mudflats/shallow areas.....	119
Figure 4.54 - KFA-1000. Rocky coasts. Vector comparison.....	120
Figure 4.55 - KFA-1000. Shoreline GLV profiles for rocky coasts.	121
Figure 4.56 - Aerial orthophoto. Rocky coasts. Vector comparison.....	122
Figure 4.57 - Aerial orthophoto. Shoreline GLV profile for rocky coasts.....	122

LIST OF EQUATIONS AND FILTER TEMPLATES

Equation 2.1 - Linear stretch.....	34
Equation 2.2 - Laplacian operator. Continuous function.....	37
Equation 2.3 - Laplacian operator. Discrete approximation.....	37
Equation 2.4a - Laplacian operator. Gradient of X.....	37
Equation 2.4b - Laplacian operator. Gradient of Y.....	37
Equation 2.5 - Laplacian operator. Short discrete approximation.....	38
Template 2.6 - 3x3 pixel coordinates window.....	38
Template 2.7 - 3x3 Laplacian kernel.....	38
Template 2.8 - 3x3 image sample window.....	39
Equation 2.9 - Sobel operator.....	39
Equation 2.10a - Sobel operator. Gradient of X.....	39
Equation 2.10b - Sobel operator. Gradient of Y.....	39
Template 2.11a - Sobel operator convolution window in X.....	39
Template 2.11b - Sobel operator convolution window in Y.....	39
Template 2.12 - 2x2 image sample window.....	40
Equation 2.13 - Roberts operator. Short discrete approximation.....	40
Equation 2.14a - Roberts operator. Gradient of X.....	40
Equation 2.14b - Roberts operator. Gradient of Y.....	40
Template 2.15a - Roberts operator convolution window in X.....	40
Template 2.15b - Roberts operator convolution window in Y.....	41
Template 2.16 - Prewitt operator convolution window in X.....	41
Template 2.17 - Prewitt operator convolution window in Y.....	41
Template 2.18 - Pixel Difference operator convolution window in X.....	42
Template 2.19 - Pixel Difference operator convolution window in Y.....	42
Template 2.20 - Separated Pixel Difference operator convolution window in X.....	42
Template 2.21 - Separated Pixel Difference operator convolution window in Y.....	42
Template 2.22 - Frei-Chen operator convolution window in Y.....	43

Template 2.23 - Frei-Chen operator convolution window in Y.....	43
Template 2.24 - Smoothing filter A. Average	45
Template 2.25 - Smoothing filter B.	45
Template 2.26 - Smoothing filter C.	45
Template 2.27 - Smoothing filter D.	45
Template 2.28 - Smoothing filter E.	46
Template 2.29 - Smoothing filter F.....	46
Template 2.30 - Smoothing filter G.	46
Template 2.31 - Median filter. MedianCross3x3.....	47
Template 2.32 - Median filter. MedianSquare3x3.....	48
Template 2.33 - Median filter. MedianSquare5x5.....	48
Template 2.34 - Median filter. Octagon5x5.....	48
Template 2.35 - Median filter. MedianCross7x7.....	48
Template 2.36 - Median filter. Diamond7x7.	49
Equation 2.37 - Gaussian distribution.....	49
Template 2.38 - Gaussian filter. Gaussian3x3	49
Template 2.39 - Gaussian filter. Gaussian7x7	50
Equation 2.40 - Laplacian of Gaussian (LoG)	50
Equation 3.1 - Second-order polynomial transformation in X.....	64
Equation 3.2 - Second-order polynomial transformation in Y.....	64
Equation 3.3 - Negate Function	70

ACKNOWLEDGEMENTS

My journey to the Great White North became possible thanks to the internet. At night, after work, I would spend hours in front of my computer searching for universities with a remote sensing/geomatics department. It was a long and tough process, from contacting potential supervisors by e-mail to gathering all the required paperwork and official documents. One and a half long year in total. And now, four years after, I have just finished the most amazing adventure of my life so far thanks to the help and advice from those I consider my truest friends.

Few people know how much I love remote sensing, image processing and all the related innovative technologies that I work with every day. My supervisor, Dr. K. Olaf Niemann, offered me a unique opportunity for which I am deeply grateful. Thank you Olaf!!!

Thanks to Dr. Gail L. Kucera and Dr. John R. Harper for having accepted my invitation to evaluate my research. Your support has been inspiring.

Thanks to Mr. Don E. Howes (BC Land Use Coordination Office) for kindly providing some of the images analysed in this study.

Thanks to the University of Victoria and the GEOIDE Network for providing funding for my research.

My closest friends have created an incredible atmosphere in the last three years, so upbeat and full of surprises that I can only wish for more! Thank you Trisalyn Nelson and Travis Ferbey, Michelle Bennett, Tandy Shepard, Manami Kaku, Kim Chambers, Debbie Jenkins and Stuart Walsh for all the fun, respect and friendship! Thanks also to my friend Dr. Maycira Costa (University of Victoria) for all the remote sensing knowledge shared and friendship! The Ferbey family accepted me as a son and I am deeply honoured to be part of it; thank you Patti and Dennis, Stasia and Ike, Grandma Doris (thanks for letting me peel the apples for the apple pies!!) and Grandma Stella, and Fred for all the fun and love!

Each person in the Remote Sensing Lab contributed to my personal and professional development and I am sincerely thankful to: Craig Dillabaugh for helping me understand C++; Rick Sykes, for the endless computer/network problem-solving; Ian O'Connell for all my GIS-related questions; Jason Campbell for Java-related clarifications; and Dr. Joji Iisaka for advice on image processing techniques.

The Department of Geography accepted me with arms wide open and I must say that I could not have made it without you: Kathie Merriam, Jill Jahansoozi, Darlene Li and Elaine Cornwell for helping me understand all the ins and outs of the UVic structure; Dr. C. Peter Keller, Diana Hocking, John Newcomb and Phil Wakefield for guiding me in the process of teaching and becoming a better instructor; Diane Braithwaite, Ole Heggen and Ken Josephson for all the creative advice in creating conference posters and assistance in organising my teaching presentations.

Thanks to Mr. Charles O'Reilly (Canadian Hydrographic Service), whose exceptional knowledge on shoreline positioning was crucial to my better understanding of the problem.

Thanks to Dr. John M. Gauch (University of Kansas) and Dr. Douglas A. Lyon (Fairfield University) for answering questions I had regarding their image processing applications.

Thanks also to Dr. Braz Sanchez (Universidade Federal Fluminense) and Dr. Ulisses L. Gomes (Universidade do Estado do Rio de Janeiro) for guiding me in the right direction.

Thanks to everyone at Threetek Geomatics Solutions for contributing to my professional development, especially João C. Vassallo for having confidence in my work.

Life in Canada would not have been as cheerful (even though I can't seem to win when playing Hearts!!) and healthy (very organic!!) without the love and respect from Jane Mackey and David Maxwell. Thank you for letting me watch Formula 1 races every other week at 5 am on Sundays and, above of all, for being my family here!

My mom Consuelo and my brothers Rudi and Rafa are my daily inspiration and give me the strength I need to overcome the 70 degrees of latitude that keep us apart. I also thank all my relatives in Brazil for their constant support and love.

My love Sarah Maxwell has always been by my side and has patiently proof-read my thesis. She has given me all the help, encouragement and love I need to become a better person. Our lives will never be the same. I love you, Sweetie!

Eddie Loos

June 2002

Victoria, BC

DEDICATION

This thesis is dedicated to Dr. Sergio, who has always been looking after me and whose help, encouragement, guidance and love will always be in my heart.

1. Introduction

The coastal zone can be simply described as the area where land and sea converge. Despite covering a relatively small area on global scale, this zone is the most widespread boundary on the planet, extending through all continents (Gross, 1972).

The settlement of coastal environments has long been of vital importance to the development of communities world-wide. Ancient civilisations of the Eastern Mediterranean, such as the Greeks, Romans and Egyptians, as well as Scandinavian civilisations, such as the Vikings, constructed their settlements near the coast. The reason for this behaviour was the proximity of water, namely rivers and oceans, enabling transportation and access to major sources of food. In the early ages, the aesthetics of the coastal zone were not considered as relevant as nowadays, when the coastal zone is desired by most people for family holidays and many other recreational purposes, including driving along coastal highways just for admiring the beauty of the scenery.

Today, this pattern of settlement along the coasts remains, with more than one half of the world's population living within 60 kilometres of the coastline (UN, 1992). With regards to military strategy, the coast represents an important feature to be protected and also to be invaded. For example, not long ago the Allied troops were landing on the beaches of Normandy during World War II (Davis, 1996).

There has been a significant number of scientific publications and political decisions that pertain to coastlines, not only because of environmental importance but

also, quite often, because of economical importance (Shallowitz, 1964). Environmental groups and developers seldom reach an agreement when it comes to the use of the coastal environment. From the ecological (and holistic) point of view, the broad range of biological, geological, physical and chemical environments along the coastline represents a dynamic, and at the same time fragile, response to natural as well as human activities. Most argue that more research on the stability of coastal environments should be conducted in order to prevent their degradation and future expenditure on restoration.

The observation of changes along the coasts using aerial photography began in the early 1930's and since then methods for studying these variations have substantially improved through the analysis of data acquired by satellites (Shepard *et al.*, 1971). The use of satellites has enabled the acquisition of imagery for virtually the entire globe. Updating maps and charts by using remotely-sensed data is faster and cheaper than using conventional *in-situ* methods such as ground-truthing and bathymetric soundings. For the last two decades, analysts and researchers have been using remotely-sensed data to look for changes along shorelines and to extract information from these datasets directly onto maps. Recent advances in computer programming languages have facilitated the creation of image processing software that can handle vast amounts of information quickly (PCI, 1997).

1.1 Goals and Objectives

In this project, the goal has been to develop a methodology for the delineation and extraction of shorelines from images acquired by remote sensors, such as those aboard

aircrafts and satellites, as well as the creation of shoreline vectors from those images. Original digital imagery was enhanced for better visualisation of distinct features and was later filtered for the delineation of individual edges. These vectors were then compared with existing shoreline vectors used by coastal authorities in order to check their accuracy.

The development of edge tracing algorithms proved difficult and sometimes yielded unsatisfactory results. Digital images can be manipulated in so many different ways that one must focus on a certain method and try to come up with the best methodology for extracting the information desired. The improvement of these procedures leads to the automated or semi-automated extraction of features. The use of computers certainly made it possible to gather information from the imagery to generate new shoreline vectors, and to alter the existing out-of-date maps. But great care must be taken when deciding what to consider as part of the shoreline. This decision is not always easy, for there are many different coastal features with rapidly-changing patterns that make the job even more difficult.

The development of edge/shoreline detection and extraction software is extremely detailed and requires that many different decisions be made regarding the identification of shoreline features. For casual, non-scientific daily purposes shoreline and coastline are considered to be the same, little being put into the real distinction between them. According to the Canadian Hydrographic Service (CHS, 1999), coastline (general term) is a general line of the coast, measured as running directly across bays or inlets, whereas shoreline is the perimeter of the land along the edge of the water, measured as accurately as possible for a particular location. Canada's shoreline is the longest of any country in

the world, with British Columbia's shoreline being only smaller than those of the Northwest Territories, Québec, and Newfoundland and Labrador (Table 1). Consequently, shoreline is usually longer for a particular location than is its coastline. Both coastline and shoreline are considered as part of the coastal zone (or shore), a broad zone encompassing water and land.

Table 1
Shoreline by Province and Territory (1972) (in kilometres)
 * Main island only. ** Includes perimeter of Cape Breton Island. - Data unavailable.
 (Modified from Sebert *et al.*, 1972)

Province/Territory	Mainland Coast	Major Islands Perimeter	Minor Islands Perimeter	Total of Province or Territory
Newfoundland and Labrador	8,172	11,548*	9,236	28,956
Prince Edward Island	-	1,107*	153	1,260
Nova Scotia	4,051	1,883**	1,645	7,579
New Brunswick	1,524	177	568	2,269
Québec	10,389	554	2,380	13,323
Ontario	1,210	-	-	1,210
Manitoba	917	-	-	917
Saskatchewan	-	-	-	-
Alberta	-	-	-	-
British Columbia	7,022	10,835	7,868	25,725
Yukon Territory	343	-	-	343
Northwest Territories (and Nunavut)	24,131	86,818	50,511	161,460
Totals	57,759	112,922	72,361	243,042

These definitions look fairly straightforward in an ideal environment but the real world presents situations that are far more complicated. The incredible diversity of natural environments around the globe creates infinite possibilities where the accurate determination of the shoreline becomes a problem. Overriding these problems requires the adoption of certain criteria that later become standard for the region of interest. However, determining these criteria depends not only on oceanographical and geomorphological characteristics of the coastal zone but also on political and legal, social and environmental regulations.

The geomorphology of the shore is directly dependent on oceanographic factors given that the shore is under permanent action of waves and tides. The shoreline is therefore constantly migrating according to the breaking and splashing of the waves, and the periodic movements of the tides during day and night. Authorities responsible for surveying and mapping the shore in many countries adopt different baselines for determining the position of the shoreline and it becomes a real problem for countries like Canada and the United States. Both countries adopt different shoreline definitions on their nautical charts and it has been agreed that on charts covering both Canadian and U.S. waters, the Canadian shoreline datum (higher high-water large tide) is to be used on the Canadian side of the boundary, and U.S. shoreline datum (mean high-water, somewhat lower than the Canadian shoreline) on the U.S. side, regardless of which country publishes the chart (Forrester, 1983; CHS, 1999; NOS, 1999).

Obviously, confusion can result from this profusion of shoreline definitions and can contribute to ill-informed decision making that may in turn have critical or even catastrophic consequences.

In short, the proposed methodology will:

1. Require minimal manipulation by the user;
2. Be faster than currently used manual tracing of shorelines; and
3. Yield accurate shoreline vectors.

1.2 Overview

This thesis has been laid out in an ascending level of specialisation, with the first chapters giving an overall view of the subject and the closing chapters dealing with case problems.

Chapter 2 presents a brief description of the different types of coasts, related social, economic and environmental aspects, historical background and importance of coastline studies using remote sensing methods, difficulties in determining the shoreline accurately and existing image processing methods used in this field.

Chapter 3 describes the methodology adopted in this research, the area of study and the imagery used. Final results are presented in Chapter 4, as well as the various attempts that led to choosing the most efficient shoreline delineation procedures.

Chapter 5 compares the different methods, analyses their outputs, and provides support for the best delineation methods. Finally, Chapter 6 provides a summary of the advantages and disadvantages of each of the methods that were evaluated and tries to give some insight on what could be done in order to maximise shoreline extraction in the future.

2. Background Information

2.1 Types of Coasts

Major coastal systems can be classified as: *gently-sloping sand beaches, rocky coasts, estuaries, deltas, fjords, barrier islands* and *tidal/mud flats*. These coastal classes can be further classified as *emergent* or *submergent*, as *erosional* or *depositional*, and as *marine* or *non-marine* depending on the physical and geological processes responsible for their formation. Waves, tides and currents comprise physical processes that shape and sometimes are strong enough to alter the shoreline (Clark, 1994; Laing, 1995).

Emergent coastlines develop mainly due to uplifting of the continental margin caused by tectonic displacements or the retreat of sea level due to glaciation. On the other hand, submergent coastlines are partially drowned when continental margins sink or when sea level rises. Most of the Pacific coast of North America is emergent due to action of plate tectonics along the coast, while most of the Atlantic coast is submergent (passive continental margins), characterised by sandy beaches and estuaries. Fjord coasts are also submergent coastlines and they occur where glaciers have deepened coastal valleys. Fjords can be extensively found along high-latitude coasts (Strahler *et al.*, 1978).

The constant remodelling of coastlines occurs through both marine and continental (non-marine) processes. For example, the influence of the discharge of rivers into coastal waters creates many different features along the coast, such as deltas and estuaries. These environments can be either erosional or depositional, depending on the

sediment input rate into the system. The formation of deltas only occurs if the quantity of sediments transported by the rivers is high enough to accumulate at the river mouth. This accumulation also depends on the ability of marine processes to rework and remove the sediments. Estuaries are also drowned river mouths but the river sediment influx is lower than that of deltas, thus allowing tides, longshore currents and waves to carry away sediments (Thurman, 1994).

Open coasts, such as sandy beaches, are usually depositional environments. However, storm waves have a strong erosive power to remove sediments. Longshore currents are responsible for transporting these sediments along the shore. Depending on sediment size and current velocity, *spits* (offshore sandbars attached or not at one end to a headland) may start to develop. These elongated spits may become later barrier islands which, due to the continuous sediment transportation by longshore currents and weak wave attack, may eventually enclose a lagoon (Getis *et al.*, 1991).

Organisms are also responsible for modifying shorelines, either by colonising (marsh and sea grasses, mangroves, oysters and tubeworms) or by fostering deep changes on the littoral morphology (moving or fixing sediment, or breaking rocks). These organisms occupy different locations ranging from infratidal to supratidal environments. Coral reefs are the dominant modifying force along some shores, such as the Great Barrier Reef in Australia that covers several hundreds of kilometres.

Chemical processes also play a role in the constant modification of the shoreline. They are usually classified as *destructive* or *constructive*. The destruction of rocks and sediment particles occurs due to chemical weathering, whereas construction takes place

through lithification of sediments into rocks. Both processes are extremely slow and are often concentrated in relatively small areas. Precipitation is the most common constructive process. The precipitation of calcium carbonate forms extensive beachrocks along some coasts, which are visible on remote sensing images (Davis, 1996).

2.2 Historical Importance of Remote Sensing Studies along the Coastal Zone

The use of remote sensing techniques in coastal studies is intimately connected to advances in aeronautics and aerial photography in the last eighty years. After the launch of the first TIROS satellite in 1960, a new era of digital imagery acquisition began, revolutionising and expanding remote sensing capabilities. Larger areas could now be covered and portions of the electromagnetic spectrum beyond the range of photography could be used to interpret features visible on the images (Chen, 1985; Curran, 1985).

According to El-Ashry (1977), coastal photographs were first taken in Germany in 1928 and in the United States in 1930 for charting purposes. During World War II, the U.S. Military acquired coastal photographs for determining the characteristics and the viability of the coasts for landing troops and vehicles. Further research was conducted by the Naval Photo Interpretation Center to determine water depth along enemy-held shorelines.

Lucke (1934) used aerial photographs and old charts to analyse the changes of the Barnegat Inlet, a tidal inlet in New Jersey. The comparison allowed him to verify that active deposition of sediments was occurring due to tidal currents.

Howard (1939) was the first to use aerial photographs to evaluate the destructive effects of hurricanes on coastlines. The modification of the barrier island off Long Island, New York, due to the striking of huge storm waves, was intense enough to create several inlets along the bar. These new inlets remained open because of tidal currents.

Dietz (1947) pointed out the usefulness of aerial photography in the study of shore features and related processes due to its ability to record constant changes. Furthermore, he emphasised the importance of taking periodical photographs at intervals as short as possible in order to keep track of changes.

In the 1970's, Magoon (1973) discussed the promising development of artificial satellites, showing the possibility that they could be used to identify, delineate and track changes in shoreline configuration.

Numerous studies have shown that the use of aerial photographs and spaceborne imagery for tracking changes is invaluable. Cameron (1965) analysed long-term coastal changes in Nova Scotia using aerial photographs ranging from 2 to 20 years apart. He demonstrated that employing existing photographs and supplementary rephotography would aid in the evaluation of coastal changes. The cost of this method would be justified considering the benefits of a more accurate and rapid examination of the coast. El-Ashry *et al.* (1967), following the aforementioned idea, studied storm effects and long-term trends of shorelines. They also came to the conclusion that sequential aerial photography

is an excellent tool for studying the morphology and changes of coastal features, providing an accurate approximation of net changes in dimension and shape between acquisition dates.

Recent studies, such as Furmanczyk *et al.* (1995), also made use of existing aerial photographs and SPOT panchromatic images for studying coastal morphology in Poland.

2.3 Importance of Remote Sensing for Coastal Studies

Today, the acquisition of airborne and spaceborne images is indispensable in the study of mapping of shoreline features and shoreline changes. It also fosters development in coastal engineering and coastal zone management processes.

The acquisition of remotely-sensed imagery relies basically on four different sensors: passive optical, active optical (Lidar), passive microwave and active microwave (Radar). The spectral sensitivity of the most commonly used optical sensors ranges from 400 nm (blue) to 12500 nm (thermal infra-red). Microwave sensors utilise much longer wavelengths, namely the microwave portion of the electromagnetic spectrum (1 mm to 1 m). Imagery provided by any of these sensors can have a spatial resolution fine enough (less than 1 m) to map shorelines accurately.

The use of remotely-sensed data for coastal mapping is growing fast. Governmental agencies concerned about the environment and national security are gradually employing more remotely-sensed data in their budgets for mapping purposes. A program called *Scanning Hydrographic Operational Airborne Lidar Survey System*

(SHOALS) is conducted by the U.S. Army to perform hydrographic and topographic surveys, including shoreline delineation for strategic planning (SHOALS, 2001). Radar is also used for shoreline delineation in areas that are constantly covered by clouds making optical imagery acquisition impossible (RSI, 2002).

Maps and charts need to be updated frequently, a process that was time-consuming and expensive before the use of remote sensing techniques. These costs and the need for specialised personnel have been significantly reduced. Consequently, maps and charts can be updated more frequently, thus providing more reliable information.

Different regions exhibit different patterns of coasts. Sandy beaches and salt marshes occur in long streaks in Florida; fjords are dominant, glacially-formed coastal features found in Norway; river deltas and estuaries are spread over all the continents and some are surrounded by alluvial plains; coral reefs can be found mainly around the Tropics; barrier islands and lagoons form the main coastal features of the Atlantic coast of North America. These examples of coastal types certainly do not illustrate the real diversity and distribution throughout the world but rather serve to give a little insight into the complexity and magnitude of accurately classifying and mapping these features using remote sensing techniques.

Remotely-acquired images have become essential for the analysis and mapping of landforms. Their use dates back to the late 1800's when aerial cameras were flown in balloons and later (in the early 1900's and during World Wars I and II) in airplanes for military strategy and reconnaissance tactics. Aerial photography has been used to delineate the US shoreline accurately since 1936 by the National Ocean Service (NOS)

(administered by the National Oceanic and Atmospheric Administration (NOAA)), replacing field surveys because they could be completed faster and less expensively. For these reasons, aerial photographs became the only source for coastal mapping in 1980 (NOAA, 2001). The shoreline mapping program is so important that in the early 1980's special purpose mapping cameras (Large Format Cameras) were built to operate at orbital altitudes (Lillesand *et al.*, 1994).

The use of film photography showed many limitations that consequently fostered the development of electronic imaging systems. The imagery acquired by these electronic instruments can be more easily manipulated using digital image processing software and hardware. This not only allows an infinite number of analyses but also reduces costs and optimises the processing/analysis time.

Once acquired, digital images are interpreted for a specific purpose by a human analyst or by a computer. This interpretation is a systematic study of pattern elements in the images, which consists of evaluating and establishing the relationship between features and to the whole environment (Mollard, 1960). Using image processing, the characteristics of main ground features require extensive analysis because of their own subtleties. It would be even harder to distinguish and separate features of interest from the ground, since many of the diagnostic features used on remotely-sensed imagery are not available for ground surveys.

2.4 Shoreline Determination

The determination of shorelines from remotely-sensed images is not an easy task due to the gradation between land and sea, and the change of forms, which can make separating and tracing features very challenging. In most cases, a thorough knowledge of the history of the terrain is most useful. Certainly, boundaries would be best described by checking them in the field, but information can be extracted much faster and with less expense using remote sensing techniques. Therefore, developing a reliable methodology for extracting accurate information from images for the delineation of shorelines is important.

Moffitt (1969) addressed the problem of fixing the position of a shoreline based on photographs of the same area but taken at different dates. Today's image analysis software packages are capable of geographically correlating different scenes described by Moffitt, but one important aspect not taken into account by these automated procedures is the change in position of the shoreline. Corrections must then be made to some standard, such as mean sea level for instance, in order to indicate the most probable location of the shoreline.

MacDonald *et al.* (1971) illustrated the use of radar for imaging shorelines. The great advantage of radar imagery concerning coastal geomorphology relies on the strong contrast between the radiometric responses of land and water, which makes the shorelines appear quite sharp. Other advantages of radar over aerial photography are its ability to acquire images in cloudy environments, such as those frequently associated with coastal regions, and its 24-hour imaging capability (independent of solar illumination).

Depending on the characteristics of the imagery, such as spectral, spatial, temporal and radiometric resolutions¹, differentiation of distinct water levels may not be possible. The lack of tidal gauges along lagoons and estuaries providing continuous data is also a major problem when trying to establish shorelines. This problem is usually not a factor when images are being interpreted for small-scale maps that do not offer great precision (Gierloff-Emden, 1977).

2.5 Difficulties Encountered in Shoreline Determination

Determining the exact shoreline directly from the observations of remotely-sensed images can be very inaccurate. The outlining of shoreline boundaries is a procedure that leads to dubious conclusions, and the elimination or minimisation of errors depends on the imagery being used (Bandat, 1962). One important factor for minimising errors is the resolution of the image. Coarse spatial resolution will not allow accurate determination of the shoreline because pixel size can be large enough to represent the entire tidal range for small-amplitude coasts. The use of computer software for delineation purposes will benefit from high radiometric resolution. In such cases, computer algorithms will have a wider range of values to evaluate slight differences among several distinct features with relatively close digital numbers.

¹ **Types of resolution:**

1. Radiometric: ability to discriminate very slight energy differences; amount of digital precision; data depth (8, 16, 32-bit images).
2. Spatial: ability to discern objects on the ground; pixel size.
3. Spectral: ability to discriminate fine spectral differences; wavelength range.
4. Temporal: time frame represented by the image acquisition.

The spectral resolution of the images is also a factor. In most cases one single feature has different spectral responses, thus using several spectral bands will certainly enhance the accuracy of determination.

Another factor is the sequential availability of images, which provides a more complete coverage of the dynamics of the shore. Comparison between time-lagged images yields information on tide level, wave patterns and even on human impact. Putting all of this information together will certainly aid in the detection of the shoreline

The aforementioned data conditions are ideal for determining an accurate shoreline, but unfortunately they are usually not available. Consequently, studies are conducted using the available data and the data must be manipulated to extract information. This often leads to the development of new, improved strategies for extracting information.

Some coastal zones do not present the gently-sloping sand beach or rocky coast profile commonly associated with these regions, thus presenting a challenge to the shoreline extraction. Mangrove and salt marshes have a large variety of vegetation types, making the geomorphological features appear in greater detail, therefore being more complicated to delineate. The usual procedure adopted world-wide by nautical chart publishers is to draw an apparent shoreline on the outer limits of the vegetation so it appears as the *true* shoreline to navigators (NOAA, 2001).

Another factor that makes the delineation of shorelines difficult is human construction along the coast. Only recently has the media begun to pay more attention to the effects of human activities on the coastal zone. People are starting to realise that the

combination of such fragile environments and dense population produces many serious problems. Both domestic and industrial uses of the coastal zone generate considerable alterations with long-term deleterious consequences. Human influence can be found in some sort of construction or alterations in water and sediment quality.

The most imaginative attempts have been used to protect and stabilise the coast. Offshore structures such as *breakwaters* are employed to break up the wave energy, preventing it from reaching and eroding the shoreline. *Groins* are structures attached normal to the shoreline, extending across the beach out into the surf zone designed to stabilise the coast by trapping sediment transported by longshore currents. *Jetties* look like groins but are typically larger and their function is to stabilise inlets and to permit continuous navigation by preventing the infilling of the inlet by sediments. The interference caused by these structures in the natural propagation of waves and currents and movement of sediments creates costly anomalies in the morphology of the shore. The position of shorelines in such regions is likely to be outdated quickly and yet another delineation of the shoreline will have to take place. However, the presence of these protection and stabilisation structures can represent a problem when interpreting the image. This may not be a big problem for a human interpreter but is something that can certainly generate misleading shoreline determinations if the process is to be undertaken automatically by computer software.

2.6 Historical Importance of Boundary Tracing Research

Automatic (or even semi-automatic) shoreline feature extraction has not received much attention in the past decades. Much of the research that has been done relates more to edge detection techniques *per se*, in road extraction, and in biomedical research.

2.6.1 Edge Detection in General Image Analysis Research

Finding edges in an image is of vital importance because much of its information content can be “summarised” by displaying the edges in a specific grey-level value (GLV) and the rest of the image in another one (Kelly, 1971). As a consequence, an image with a radiometric resolution (quantisation level) of n bits can be reduced to one bit (binary image). For instance, an image with 256 GLVs (or 2^8 GLVs, also known as 8-bit images) can be reduced to 2^1 GLVs, where 1 can be assigned to the edges and 0 to all the other pixels.

Edges have been thoroughly studied and are relatively easy to identify in images because of their own distinctive nature. Roberts (1965), Prewitt (1970), Sobel (1970), Kirsch (1971), Frei and Chen (1977), Robinson (1977), and Canny (1986) all have developed edge operators, which in turn carry their creator’s names. They are just a few of the researchers that have been studying edges since early 1960’s.

Human image analysts can easily pick up tonal differences and determine boundaries. These boundaries are grey-level discontinuities that might indicate the location of a feature of interest. However, the task of simply examining a dataset visually

does not always yields the best boundaries. There are cases when the tonal differences of image pixels vary over longer distances, and depending on what one is looking for an edge might just as well be overlooked.

Sudden changes in GLVs from one pixel to another indicate the existence of a boundary. But the magnitude of the change has to be decided first in order to determine if the changes are significant. Ideally, edges are a *step* function and can be either *sharp* or *diffuse*. Diffuse or wide edges (Figure 2.1b and 2.1c) consist of closely-located, smaller edges representing a transition over a number of pixels (Galbiati, 1990). High spatial frequencies are associated with sharp edges and low spatial frequencies with diffuse edges.

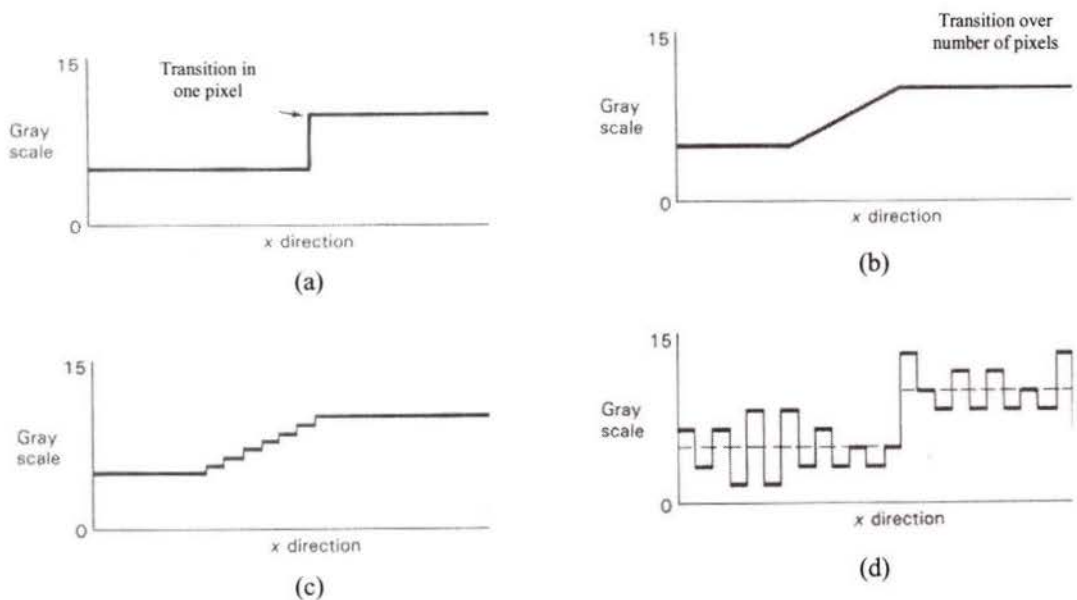


Figure 2.1 – Edge types. (a) ideal step; (b) wide; (c) breakdown of wide edge; (d) ideal step with noise (Modified from Galbiati, 1990).

The term *spatial frequency* refers to the degree of change in tonal variation of an image. Areas of high spatial frequency are represented by pixels whose grey-level values change abruptly, giving the image a “rough” appearance. On the other hand, low spatial frequencies provide a “smoother” appearance to the image because grey-level values vary gradually over large areas.

Edges are usually associated with noise (Weeks, 1996). By removing only low spatial frequencies from the image, noise will be reduced and edges enhanced. Edge detection operators (filters) were developed for this purpose and there are three different approaches to this process (Richards, 1986). These are:

1. By using edge-detection templates (filtering kernels);
2. By calculating spatial derivatives (differential gradients); or
3. By subtracting a smoothed image from its original.

The use of matching templates (Chapter 3) is preferred over the other two because of its ease of use and also because the more-sophisticated spatial derivatives can be approximated to equivalent discrete templates (Davies, 1990).

The edge-enhanced image is then thresholded to produce an edge map, in which noise or minor gradients are suppressed. The third method does not allow for the creation of an edge map. It only sharpens the original image by maintaining the high spatial frequencies, thus yielding better visualisation of edges and lines of high gradient (Richards, 1986; Lillesand *et al.*, 1994).

2.6.2 Road Detection

The most similar case to the shoreline detection problem is road detection, which involves road finding, tracking, and linking. Applications derived from these seem to proliferate, but the methodologies employed sometimes differ radically from the one used here. Certain algorithms use low-resolution images for finding single-pixel roads whereas others find roads that can be many pixels wide by using high-resolution images (Dillabaugh, 2000).

Gong *et al.* (1996) compared the efficiency of a linear extraction algorithm and three different classification algorithms (unsupervised, supervised and contextual). The gradient direction line detector seemed to work well only on infra-red bands of high-resolution aerial images (1.6 m) whereas previous studies using the same algorithm showed good results on visible bands of lower-resolution spaceborne images (10 m and 30 m). Results from the classification methods showed that the variability of GLVs among different road surfaces creates too many classes (unsupervised) and makes the creation of few training classes (supervised) impossible. The cover frequency-based contextual classifier suffered from the interference of trees along the roads but yielded the best results where the GLVs were homogeneous.

Steger *et al.* (1996) used high-resolution and low-resolution (decimated, resampled high-resolution images) aerial images to extract roads and update *geographical information systems* (GIS) databases automatically. Roads were first extracted from low-resolution images using Deriche's edge detector (Deriche, 1987). The single-pixel lines were then combined with the extracted, multiple-pixels roads from

high-resolution images. These parallel lines enclosing homogeneous areas are selected if a line extracted from the low-resolution images is found between them. The edge detector used here was a modification of Canny's algorithm which assumes that edges are image points with locally-maximal gradient magnitude (Canny, 1986). The directional derivative of the gradient magnitude in the direction of the gradient is then calculated, showing only dominant structures in the output image. The single-pixel roads are important for guiding the selection of the parallel lines because Deriche's operator confuses the edges of nearby features sometimes joining them together.

Fortier *et al.* (2001) and Hui *et al.* (2001) employ *active contours* or "snakes" for updating road databases using low- and high-resolution aerial imagery. Active contours can be used to update the position of features given an existing linear dataset is overlaid on an image. The algorithm deforms the lines based on their energy (function of geometry, pixel values, and user interaction) until they fit the feature of interest in the image. The best fit for a feature represents the minimum total energy for all possible lines. Results are usually good but the requirement of existing lines representing features of interest adds one more step to the process.

A semi-automatic road extraction method using wavelet transforms was proposed by Couloigner *et al.* (2000). Wavelets make it possible to recover weak signals from noisy images without blurring details. It is basically a Gaussian smoothing operator that attenuates high frequencies. The smoothed image is subtracted from the original, resulting in an image with only high spatial frequencies. A searching algorithm then looks for edges in both the original and smoothed images while a different algorithm searches for edges in the sharpened image.

Fischler *et al.* (1981) present a methodology similar to the one used in this research for low-resolution images. Note that in their case, low resolution was defined as roads having an image width of less than four pixels, with no mention of how large the image pixels could be. Different edge operators were used in order to try to extract the best roads. Then, A* and F* tracking algorithms were employed to find an optimal path/road (A* discussed in Chapter 3). These algorithms try to find a minimal cost path starting and ending at specified points. Pixels are assigned a cost according to an evaluation function entered by the user. The main difference between both is that the A* algorithm focuses on individual pairs of pixels whereas F* calculates costs for all line and column elements.

2.6.3 Boundary Delineation in Medical Research

Medical papers concerned with boundary delineation problems are extensively found in scientific literature. Beginning in the late 1970's, a rapid development of image processing techniques boosted by a technological evolution of computers provided new tools for medical research. One area that rapidly absorbed all these new resources was ultrasonic echocardiography (Kisslo *et al.*, 1976; Schiller *et al.*, 1978; Keating *et al.*, 1979).

Skorton *et al.* (1981) describe a method for identifying endocardium borders using ultrasound images. They compared the results obtained from a simple image bilevel thresholding and edge detection using the Sobel operator, after image smoothing through averaging. The accuracy of the edge map boundaries was then tested against boundaries

obtained through manual tracing on echocardiograms. Accurate semi-automatic endocardial definition benefits from the fact that these images have a very high contrast with simple bimodal histograms. For this reason, either thresholding or edge detection operators will always yield clear boundaries.

Similar methodology was used by Zwehl *et al.* (1983) but they employed the Laplacian operator on the smoothed image in order to enhance the edges. The endocardial boundaries were then traced on the thresholded image automatically.

Collins *et al.* (1984) compared the endocardial boundaries obtained by applying four different edge detector operators on images that had been first contrast-enhanced, edge-sharpened and median-smoothed. Again, due to the characteristics of endocardial images, all four operator (Sobel, Laplacian, Roberts and bilevel thresholding) performed very well when compared to manually-traced boundaries.

The “pre-processing” (smoothing, edge enhancement) seen in almost all of the above works reflect the sensitivity of the edge operators. Thus, it is necessary to “clean up” the images by increasing image contrast, sharpening edges, and/or smoothing out textural variations before the edge detection takes place.

2.6.4 Delineation of Coastline Contours

Recent papers address the shoreline detection problem from different perspectives. Onana *et al.* (2001) compared two texture analysis methods on synthetic-aperture radar (SAR) magnitude images. Both methods employ classification procedures

(supervised and unsupervised, respectively) to detect shorelines by using statistical first-order texture measures (TM), such as mean, skewness, and variance. The textural-multiresolution (TMR) approach classifies TM images into water and land classes using the *maximum likelihood classifier*. The geometric multiresolution (GMR) approach first produces an image with a lower spatial resolution than the original through a second *derivate gaussian wavelet* decomposition. TM images are then generated at lower resolution for subsequent segmentation using a *k-means classifier*. The shoreline is then extracted by running an edge detection algorithm on the image. The number of steps is quite large and can be summarised in Table 2.

Table 2
Comparison between TMR and GMR methods

TMR	GMR
Histogram analysis of original image for two distinct classes – water and land	Wavelet decomposition of original image and generation of lower-resolution images
Generation of TM images	Generation of TM images
Collection of training datasets on TM images	Histogram analysis of best TM image that shows good separation of water and land
Maximum likelihood classification on TM images using water and land classes	Segmentation of chosen TM using k-means classification
Edge detection of classified image	Resulting lower-resolution classified TM is transformed back to original resolution
---	Edge detection of classified image

Classification and segmentation methods present an interesting approach for detecting shorelines by separating water pixels from other pixels. However, in most cases, water and land cannot be easily grouped into only two homogeneous regions. Water pixels tend to have lower GLVs than land pixels in visible (0.38 μm to 0.76 μm approximately) and infra-red (0.7 μm to 0.9 μm) images. The problem arises from pixels whose GLVs cannot be clearly classified as either region. Intermediate classes could be created and then merged into a respective “primary” class, either water or land, but a decision criterion would have to be established for the assignment.

Bo *et al.* (2001) presented another method similar to the one above. Again, a texture analysis for detecting shorelines was implemented but this time a soft classifier was used on SAR images. A *fuzzy connectivity algorithm* based on fuzzy sets theory (Zadeh, 1965) tried to overcome the problem of non-homogeneous regions by creating classes based on GLVs distribution. Pixels were classified according to four different texture features (mean, variance, brightness uniformity, and contrast) after the original image was segmented by the fuzzy connectivity algorithm. Each of the four new images had pixels with values representing their degree of membership to the respective texture feature, varying from 0 to 1. A threshold was then applied to extract the shoreline.

It should be stressed that neither of the above methods creates shoreline vectors but only an edge map that could be later thinned or vectorised, adding another step to the process. Furthermore, SAR images present a different challenge from optical images when processing the dataset for edge detection, thence the use of classification methods using textural information.

The positional accuracy of the extracted shorelines was not mentioned in either paper. This could indicate lack of official shoreline vectors or accuracy assessment based only on visual inspection.

2.7 Digital Image Processing Methods Used in Feature Delineation and Extraction

The manipulation and interpretation of digital images started slowly in the 1960's with a few researchers venturing onto the new and promising area of analysing digitised aerial photographs. After the launch of National Aeronautics and Space Administration's (NASA) Landsat-1 in 1972, digital imagery data became widely available for the scientific community at the same time that digital computers were becoming more powerful and commonplace. Today, four decades later, digital image processing has become the standard practice with the technological advances in hardware and software (Lillesand *et al.*, 1994).

The delineation of specific features in digital images requires the user to process digital values so to first increase the visual distinction between them by applying enhancements to the imagery, then by manipulating individual spatial features of interest and finally by extracting the feature of interest from the imagery. *Enhancements* include different contrast manipulation methods such as *grey-level thresholding*, *level slicing* and *contrast stretching*. The *manipulation of individual spatial features* is usually

accomplished through *edge enhancement, spatial filtering, Fourier analysis* and *neural networks* (Moik, 1980). The complimentary and final step would be the extraction of the delineated features. This step consists of tracing individual features according to pre-determined criteria. In this particular case, shorelines would be traced and compared to already available shoreline vectors.

2.7.1 Preprocessing Operations

Image data acquired by airborne or spaceborne remote sensors are usually not ready for immediate analysis. It is first necessary to restore the data, both radiometrically and geometrically, to the radiant energy characteristics of the original scene in order to guarantee a minimum impact on the accuracy of the image analysis.

2.7.1.1 Radiometric Calibration

Sensor malfunction, improper sensor calibration, atmospheric attenuation and topography contribute to different degrees of data quality degradation or distortion. This means that the data recorded by the sensor may end up not being an accurate representation of the energy that was reflected or emitted by the features of interest.

2.7.1.2 Geometric Restoration

Preparing the data for the analysis also involves the removal or attenuation of geometric distortions introduced by different factors. These distortions can be *systematic* or *non-systematic*.

Systematic distortions (sometimes referred to as *systematic errors*) are mainly caused by earth's curvature (panoramic distortion), eastward rotation, scan skew, mirror-scan speed and platform velocity. They are often corrected at the receiving stations by using pre-defined models (Jensen, 1996). These distortions can be corrected by analysing sensor characteristics and platform ephemeris.

Non-systematic distortions (or *random errors*) are unpredictable and dealt with on an individual basis (Lillesand *et al.*, 1994). They are introduced by variations in platform's altitude and attitude (roll, pitch and yaw) and can only be corrected by using *ground control points* (GCP). These GCPs are points on Earth's surface where both image and map coordinates can be identified and used to rectify image data. A successful geometric restoration will generate an image with some very basic georeferencing information, thus only providing corner or centre coordinates. Further geometric correction (or even an orthorectification) is necessary to rectify every pixel of the image to both projection and coordinate system specified by the user.

2.7.2 Geometric Correction

Georeferenced images do not possess all the characteristics of geometric integrity that maps are supposed to display. Maps can be thought of as a flat representation of a part of Earth's spheroidal surface and all points/features must be accurately located on it.

The importance of geometrically correcting the data lies on the fact that for it to be used it must comply with standard cartographic projections and scales adopted world-wide. By doing so it is possible to compare data acquired at different periods of time with relatively good accuracy.

In most cases, only basic preprocessing operations (as mentioned in Section 3.5.1) will have been performed on remotely-sensed data before they are delivered to end-users. Thus, users are allowed to apply any projection, datum or ellipsoid, and resampling methods on their images. And by doing so, the previously-georeferenced images will have map coordinates for each pixel, being therefore called *geocoded images*.

The geometric correction technique consists of two primordial steps: Transformation of pixel coordinates and Resampling.

2.7.3 Image Enhancement

Image enhancement techniques are designed to improve the visual interpretability of remotely-sensed images by increasing the apparent distinction between the features in the scene. Airborne and spaceborne sensors acquire image data under a wide range of

illumination conditions. For example, the sensor responses over a poorly-lit arctic region are very different from those over a high-reflectance desert region. Consequently, the pixel values in most scenes occupy a small portion of the possible range of grey-level values. This is the main reason for the low contrast of raw images where similar features are almost indistinguishable. Basic image enhancements are classified as *point* or *local* operations. Point operations modify individual pixel brightness values independently (by performing simple arithmetic operations such as adding or subtracting a constant from each pixel, for example), whereas local operations take into account neighbouring brightness pixel values when modifying the values of individual pixels (therefore also known as *spatial* or *area* operations) (Lillesand *et al.*, 1994; PCI, 1997).

2.7.3.1 Grey-Level Thresholding

The simplest way of creating two distinct features is achieved by segmenting an image into two different classes. First, a threshold point is established based on visual as well as numerical inspection of image histograms in order to determine the *dominant minimum GLV* (or *threshold point*). All pixels with values above the threshold point are assigned a higher value specified by the analyst, and all pixels with values below the threshold point are assigned a lower value, usually 255 and 0, respectively (Russ, 1992).

Thresholding is a very effective method for finding rough boundaries and can be used as a first step in delineating approximations to the final boundary.

2.7.3.2 Density Slicing

This enhancement is similar to grey-level thresholding but it allows grouping (i.e. "slicing") the image grey-levels into an analyst-specified number of ranges. This feature makes it not as restrictive as thresholding, which created only two new classes. By looking at the histogram distribution of the image, the analyst can determine grey-level ranges that best represent individual features (for instance: water, beach or vegetation). Each of these ranges are assigned a unique digital number (DN) in the output image (Lillesand *et al.*, 1994; PCI, 1997). The advantage of this method lies on the creation of areas containing pixels with the same DNs. These areas end up giving the image the appearance of a contour map that might be beneficial for making a decision on the best location of the shoreline because coastal images usually do not show clear boundaries between water and land.

2.7.3.3 Contrast Stretching

Raw images (original images) acquired by remote sensors usually possess very low contrast. This is simply exemplified in Figure 2.2, where most image pixels are concentrated along a very narrow radiometric range (X-axis is also called the *brightness* or *quantisation* range; Y-axis represents the *absolute frequency* or *pixel count*). The increase of the image contrast is accomplished by stretching the image in display so that all the available DNs ranging from black to white (DNs 0 to 255 in an 8-bit image) are

used. There are different models available and they all follow the same idea. The most commonly used contrast stretch is the *linear* (or *min-max*) stretch **2.1**.

$$DN' = \left(\frac{DN - Min}{Max - Min} \right) 255 \quad (2.1)$$

where

DN' ≡ digital number assigned to pixel in the output image;

DN ≡ original digital number of pixel in the input image;

Min ≡ minimum digital number of the input image; and

Max ≡ maximum digital number of the input image.

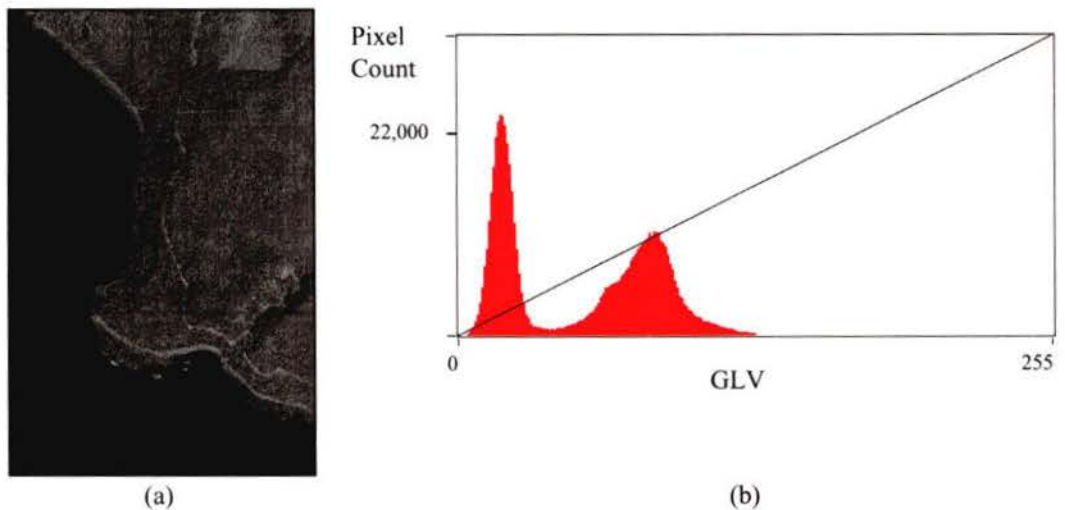


Figure 2.2 – (a) Satellite imagery with low contrast; (b) GLV Histogram.

Jensen (1996) recommends applying the linear stretch to images with Gaussian or near-Gaussian histograms (Figure 2.3). However, more sophisticated models have been created for images with multimodal histograms. The most commonly used are: *histogram equalisation*, *histogram normalisation* (also known as *Gaussian*), *square root*, *logarithmic*, *infrequency*, *adaptive*, *piecewise linear* and *percentage linear* (Lillesand *et al.*, 1994; PCI, 1997; ENVI, 2000).

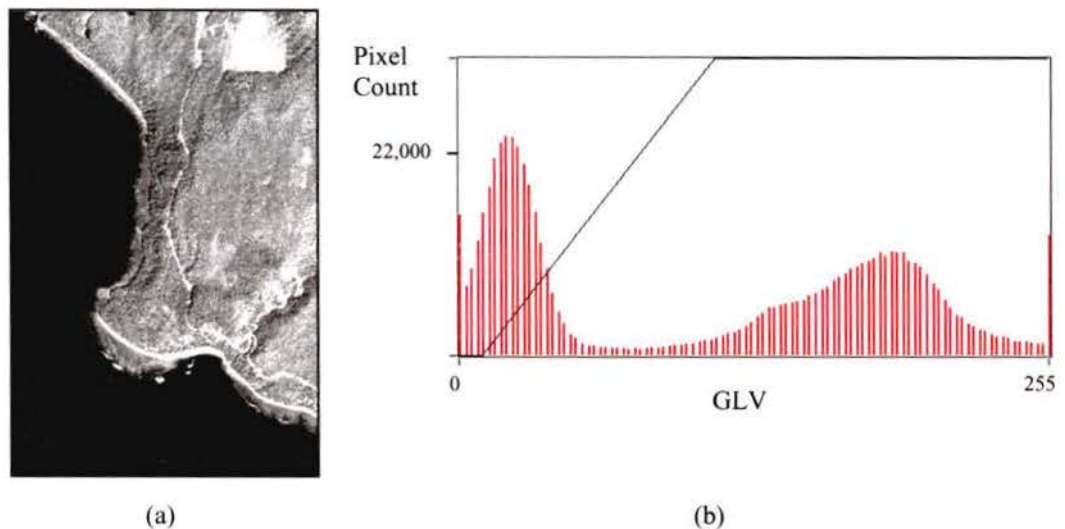


Figure 2.3 - (a) Satellite imagery with linear enhancement; (b) GLV Histogram.

Histogram equalisation is another stretch used in this research. It overcomes a major drawback inherent of the linear stretch, which assigns the same amount of GLVs in the new image to the pixels with lower count as it does to pixels with higher counts. By “equalising” the histogram, pixels are assigned GLVs based on their frequency of occurrence. This is an advantage over linear enhancement because some features with

similar GLVs will be enhanced according to influence they have on the dataset (Figure 2.4).

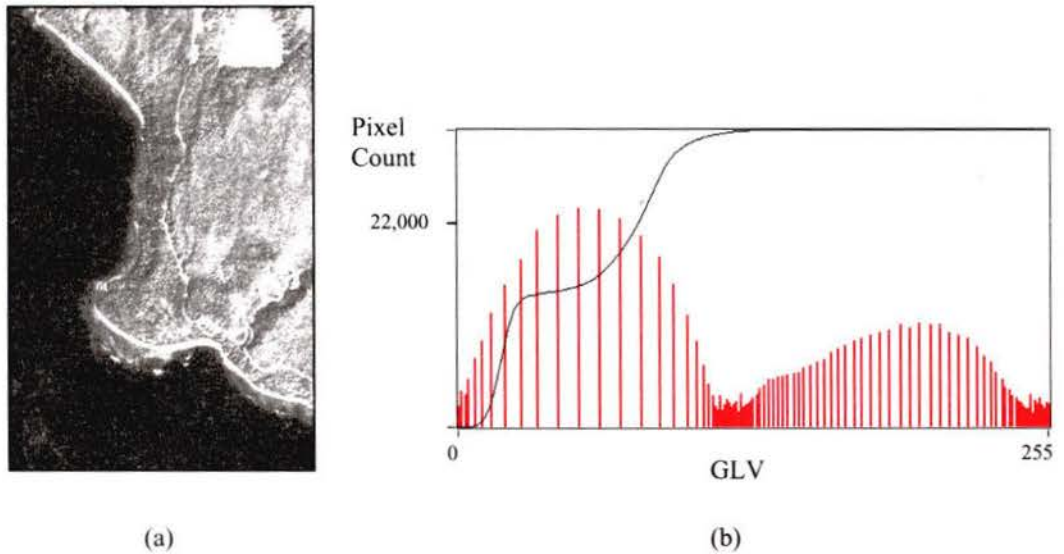


Fig 2.4 –(a) Satellite imagery with equalisation enhancement; (b) GLV Histogram.

2.7.4 Filters/Spatial Feature Manipulation

Various different spatial frequencies are responsible for tonal variations (sometimes called *roughness*) within digital images. High frequencies are responsible for abrupt changes in the brightness values of adjacent areas, whereas low frequencies describe gradual changes in the grey-levels of neighbouring pixels. When working with imagery data, it is often necessary to reduce their influence through the application of spatial filters. *High-pass* filters attenuate low frequencies while *low-pass* filters attenuate high-frequency components. They are all considered *local* operators because they modify

the original GLV of each pixel based on the brightness values of neighbouring pixels (Moik, 1980). See Appendix B for image examples.

2.7.4.1 Laplacian Filter

As it will be noted below, the coefficients of high-pass filter kernels can be positive or negative, and their sum is usually zero. There are many different types of edge detection filters available and the most commonly used for *wide* transition regions are based on the Laplace operator (*Laplacian*) for two dimensions in continuous functions 2.2 (Jain, 1989).

$$\nabla^2 f(x, y) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \quad (2.2)$$

The discrete approximation to the Laplacian would be 2.3:

$$\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \approx f(x+1, y) - 2f(x, y) + f(x-1, y) + f(x, y+1) - 2f(x, y) + f(x, y-1) \quad (2.3)$$

where

$$\Delta^2 x = [f(x+1, y) - f(x, y)] - [f(x, y) - f(x-1, y)] \quad (2.4a)$$

$$\Delta^2 y = [f(x, y+1) - f(x, y)] - [f(x, y) - f(x, y-1)] \quad (2.4b)$$

Therefore:

$$\nabla^2 f(x, y) \approx \Delta^2 x + \Delta^2 y \quad (2.5)$$

The application of this function can be more readily visualised using a 3x3 window, as in 2.6:

$$\begin{pmatrix} f(x-1, y-1) & f(x, y-1) & f(x+1, y-1) \\ f(x-1, y) & f(x, y) & f(x+1, y) \\ f(x-1, y+1) & f(x, y+1) & f(x+1, y+1) \end{pmatrix} \quad (2.6)$$

Consequently, 2.3 is implemented directly on an image by convolving 2.6, thus yielding 2.7:

$$\begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix} \quad (2.7)$$

2.7.4.2 Sobel Filter

Abrupt grey-level transitions are better estimated using other types of operators than the Laplacian. In this case, *gradient* operators are used to measure the gradient of the image in two orthogonal directions (x, y). The *Sobel* operator 2.9 uses a non-linear combination of pixels when calculating the magnitude of the edges (Sobel, 1970). In fact, the algorithm calculates two convolutions for each pixel, detecting vertical and horizontal

edges, respectively. Using a modified version of **2.6** for ease of understanding, the image sample can be described as **2.8**.

$$\begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix} \quad (2.8)$$

$$S = \sqrt{(\Delta x)^2 + (\Delta y)^2} \quad (2.9)$$

where

$$\Delta x = (c + 2f + i) - (a + 2d + g) \quad (2.10a)$$

$$\Delta y = (g + 2h + i) - (a + 2b + c) \quad (2.10b)$$

The Sobel convolution windows based on **2.10a** and **2.10b** are seen below:

$$\Delta x = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} \quad (2.11a)$$

$$\Delta y = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix} \quad (2.11b)$$

The value S will be assigned to the pixel at the centre of the kernel in the output image.

2.7.4.3 Roberts Filter

Roberts gradient operator is a simpler filter than the previous two. It employs a 2x2 window reducing computational time (Roberts, 1965). The gradient is calculated as the partial derivatives in X- and Y- directions. The discrete approximation is shown below.

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \quad (2.12)$$

$$|\nabla f(x, y)| \approx \sqrt{(\Delta x)^2 + (\Delta y)^2} \quad (2.13)$$

where

$$\Delta x = a - d \quad (2.14a)$$

$$\Delta y = b - c \quad (2.14b)$$

As with the previous filters, Roberts convolution windows are shown below:

$$\Delta x = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \quad (2.15a)$$

$$\Delta y = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \quad (2.15b)$$

The value $|\nabla f(x, y)|$ will be assigned to the pixel at the top left corner (a) of the kernel in the output image.

2.7.4.4 Prewitt Filter

The Prewitt filter employs the same computational rationale of the Sobel filter presented in Section 2.7.4.2 (Prewitt, 1970). However, the difference lies on the coefficients of the convolution windows, as seen in 2.16 and 2.17 below.

$$\Delta x = \begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix} \quad (2.16)$$

$$\Delta y = \begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix} \quad (2.17)$$

2.7.4.5 Pixel Difference Filters

Two other examples of how the variation on the coefficients of the Sobel filter convolution mask can yield very different results are demonstrated below. The convolution masks for the *Simple Pixel Difference* filter and the *Separated Pixel*

Difference filter are shown in **2.18** and **2.19**, and **2.20** and **2.21**, respectively (Lyon, 1999).

$$\Delta x = \begin{pmatrix} 0 & -1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad (2.18)$$

$$\Delta y = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & 0 \end{pmatrix} \quad (2.19)$$

$$\Delta x = \begin{pmatrix} 0 & -1 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \quad (2.20)$$

$$\Delta y = \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & -1 \\ 0 & 0 & 0 \end{pmatrix} \quad (2.21)$$

These filters perform relatively well in detecting edges but the *Separated Pixel Difference* filter is more suitable in situations where edge lines are exactly two pixels wide. The reason for this is in the layout of masks **2.20** and **2.21**.

2.7.4.6 Frei-Chen Filter

This filter is another attempt to pick out horizontal and vertical edges and it also employs X- and Y-templates, shown in **2.22** and **2.23** (Frei *et al.*, 1977).

$$\Delta x = \begin{pmatrix} -1 & 0 & 1 \\ -\sqrt{2} & 0 & \sqrt{2} \\ -1 & 0 & 1 \end{pmatrix} \quad (2.22)$$

$$\Delta y = \begin{pmatrix} -1 & -\sqrt{2} & -1 \\ 0 & 0 & 0 \\ 1 & \sqrt{2} & 1 \end{pmatrix} \quad (2.23)$$

2.7.4.7 Smoothing Filters

These filters are designed to attenuate areas where brightness values change dramatically over short distances. The removal of the high-frequency detail in these areas will emphasise low-frequency features - hence their commonly-used name *low-pass filters* - what could potentially be helpful for the performance of the aforementioned edge detectors. High-frequency features are usually short ocean waves and/or *salt-and-pepper noise* (also known as *shot noise* or *bit error*), the latter caused by the resolution limitations of the digital scanning/imaging system of Landsat and SPOT; or, in the case of the 1-m aerial orthophotos and 4-m KFA-1000 photograph, due to the mechanical/optical system (Weeks, 1996). By removing high-frequency features, simply called *noise* in this particular regard from now on, the brightness values of large areas will be maintained and the edge detectors may be able to detect the shoreline edges without interruption caused by small, noisy features.

At the same time, it is necessary to keep the smoothing kernel to a reasonable size so as to avoid too much blurring of the image. By increasing the kernel size, important edge features become diffuse, causing them to migrate from their original location. Consequently, extracted shoreline vectors might be displaced a certain number of pixels in either direction. Blurring is also affected by maintaining the kernel size and assigning different weights to the coefficients, as demonstrated below in 2.24 to 2.28. Note that the coefficients of low-pass filters must be positive and their sum is usually equal to one (Galbiati, 1990).

$$A(\textit{Average}) = \begin{pmatrix} \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \end{pmatrix} \quad (2.24)$$

$$B = \frac{1}{10} \begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{pmatrix} \quad (2.25)$$

$$C = \frac{1}{12} \begin{pmatrix} 1 & 1 & 1 \\ 1 & 4 & 1 \\ 1 & 1 & 1 \end{pmatrix} \quad (2.26)$$

$$D = \frac{1}{20} \begin{pmatrix} 1 & 1 & 1 \\ 1 & 12 & 1 \\ 1 & 1 & 1 \end{pmatrix} \quad (2.27)$$

$$E = \begin{pmatrix} \frac{1}{81} & \frac{1}{81} & \frac{1}{81} & \frac{1}{81} & \frac{1}{81} & \frac{1}{81} & \frac{1}{81} & \frac{1}{81} & \frac{1}{81} \\ \frac{1}{81} & \frac{1}{81} & \frac{1}{81} & \frac{1}{81} & \frac{1}{81} & \frac{1}{81} & \frac{1}{81} & \frac{1}{81} & \frac{1}{81} \\ \frac{1}{81} & \frac{1}{81} & \frac{1}{81} & \frac{1}{81} & \frac{1}{81} & \frac{1}{81} & \frac{1}{81} & \frac{1}{81} & \frac{1}{81} \\ \frac{1}{81} & \frac{1}{81} & \frac{1}{81} & \frac{1}{81} & \frac{1}{81} & \frac{1}{81} & \frac{1}{81} & \frac{1}{81} & \frac{1}{81} \\ \frac{1}{81} & \frac{1}{81} & \frac{1}{81} & \frac{1}{81} & \frac{1}{81} & \frac{1}{81} & \frac{1}{81} & \frac{1}{81} & \frac{1}{81} \\ \frac{1}{81} & \frac{1}{81} & \frac{1}{81} & \frac{1}{81} & \frac{1}{81} & \frac{1}{81} & \frac{1}{81} & \frac{1}{81} & \frac{1}{81} \\ \frac{1}{81} & \frac{1}{81} & \frac{1}{81} & \frac{1}{81} & \frac{1}{81} & \frac{1}{81} & \frac{1}{81} & \frac{1}{81} & \frac{1}{81} \\ \frac{1}{81} & \frac{1}{81} & \frac{1}{81} & \frac{1}{81} & \frac{1}{81} & \frac{1}{81} & \frac{1}{81} & \frac{1}{81} & \frac{1}{81} \\ \frac{1}{81} & \frac{1}{81} & \frac{1}{81} & \frac{1}{81} & \frac{1}{81} & \frac{1}{81} & \frac{1}{81} & \frac{1}{81} & \frac{1}{81} \\ \frac{1}{81} & \frac{1}{81} & \frac{1}{81} & \frac{1}{81} & \frac{1}{81} & \frac{1}{81} & \frac{1}{81} & \frac{1}{81} & \frac{1}{81} \end{pmatrix} \quad (2.28)$$

In cases where the sum of the coefficients is greater than one (2.29), amplification of the brightness values results. And if the sum is less than one (2.30), attenuation will result and pixel brightness values will be reduced, resulting in lighter or darker images.

$$F = \frac{1}{20} \begin{pmatrix} 5 & 4 & 5 \\ 4 & 20 & 4 \\ 5 & 4 & 5 \end{pmatrix} \quad (2.29)$$

$$G = \frac{1}{16} \begin{pmatrix} 1 & 1 & 1 \\ 1 & 4 & 1 \\ 1 & 1 & 1 \end{pmatrix} \quad (2.30)$$

The key for a successful edge detection might lie on smoothing an image to the point where no other features (be they point, linear or area features) else than shorelines would be detected.

2.7.4.8 Median Filters

These filters are similar to smoothing filters but differ by the fact that they do not calculate a weighted sum. Instead, the sliding kernel ranks underlying pixels according to their GLVs and the median value is chosen as the new GLV for the pixel in the output image located exactly under the kernel's centre (Castleman, 1996). In other words, when the number of elements n in a kernel is odd, the output GLV is the GLV of the input pixel which is greater than $(n^2-1)/2$ of the neighbouring pixels and less than $(n^2-1)/2$ of them.

It is a very effective filter type for reducing noise but is usually slower than smoothing filters (Showengerdt, 1983). Some median filters used in this research can be seen below in 2.31 to 2.36. Note that the value 0 indicates that image pixels in that position of the kernel will be ignored during the calculation of the median.

$$M_1(\text{MedianCross}3x3) = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{pmatrix} \quad (2.31)$$

$$M_2(\text{MedianSquare}3x3) = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \quad (2.32)$$

$$M_3(\text{MedianSquare}5x5) = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix} \quad (2.33)$$

$$M_4(\text{Octagon}5x5) = \begin{pmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{pmatrix} \quad (2.34)$$

$$M_5(\text{MedianCross}7x7) = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix} \quad (2.35)$$

$$M6(\text{Diamond}7 \times 7) = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix} \quad (2.36)$$

2.7.4.9 Canny Filter

This operator tries to find optimal edges by first applying a smoothing filter followed by a simple first-derivative operator. Then, a tracking operator applies a bilevel threshold to the filtered image (Canny, 1986).

A Gaussian convolution is used to smooth the image. The two-dimensional (2-D) Gaussian distribution $G(x,y)$ has the form:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (2.37)$$

where

$\sigma \equiv$ standard deviation of the distribution.

Masks **2.24** is a simple example of a Gaussian filter, while masks **2.25** to **2.28**, **2.38** and **2.39** approximate more closely to a Gaussian profile.

$$G(\text{Gaussian}3 \times 3) = \frac{1}{16} \begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix} \quad (2.38)$$

$$H(\text{Gaussian}7x7) = \frac{1}{140} \begin{pmatrix} 1 & 1 & 2 & 2 & 2 & 1 & 1 \\ 1 & 2 & 2 & 4 & 2 & 2 & 1 \\ 2 & 2 & 4 & 8 & 4 & 2 & 2 \\ 2 & 4 & 8 & 16 & 8 & 4 & 2 \\ 2 & 2 & 4 & 8 & 4 & 2 & 2 \\ 1 & 2 & 2 & 4 & 2 & 2 & 1 \\ 1 & 1 & 2 & 2 & 1 & 1 & 1 \end{pmatrix} \quad (2.39)$$

After smoothing the image, usually a Laplacian filter is applied to detect edges. Marr *et al.* (1980) combined the Gaussian and Laplacian operators and developed the *zero-crossings filter (Laplacian of the Gaussian, LoG)*. By implementing the LoG operator 2.40, only one convolution is performed. Canny used LoG and extended its utility by adding a thresholding operator to it. This operator tracks the best edges, and assigns zero to all pixels that are not part of the edge. Hysteresis controls the tracking process by starting the process at a point with GLV higher than a threshold T_1 and continuing until a point with GLV lower than a threshold T_2 is reached. Note that T_1 must be higher than T_2 (Fisher *et al.*, 2002).

$$\text{LoG}(x, y) = -\frac{1}{\pi\sigma^4} \left[1 - \frac{x^2 + y^2}{2\sigma^2} \right] e^{-\frac{x^2 + y^2}{2\sigma^2}} \quad (2.40)$$

2.7.5 Spatial Feature Extraction/Line Tracing

Identifying specific objects in images is often performed naturally and it is usually accomplished by identifying their outlines. Making machines that simulate human behaviour is a challenge that has been the focus of research for the past 40 years. Initially,

Greanias *et al.* (1963) and Mason *et al.* (1968) implemented contour following algorithms for the extraction of hand-written and digitised machine printing numerals. Similar algorithm was used by Ledley (1964) in the analysis of chromosome images.

The process of delineating the shoreline automatically looks for abrupt changes in the pixel brightness values usually identified as *edges* or *edge elements*. Edge operators and other spatial filters are used to extract as much information about edges as possible. This newly-transformed image should be ready for the edge tracing algorithm

When using this processed image for the extraction of the shoreline, one would soon verify that many different edges had been detected and that therefore several different lines were drawn by the software. Every single edge that does not represent the shoreline would then have to be deleted, or have its influence on other edges reduced, in another step.

Minimising the number of steps is essential for raising productivity. Consequently, a better approach would consist of 'guiding' the software. In this case, the image analyst would seed points along the shoreline (assuming that the analyst has previously determined the parameters in which to base the shoreline) and the software would examine the edges following the seed-points. This *heuristic search* is faster and somewhat more reliable than fully automated delineation because of the input of previously gathered information by the analyst. As a result, a line identifying the shoreline is generated on the image (Lyon, 1999).

3. The Search for the Holy Algorithm

3.1 Methodology

The methods outlined here incorporate knowledge about coastal geomorphology, imagery characteristics, digital image processing techniques, and processes involved in the semi-automated tracing and digital extraction of shoreline features. They extend information presented in the previous chapter and give an overview of how the research was conducted.

3.2 Guidelines for Shoreline Determination

The true determination of the shoreline is hard to accomplish using remote sensing data. This might be due to the slightly different spectral responses of water and land in some specific wavelengths. However, even when it is possible to make a clear distinction between water and land, there is a certain lack of confidence about the real position of the shoreline regarding the tides and wave action. According to Shalowitz (1964) and NOAA, the definition of shoreline adopted by the US National Geodetic Survey (NGS) on nautical charts and surveys is the mean high-water level (and consequently it is synonymous with coastline). The Canadian Hydrographic Service (CHS) adopts a definition slightly different, where the shoreline is referred to as the level

of the higher high-water large tide (HHWLT). This datum is defined as the average of the highest high waters (one from each of 19 years of tidal predictions) (Figure 3.1).

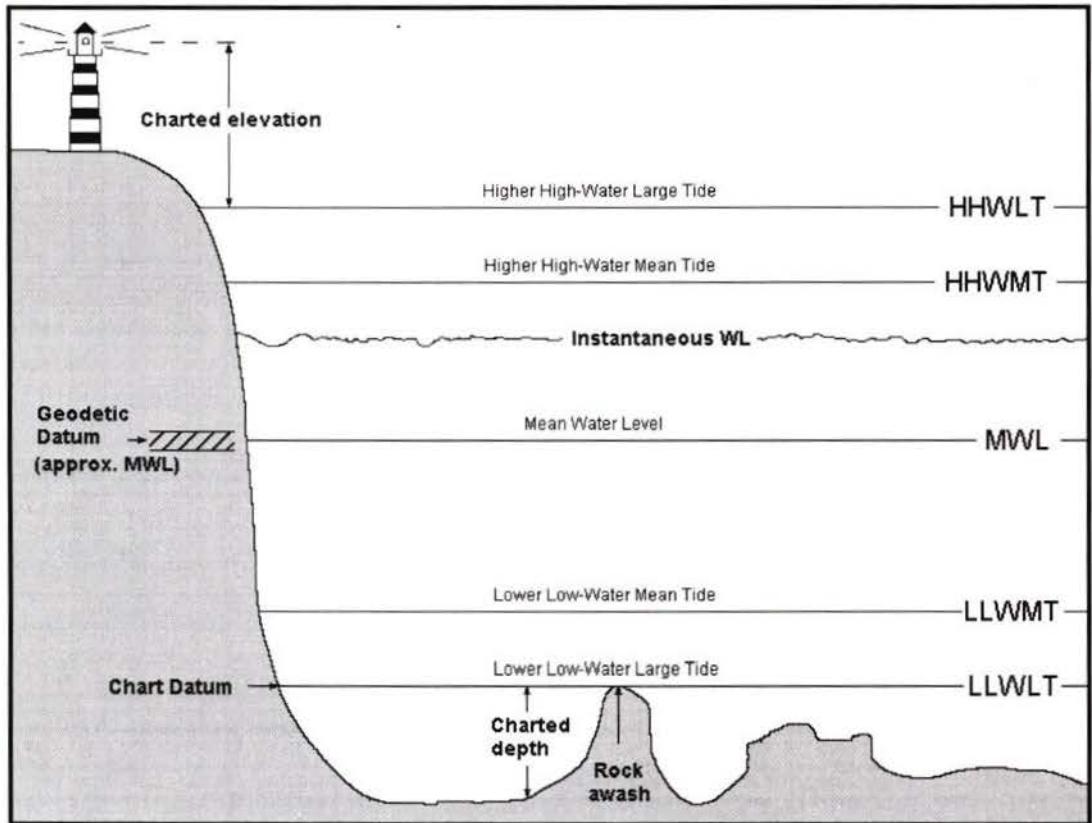


Figure 3.1 – Water Levels adopted by the Canadian Hydrographic Service (modified from Forrester, 1983).

These definitions show the need for overcoming the problem of the migration/variation of the shoreline during a day. However, it is still possible to impose further criteria on them. Shalowitz (1964) mentions the possibility of designating high-tide and low-tide shoreline according to the position of the water line at high and low tides. This might turn out to create more confusion amongst mariners and coastal zone users. For this reason this definition is no longer used.

The Remote Sensing Division of the U.S. National Geodetic Service is responsible primarily for surveying the coastal region and navigable shoreline water areas. The delineation of the shoreline is obtained through stereo-photogrammetry using tide-coordinated aerial photography controlled by kinematic Global Positioning System (GPS) techniques. All this work is done in order to ensure the accuracy of the shoreline position (NGS, 1999).

In Canada, the delineation of shorelines is done by CHS. Different methods are adopted depending on the availability of resources allocated for that purpose. In some areas, demarcating water lines for the purpose of establishing a vertical datum is performed locally using stakes for measuring tide height while in others, tide-controlled water line surveys (through aerial photography using air-to-ground communication) are conducted (Nichols, 1983).

But updating shoreline limits could still be performed using less expensive methods and keeping budget, errors and shifts reasonably low. Three different scenarios of data availability could be presented for the delineation of the shoreline:

- 1) Availability of remotely-sensed digital imagery - most common situation where only digital imagery is available to the user.
- 2) Availability of remotely-sensed digital imagery and tide height data for the time of image acquisition.
- 3) Availability of remotely-sensed digital imagery, tide height data for the time of image acquisition and digital elevation model (DEM) of the area.

Using tide height tables would certainly provide further information that could help determine the position of the shoreline more accurately but matching this information with previously-acquired imagery can be a difficult problem to overcome. The information derived from digital elevation models could also help delineate the shoreline provided that the terrain with elevation values of zero match the *datum of elevations* of the nautical charts (the same higher high-water large tide datum used to define the shoreline). But this is usually not true for land/above-sea surface DEMs. In this case, the Geodetic Survey of Canada adopts the *geodetic datum* as the reference surface for elevations, which turns out to be an approximation of the geoid or the mean sea level. According to the Canadian Hydrographic Service, the mean sea level (or mean water level) is the average of all hourly-observed water levels at a particular gauging station over some period of time. Consequently, the DEMs lowest point would not match the shoreline definition adopted by CHS. An alternative in this case would be to use bathymetric digital elevation models. However, the datum for these bathymetric DEMs is usually the *chart datum* (lower low-water large tide for CHS and mean lower low-water for NOAA). Again the shoreline would not be realistically represented.

It seems though that the three aforementioned options for shoreline delineation lack the desired accuracy standards. Apparently, tide-coordinated remote sensing techniques are the only and most accurate methods for mapping the shoreline. However, these procedures are expensive and not always available for short-term mapping projects.

Another option could still be proposed. In this scenario, some assumptions would have to be made prior to the shoreline determination and they would have to be based on the availability of data, time constraints and coast type. In most cases, only imagery data are available and no further data are provided. Under such circumstances, determining the shoreline can still be accurate enough to overcome the problem of mismatching the official shoreline depicted on nautical charts. Still, it will not always be possible to clearly distinguish the shoreline for the reasons explained above. Therefore, in such cases, the position of the shoreline would have to be determined arbitrarily according to some criterion.

The great variety of coast types plays a major role in the shoreline determination when it is really difficult to visualise the *land-water line*. The shoreline would then have to be assigned to the feature closest to the water. This would allow classifying the vegetation-land boundary as the shoreline. Certainly this is not the best result to be expected but it could serve primarily as a guide for a future shoreline determination using imagery that depicts the water-land line more clearly.

Maybe this decision would not be suited for steep rocky coasts, where the vegetation boundary is usually too high to represent the true shoreline. Consequently, the shoreline would be assigned to the water-land boundary. These particular situations would have to be dealt with on each individual region within each image in order to evaluate the real impact – in legal terms – of this apparent discrepancy of standards. In general, because of the difficulty of giving the shoreline a solid definition, the Canadian

and U.S. courts often use the vegetation line or high-water level approach, respectively, as a guide (Shalowitz, 1964; O'Reilly, 2000).

As outlined in Section 2.6.4, none of the references mentioned how the accuracy of the extracted shoreline was obtained. The lack of consistency in the methodologies for the shoreline determination in the scientific, legal and general literature made us come up with our own design based on visual comparisons made between official shoreline vectors and shoreline vectors extracted using the methods outlined here.

3.3 Area of Study

The area of study covers 350,000 ha of Clayoquot Sound, located on the west coast of Vancouver Island, British Columbia. It extends from 49°N, 125°35'W (Ucluelet) to approximately 49°49'N, 126°35'W (Hesquiat Peninsula) (Figure 3.2). The area provides habitat for a wide variety of animals and is known for its vast, old-growth forests. It is also important to mention that it is constantly the subject of land-use, logging and wildlife protection debates. Several different types of coasts can be found in the area, thus making it an ideal area for testing the performance of the shoreline extraction algorithm.

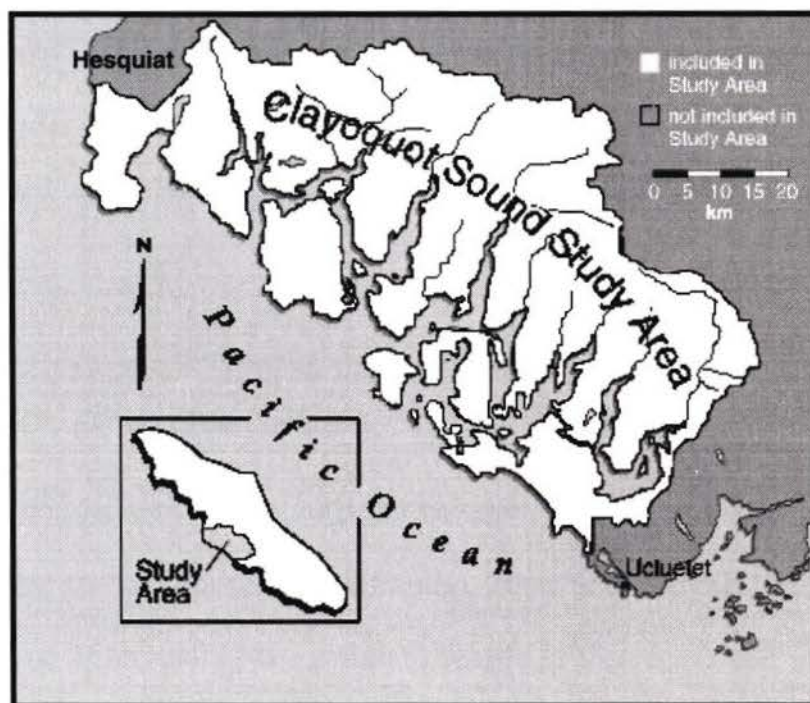


Figure 3.2 – Clayoquot Sound (Modified from the Province of British Columbia, 1993). Projection: Albers Equal-area.

3.4 Datasets

3.4.1 Images

The images used for this research were acquired in different years (during the 1980's and 1990's) and at different times of day. Initially, efforts were made to keep some sort of time consistency between images in order to minimise the influence of tides. However, due to time constraints, data availability and a lack of tide information, tidal influence would only be considered a direct factor in the determination of shorelines when it could be visually verified on the images and supporting metadata (such as time of

acquisition) were made available. At this stage, only optical sensors were used. Radar imagery was not considered for this study because the complex nature of radar data involving different frequency and polarisation modes requires extra image processing techniques in order to use the shoreline algorithm presented here (Lillesand *et al.*, 1994; Jensen, 1996; PCI, 1997). Table 3 shows which images were used.

Imagery acquired by the KFA-1000 camera on board of the Russian Resurs-F1 satellite has a spectral sensitivity ranging from 450 nm to 680 nm (blue to red) and the spatial resolution of 4 m, making it possible to see many different types of coasts quite clearly. Aerial photographs with 1-m spatial resolution were also used, as well as 20-m SPOT 1 multispectral images (blue to infra-red), 10-m SPOT 3 panchromatic images (blue to red), and 15-m and 30-m Landsat 7 images of the region. All images had a radiometric resolution of 8 bits, i.e. images were displayed with 256 GLVs (0 for black, 1 to 254 for different shades of grey and 255 for white).

3.4.2 Vectors

The shoreline vectors extracted by the tracing algorithm were compared with official shoreline vectors for the area of study provided by the Terrain Resource Information Management (TRIM) program of British Columbia. These vectors were originally digitised at 1:20,000 scale (using the Universal Transverse Mercator (UTM) projection and North American Datum (NAD 1983)) from 1:60,000-scale aerial orthorectified photographs (orthophotos) (TRIM, 2002).

Table 3
Acquisition Date and Resolutions of Imagery Used

	Acquisition Date	Spatial Resolution (m)	Spectral Resolution (μm)
Aerial Orthophotos	1996	1	0.30-0.70
KFA-1000 Photograph	Unknown	4	0.45-0.68
Landsat 7	27 August 1999	15 30	0.50-0.90 0.45-0.52 0.52-0.60 0.63-0.69 0.76-0.90 1.55-1.75 2.08-2.35
SPOT 1	26 May 1993	20	0.50-0.59 0.61-0.68 0.79-0.89
SPOT 3	23 September 1995	10	0.51-0.73

3.5 Digital Image Processing

The computer system used in this study consisted of the following:

- Pentium™ 4 - 900 MHz central processor unit (CPU);
- 256 MB random-access memory (RAM);
- 4 MB video memory;
- 80 GB hard-drive memory;

- 19" video monitor with 1600 by 1200 pixels display resolution, with 16,777,216 display colors;
- Microsoft® Windows NT® Workstation 4.0 – Service Pack 6 (Operating System);
- Java™ 2 Platform – Standard Edition; and
- Keyboard and mouse.

Image processing and analysis were performed in two different software packages. Pre-processing operations and geometric correction (Sections 3.5.1 and 3.5.2) were performed using PCI Geomatica™ 8.0 from PCI Geomatics Enterprises, Inc. (Richmond Hill, Ontario). Spatial filtering and shoreline extraction were performed using the Kahindu Interface (Lyon, 1999).

3.5.1 Preprocessing Operations

The images used in this research were acquired from different sources and the only corrections applied to suppress systematic errors introduced by the sensor systems themselves were those regarding geometric restoration (Section 2.7.1.2). Random system errors like *striping*, *line drop-out*, *banding* or *line-start problem* would only be fixed on an individual basis.

No further radiometric calibration was performed on any of the images used. This decision was based on the fact that no real quantitative analysis that could be intimately

dependent on atmospheric or other influences (such as those mentioned in Section 2.7.1.1) would be performed on the data. In other words, by keeping the purpose of this research in mind, which was the extraction of shorelines from digital images acquired by different remote sensors, the radiometric attenuation imposed by the atmosphere would not necessarily be as significant as in studies where, for instance, the determination of biophysical information from water bodies (such as chlorophyll content, suspended sediment or temperature) or from vegetated areas are the main focus. Atmospheric attenuation plays a direct role in such cases by masking subtle differences in reflectance (Jensen, 1996).

In the end it might be even more valuable to assess the performance of the algorithm under the more complex situation of taking into account atmospheric influences.

3.5.2 Geometric Correction

3.5.2.1 Transformation of Pixel Coordinates

A matrix of "empty" pixels that conforms to the geometry of the desired output map projection is created. This is done through the collection of *ground-control points* (GCPs) evenly distributed over both the source file (e.g. geocoded image or geocoded vectors) and the uncorrected image (Figure 3.3).

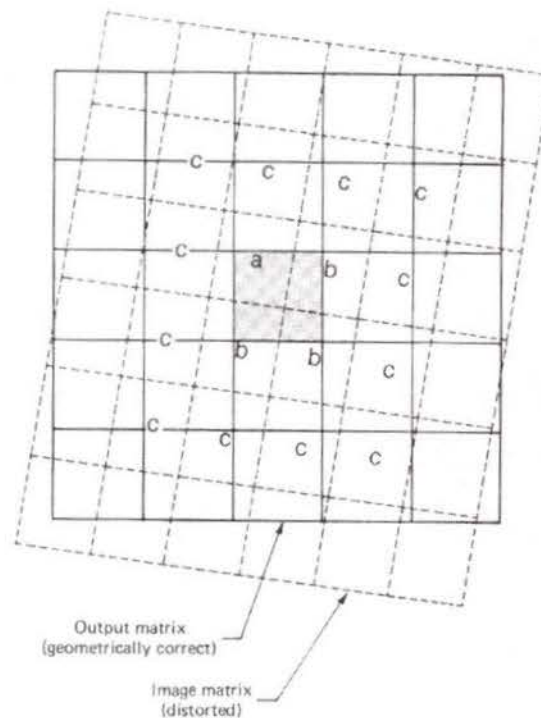


Figure 3.3 – Geometric correction and nearest neighbour resampling. Image matrix is the original image to be geocorrected. Output matrix is the resulting geocorrected image, whose pixel GLVs have not been modified because the nearest neighbour resampling. As seen here, the shaded pixel is assigned the GLV of pixel “a” (Modified from Lillesand *et al.*, 1994).

Different mathematical models are used to transform (*rotate, translate, and scale*) the new matrix of pixels. The most used method is the *polynomial transformation*, which describes mathematically how the original uncorrected image has to be warped to make it fit over the new, corrected matrix of pixels (Figure 3.4) in a process known as *registration*. The model uses the least-squares method to determine the degree of GCPs displacement (positional error) between the uncorrected and geocoded datasets. It then generates coefficients for a two-dimensional coordinate system that will be used to relate the uncorrected image to the final dataset projection.

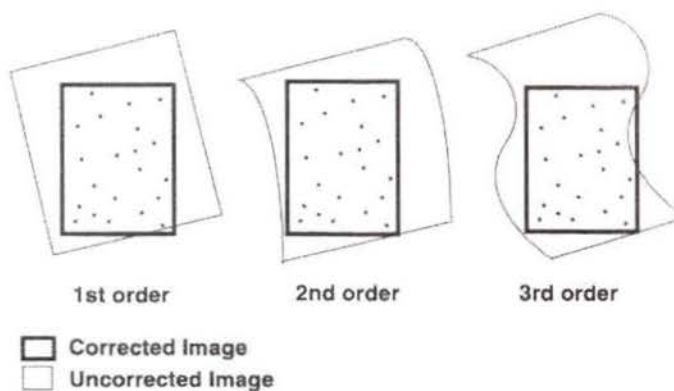


Figure 3.4 – Influence of polynomial order on the distortion of geocoded images (Modified from Lillesand *et al.*, 1994).

The number of GCPs collected will dictate the order of the polynomial transformation. As an example, at least seven GCPs are required for a second-order polynomial transformation (Equations 3.1 and 3.2) in the GCPWorks program from PCI Geomatics Enterprises, Inc. (Richmond Hill, Ontario) (PCI, 1998).

$$x' = a_0 + a_1x + a_2y + a_3xy + a_4x^2 + a_5y^2 \quad (3.1)$$

$$y' = b_0 + b_1x + b_2y + b_3xy + b_4x^2 + b_5y^2 \quad (3.2)$$

where

x' and y' are the coordinates in the uncorrected image, and

x and y are the coordinates in the geocoded dataset.

The polynomial transformation model creates a new geocoded image whose pixels will be filled in with interpolated values during the next stage.

3.5.2.2 Resampling

This technique is used to assign the appropriate digital number (DN) to an output pixel by extracting and interpolating GLVs from the original uncorrected image into the new, corrected image (Figure 3.3). Resampling is used when registering (or overlaying) several images of the same scene taken on different dates, registering images of different resolutions or registering raster data and other data types in a *geographical information system* (GIS).

The simplest resampling method used by most applications is the *nearest neighbour* because of its speed and efficiency and also because it does not alter the GLVs of the resampled pixels (Figure 3.3). The pixel values of the original (uncorrected) image are assigned according to the pixel (in the corrected image) closest to the input coordinate.

The KFA-1000 photograph was geocoded to existing shoreline TRIM vectors (see Section 3.4.2). A second-order polynomial equation and the nearest neighbour resampling method were used in order to try to minimise the geocoding errors to a maximum of one pixel in each X- and Y-direction of all images. This decision was based on the fact that the KFA-1000 image has the second finest spatial resolution of all images analysed (Table 3) and that it would be extremely difficult to collect points in images with lower

spatial resolution in order to achieve the same level of accuracy. By trying to keep the geocoding errors to that standard all images would have a maximum positional displacement of no more than its own spatial resolution unit in each direction.

Subsequently, the KFA-1000 was used to geocode all other images. This was chosen as the best way to avoid the addition of geometric inconsistencies by using the same vectors over images with very different spatial resolutions.

The new large image files (ranging from approximately 600 MB to 2 GB) were then subset into smaller areas to accommodate memory requirements of the extraction software and also to minimise the number of different shoreline types per image.

Even though the shoreline extraction algorithm is completely insensitive to the imagery georeference system, geometric correction was still performed in order to display the results with proper georeferencing grids and to allow the comparison with TRIM shoreline vectors (Appendix C).

3.5.3 Regarding Spatial Resolution or Which Sensors Should Be Used?

Initially, low-resolution images (from Table 3: Landsat 7, Spot 1 and Spot 3) were used to assess the speed and accuracy of the algorithm. The extracted shorelines were then compared to the TRIM vectors (TRIM, 1992).

The advantage of using low-resolution images is the fact that by decreasing the spatial resolution from 10 m to 20 m, the image size (in number of pixels) will decrease four times (or 100 times when decreasing the resolution from 2 m to 20 m).

Consequently, these images contain pixels that encompass larger areas, requiring smaller amounts of memory for storage and processing (Figure 3.5).

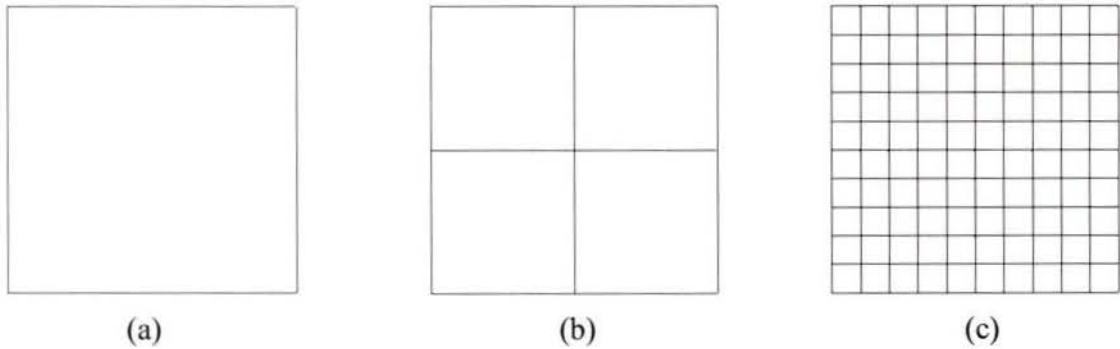


Figure 3.5 – Resolution comparison: (a) original dataset composed of one 20-m pixel; (b) original dataset resampled to 10-m pixels requires four times more memory storage; (c) original dataset resampled to 2-m pixels requires 100 times more memory storage.

On the other hand, the reduced spatial resolution does not provide as much detail. Features smaller than the spatial resolution of the sensor are not picked up individually, thus brightness values are averaged with the brightness values of all other features on the area covered by every pixel.

High-resolution images (from Table 3: Aerial Orthophotos and KFA-1000 photograph) offer a greater challenge for the shoreline extraction. The amount of detail and information are oftentimes difficult to overcome. Still, this research focussed primarily on the extraction of shorelines from high-resolution images in order to obtain the most accurate shoreline vectors as possible.

3.5.4 Regarding Spectral Resolution or Which Bands Should Be Used?

Sensors with several spectral bands (*multispectral*) are of special interest because they provide the same spatial information as viewed at different parts of the electromagnetic spectrum. This allows for the extraction of more information from the same features on different image bands.

A human analyst can identify several different features when looking at any remotely-sensed image. It is fairly easy to pick out the shoreline because the contrast between land and water is usually very high. Water bodies have a very particular quality of absorbing all wavelengths in the near infra-red part of the spectrum (approximately from 0.7 μm to 1.3 μm) giving them a very dark tone (dark pixels represent low values of reflected brightness). Therefore, the use of reflected infra-red bands is often preferred over visible (0.4 μm to 0.7 μm) or thermal/emitted infra-red (8 μm to 13 μm) bands when tracing shorelines.

Infra-red bands from the low-resolution images, mainly Landsat 7 (30 m) and Spot 1 (20 m), were then used. As for the high-resolution images, they are all panchromatic images. This means that all wavelengths within the visible light part of the electromagnetic spectrum were acquired in only one image band (CCRS, 2002; KODAK, 2002). It might be a disadvantage to use panchromatic images for identifying the shoreline but at this high spatial resolution, only such images are available.

The reason is that images with high spatial resolution have a very small *instantaneous field of view* (IFOV). A small IFOV yields a signal that is far smaller than the background electronic noise of the system. Hence, the *signal-to-noise ratio* (SNR) is

very low. In order to maintain a high spatial resolution, it is necessary to sacrifice spectral resolution by broadening the wavelength band of the remote sensing system, thus increasing SNR (Lillesand *et al.*, 1994).

3.5.5 Enhancements and Filters

As discussed in Chapter 2, different enhancements and edge detectors were used depending on the image characteristics and coast type. Basically, a linear stretch (without saturation) or histogram equalisation were applied on the images whenever their histograms were confined to a narrow interval of GLVs.

3.5.6 Feature Image

The purpose of enhancing and filtering was to create an image that would contain only information on shoreline features (Pingle, 1969). This *feature image* (FeIm) would then be fed into the tracing algorithm for the extraction of the shoreline. The main point to note here is the establishment of criteria for designing the most suitable FeIm for each coast type. Thus, different smoothing and edge detection filters were applied on all images, trying several different combinations that would minimise the amount of tracing time in the next stage. It is important to note that edges in FeIm must be dark (GLVs usually below 128) because the evaluation function is minimised (Appendix A). Therefore, edge-detected images are negated using equation 3.3.

$$GLV' = 255 - GLV \quad (3.3)$$

where

GLV' \equiv new GLV in the negated image; and

GLV \equiv original digital number.

The process of generating FeIm is controlled by two factors vital to the success of the shoreline extraction:

- FeIm must be fast enough to enable the image analyst to rely on it and not employ commonly-used manual tracing of shorelines; and
- FeIm must be accurate enough to minimise the search time of the algorithm.

In addition to the filters presented in Chapter 2, histogram thresholding and density slicing were also performed on certain images. Sometimes edge-detected images are not smooth enough for a fast shoreline tracing. By thresholding these filtered images, the performance of the tracing algorithm is often increased because of the reduction in GLVs, thus decreasing radiometric variation.

3.5.7 Shoreline Tracing

Martelli (1972) devised an algorithm to detect edges in images based on the seminal works of Kruskal (1956) and Dijkstra (1959). Dijkstra's *uniform cost algorithm* finds the shortest path from a start node to an end node in $O(N^2)$, where O describes the computational complexity of the algorithm and N equals the number of pixels (or nodes) in an image. Thus, given an image dataset of size N , the algorithm may have to be run N^2 times (Dillabaugh, 2000), implying that the smaller the image the faster the algorithm will be processed.

The improvement given by Martelli to Dijkstra's algorithm comes from the use of *heuristic functions*. The A^* heuristic algorithm devised by Nilsson (1971, 1980) applies perfectly to the edge-tracing problem in that instead of searching for the optimal least-cost path in an image it only looks for a "good" solution. This ensures that only the most promising edges are selected for the search and that unpromising edges are discarded. Thus, processing time is greatly reduced.

Basically, Martelli's algorithm is simple. A feature image $FeIm$ depicting only promising shoreline edges is created from an original image. At least two points are marked along promising edges, namely *start* and *end* points. Intermediate points can be selected to guide and speed up the search, if necessary, this being intimately dependent on the quality of the edges on $FeIm$. Here is where a trade-off lies: Well-defined edges require fewer points but it might increase the search time, whereas more points decrease search time. At the same time, we are trying to avoid as much as possible user interaction

in order to speed up the whole process. The increase in points might turn out to be as cumbersome as tracing the shoreline manually.

The algorithm then checks the cost of edge elements. An edge element (or *edgel*) consists of a pair of neighbouring pixels. Each edge element has a cost associated with it. This cost is basically an evaluation function (from there the use of the name cost image, or *Felm*, in our case) that determines whether that edge element should be expanded or not. The cost function evaluates the difference between neighbouring pixels of an edge element. In order to ensure that the cost function is always greater than or equal to zero, it is subtracted from the absolute value of the maximum difference between two adjacent points in the image. This value is then added to the Euclidean distance between two consecutive edge elements so to minimise the distance between the current edge element and the end node (Appendix A). An edgel can be expanded in three directions depending on its orientation (Figure 3.6) and it will lead to the evaluation of the following edgel until the end node is reached or until a stopping criterion is fulfilled.

Figure 3.7 shows how Martelli's search algorithm works, assuming that start and end nodes are located at (X_4, Y_5) and (X_6, Y_2) , respectively, on this small raster grid. Because the search for edges with lowest cost is counter-clockwise, the first edgel is located at $(X_4, Y_5)(X_3, Y_5)$. And because the X-coordinate in (X_4, Y_5) is larger than the X-coordinate in (X_3, Y_5) , the coordinate difference dx is positive. Thus, the costs of edge elements $(X_3, Y_4)(X_3, Y_5)$, $(X_4, Y_4)(X_3, Y_4)$ and $(X_4, Y_5)(X_4, Y_4)$ are evaluated. The algorithm continues to evaluate the costs of expanded edge elements until it reaches the end node. Consequently, an edge separating elements with value of seven from elements

with value of two is traced. The node expansion and final edgel costs are summarised in Table 4.

The algorithm used in this research offers two different possibilities that could be used in conjunction with each other or independently to stop the process. The limiting factor of the search could be the number of nodes expanded and/or elapsed time. In this particular case, time should be set to no more than 10 seconds in order to try to achieve the best results in a very short time. The number of expanded nodes will not be limited, allowing the software to expand as many nodes as possible in that timeframe.

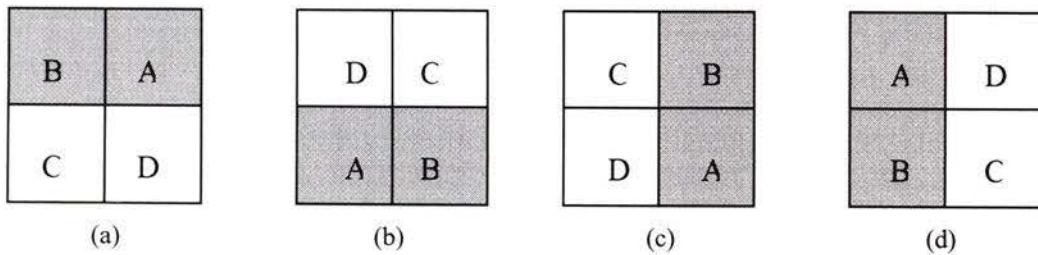


Figure 3.6 – Edge Element Expansion Rationale. Initial edge element AB can be expanded onto three different edges, BC, CD or DA depending on the orientation of the search on the Cartesian Coordinate System. (a) $dx > 0$; (b) $dx < 0$; (c) $dy > 0$; (d) $dy < 0$, where dx and dy are the differences in position between the farther and the closer coordinates.

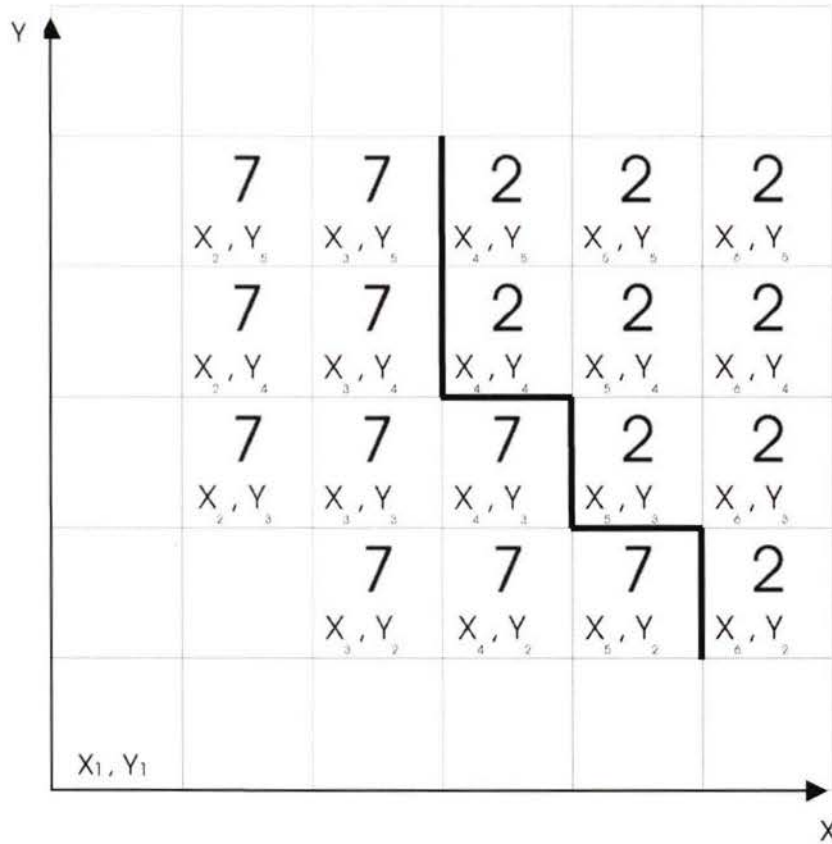


Figure 3.7 – Martelli search algorithm on a subset of an image dataset.

The search occurs in FeIm, being guided by the points entered by the user in the original image. This allows choosing the most promising shoreline in the original image. The user can also view the search being carried out interactively on FeIm. This feature provides some extra knowledge about the quality of FeIm and user-entered points. Ultimately, the search is done when the end node or the maximum search time is reached. The resulting shoreline vector is drawn on the original image and a corresponding list of points (with identification number, and X- and Y-coordinates) is generated.

Table 4
Node Expansion and Final Node Costs of the Search in Figure 3.7

	Local Cost Function	Cost Function (Maximum Difference – Local Cost)	Edge Element and Final Cost
dx > 0	(X ₃ , Y ₅)-(X ₃ , Y ₄) → 7-7 (X ₃ , Y ₄)-(X ₄ , Y ₄) → 7-2 (X ₄ , Y ₄)-(X ₄ , Y ₅) → 2-2	5-0 5-5 5-0	(X ₃ , Y ₄)-(X ₄ , Y ₄) → 0
dx > 0	(X ₃ , Y ₄)-(X ₃ , Y ₃) → 7-7 (X ₃ , Y ₃)-(X ₄ , Y ₃) → 7-7 (X ₄ , Y ₃)-(X ₄ , Y ₄) → 7-2	5-0 5-0 5-5	(X ₄ , Y ₃)-(X ₄ , Y ₄) → 0
dy < 0	(X ₄ , Y ₃)-(X ₅ , Y ₃) → 7-2 (X ₅ , Y ₃)-(X ₅ , Y ₄) → 2-2 (X ₅ , Y ₄)-(X ₄ , Y ₄) → 2-2	5-5 5-0 5-0	(X ₄ , Y ₃)-(X ₅ , Y ₃) → 0
dx > 0	(X ₄ , Y ₃)-(X ₄ , Y ₂) → 7-7 (X ₄ , Y ₂)-(X ₅ , Y ₂) → 7-7 (X ₅ , Y ₂)-(X ₅ , Y ₃) → 7-2	5-0 5-0 5-5	(X ₅ , Y ₂)-(X ₅ , Y ₃) → 0
dy < 0	(X ₅ , Y ₂)-(X ₆ , Y ₂) → 7-2 (X ₆ , Y ₂)-(X ₆ , Y ₃) → 2-2 (X ₆ , Y ₃)-(X ₅ , Y ₃) → 2-2	5-5 5-0 5-0	(X ₅ , Y ₂)-(X ₆ , Y ₂) → 0

The above algorithm devised by Martelli was chosen because of its ease to use, simplicity and quality of results. The whole process flows quickly and requires very basic image processing techniques for both the creation of the evaluation image and understanding of the tracing algorithm. Lyon (1999) adapted Martelli's algorithm and implemented it in Java™ (Appendix A).

4. Results

4.1 Control Images

Control images with GLVs of 0 (black) and 255 (white) and a resolution of 38 pixels/cm (or 97 dots per inch (DPI)) were created to test how the tracing algorithm performed under ideal-edge conditions. Three different geometric shapes, a circle, a square, and a triangle (Figure 4.1) were used to determine the algorithm's ability to trace different angles.



Figure 4.1 – Control Images

Edges were detected using the operators described in Chapter 2: Roberts, Sobel, Prewitt, Laplacian, Frei-Chen, Canny, and Pixel Difference. The images were then negated and their boundaries were traced. The number of points used and processing times are shown in Table 5. The Canny operator was modified to accelerate processing

Figure 4.2 shows an example of a line traced by the algorithm. This result is consistent with the quality of the Feature Image FeIm. The simple edge detection and negation operations performed in reasonably low processing times for images with similar dimensions (Table 6).

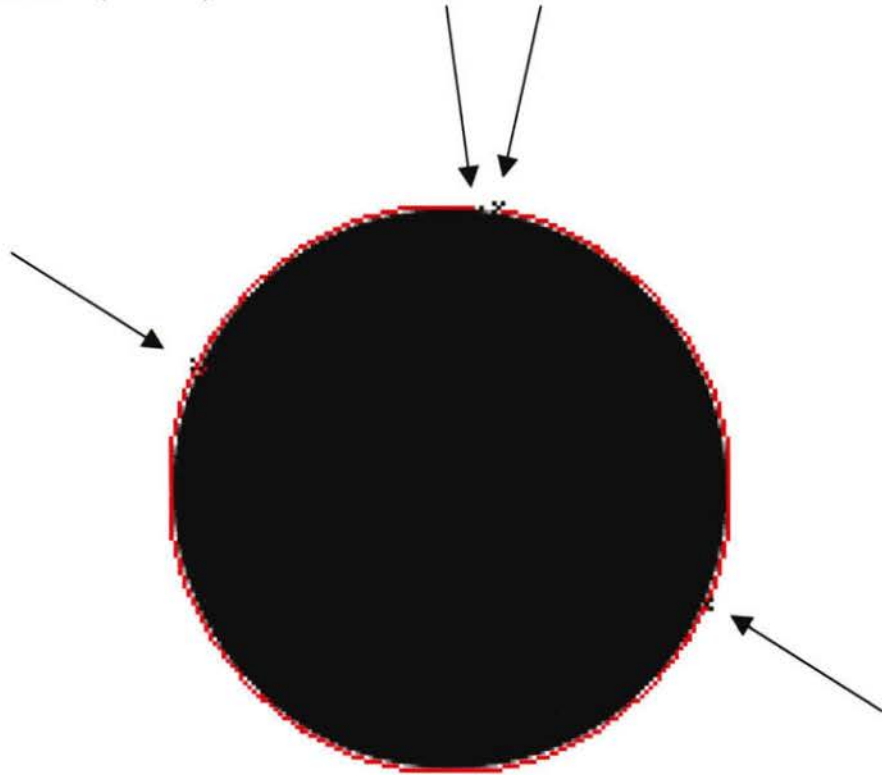


Figure 4.2 – Enlarged control image. The red line is the boundary traced by the algorithm and the X marks represent seed-points. (The Frei-Chen edge operator was used.)

Table 6
Characteristics of Control Images

	Number of pixels (GLV=0)	Perimeter (pixels // cm)	Area (cm ²)
Circle	13,097	406 // 10.68	9.07
Square	11,881	436 // 11.44	8.18
Triangle	14,469	554 // 14.60	10.02

4.2 Shoreline Feature Extraction

For the purpose of testing the algorithm's ability to delineate different shorelines, the original images were subset into smaller areas featuring different coast types, and different spatial and spectral resolutions. Appendix C shows an overview of the study area with outlines of the chosen subsets. Four different major coast types that best represent the area were chosen based on the diversity and complexity of these environments (see Chapter 2):

- Coasts with vegetation line: only water and forest are visible;
- Sandy beaches: water, beach and forest are visible;
- Mudflats/shallow areas: water, mudflat and forest are visible; and
- Rocky coasts: water, rock outcrops and forest are visible.

The image processing and tracing algorithm are summarised in Appendix D.

4.2.1 Low-Resolution Images

Images acquired by Landsat 7, SPOT 1, and SPOT 3 were grouped as low-resolution images due to the fact that ground features are not as clearly discernible as in the aerial orthophotos and KFA-1000 photograph. In general, low-resolution images required a very simple tracing scheme for extracting the shoreline. This is due partly to the use of panchromatic and infra-red images which display a pronounced contrast between land and water (explained in Section 3.5.4). Shorelines on these images are

easily detected and creating FeIm and performing the shoreline extraction can usually be achieved in just under the maximum search time set in the algorithm (explained in Section 3.5.7). Table 7 summarises the most efficient rationale for creating FeIm for each of the four different environment groups. Note that operators must be used in the order shown for each coast and image types.

Table 7

FeIm Rationale for Low-resolution Images

	Coasts with Vegetation Line	Sandy Beaches	Mudflats/Shallow Areas	Rocky Coasts
SPOT 3 (10 m)	Sobel-Negate-Equalisation-Enhancement-Average	Prewitt-Negate	Equalisation Enh.-Gaussian 7x7-Sobel-Negate-Roberts-Negate-Average	Gaussian 7x7-Roberts-Negate-Average
Landsat 7 (15 m)	Prewitt-Negate-Average	Prewitt-Negate	Roberts-Negate	Sobel-Negate
SPOT 1 (20 m)	Prewitt-Negate-Average	Prewitt-Negate-Average	Roberts-Negate-Average	Roberts-Negate
Landsat 7 (30 m)	Sobel-Negate-Average	Roberts-Negate-Average	Roberts-Negate	Prewitt-Negate-Average

4.2.1.1 Coasts with Vegetation Line

Coasts with vegetation lines usually have high contrast between land and water. Thus, the shoreline on those images was visible and despite the low resolution it could even be traced manually without major difficulty. Nevertheless, the accuracy of manual tracing at this scale is questionable. Edge operators easily found the shoreline with seed-points located at main features. At this coarse resolution the vegetation line is not

distinguishable, hence the accurate positioning of seed-points is the deciding factor for the quality of shoreline tracing. The number of seed-points used depends on the quality of FeIm and the length of the shoreline.

The best shorelines were obtained by first applying an edge operator and then a smoothing filter. The Roberts, Prewitt, and Sobel operators consistently performed faster than the others. A simple 3 x 3 averaging filter (see Template 2.24) was applied on the negated edge-detected image to smooth jagged edges. Smoothing the image was also tried before edge detection took place, but the results were not nearly as good. Figures 4.3 to 4.6 show the resulting extracted shoreline after smoothing the edge-detected image.

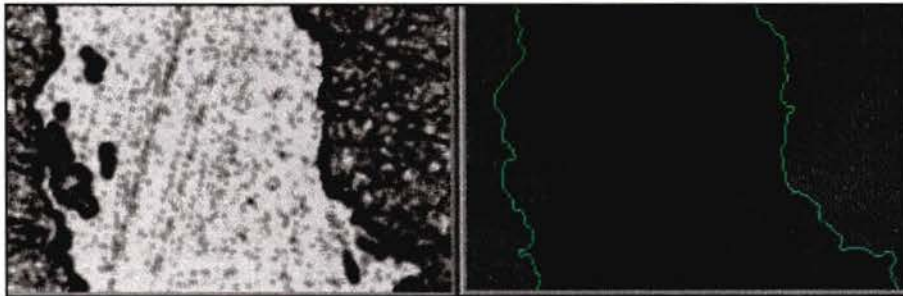


Figure 4.3 – SPOT 3 (10 m). Left: edges detected in 8.3 s using Sobel – Negate – Uniform Enhancement – Average – 6 points. Right: original image and extracted shoreline in green.

Figures 4.7 to 4.9 show the relationship between extracted shoreline vectors and TRIM vectors, and also the 100-m perpendicular shoreline profiles. It is possible to see that the extracted shorelines (in yellow) are very close to the TRIM vectors (in red). The distance between them is about one pixel in the Landsat and SPOT images with the exception of some areas where the gap seems to extend to about four pixels in the SPOT

3 image. Thus, the general displacement is no larger than 40 m (SPOT 3) with overlapping vectors occurring quite often in all images.

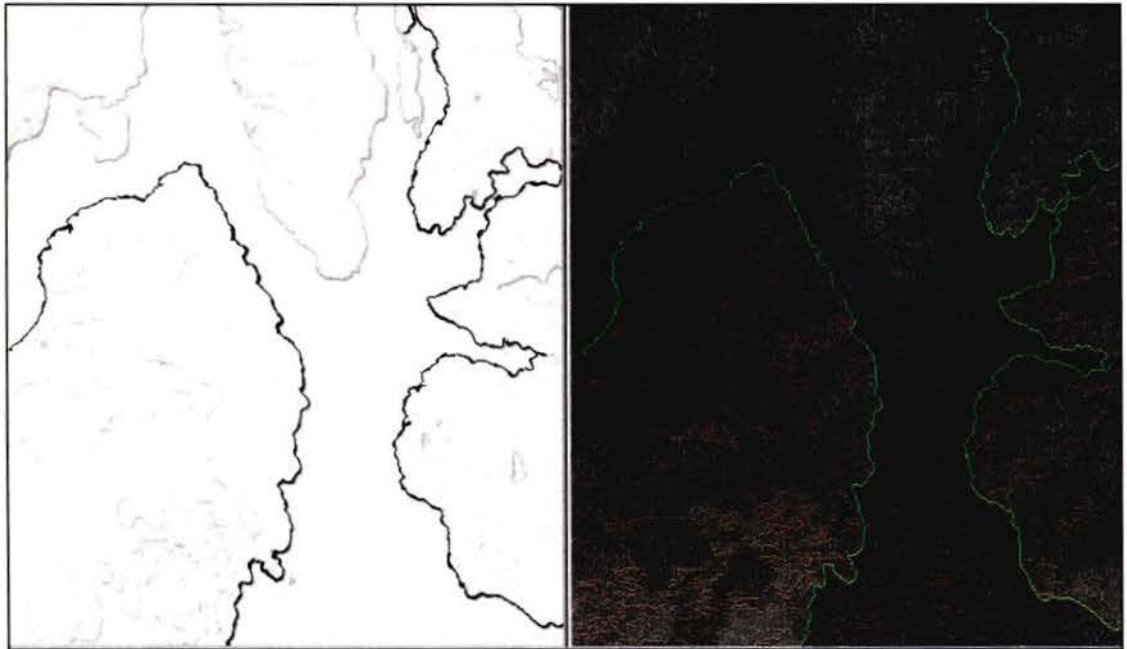


Figure 4.4 - Landsat 7 (15 m). Left: shorelines (in black) detected in 9.8 s using Prewitt – Negate – Average – 24 points. Right: original image and extracted shoreline in green.

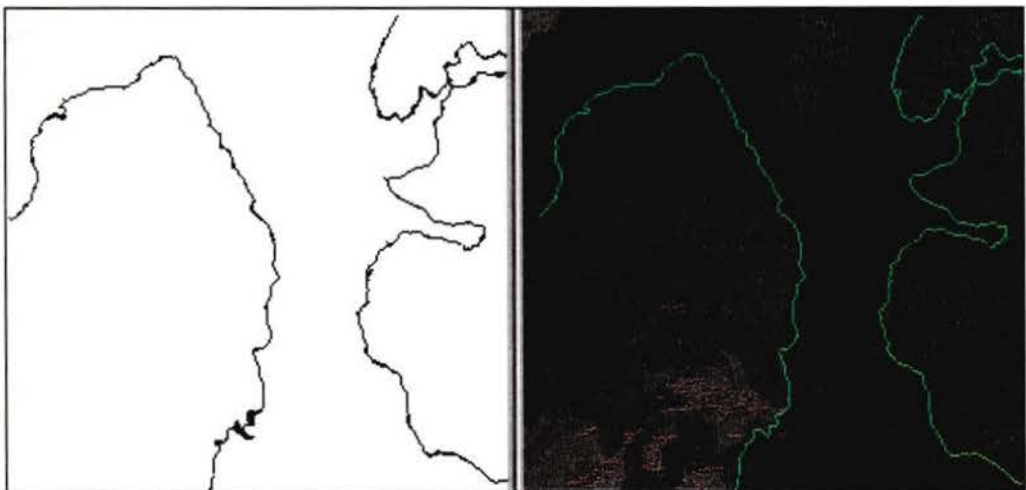


Figure 4.5 – SPOT 1 (20 m). Left: shorelines (in black) detected in 1.5 s using Prewitt – Negate – Average – 18 points. Right: original image and extracted shoreline in green.

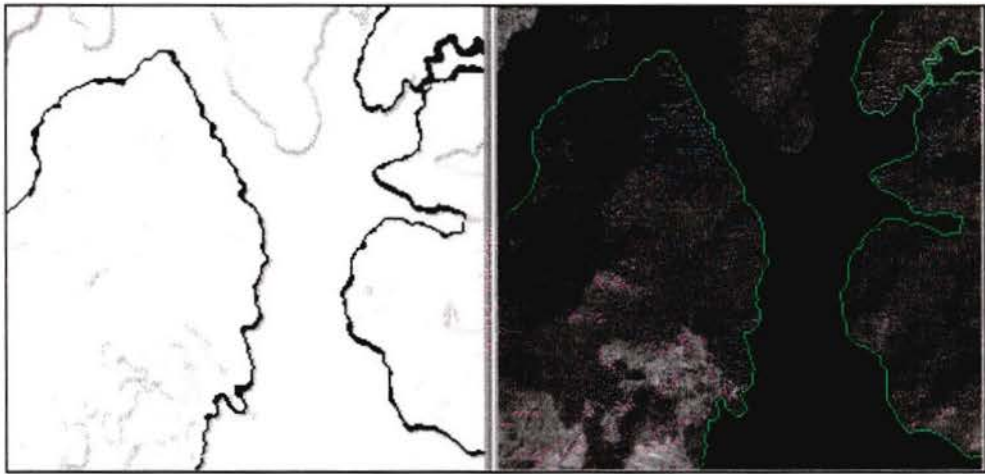


Figure 4.6 – Landsat 7 (30 m). Left: shorelines (in black) detected in 4.5 s using Sobel – Negate – Average – 13 points. Right: original image and extracted shoreline in green.

Edge profiles (Figures 4.10a to 4.10d) show clear differences between land and ocean. Wide edges represent the transition from land features to water. These profiles are a very limited representation of the edge characteristics of these areas but serve the purpose of giving a general idea of the behaviour of the shorelines. It can be seen that the maximum GLV amplitude is found on the Landsat 7 (30 m) image, where land pixel GLVs range from 57 to 75, and water pixel GLVs stay below 20, which is a drop of more than 50 GLVs.

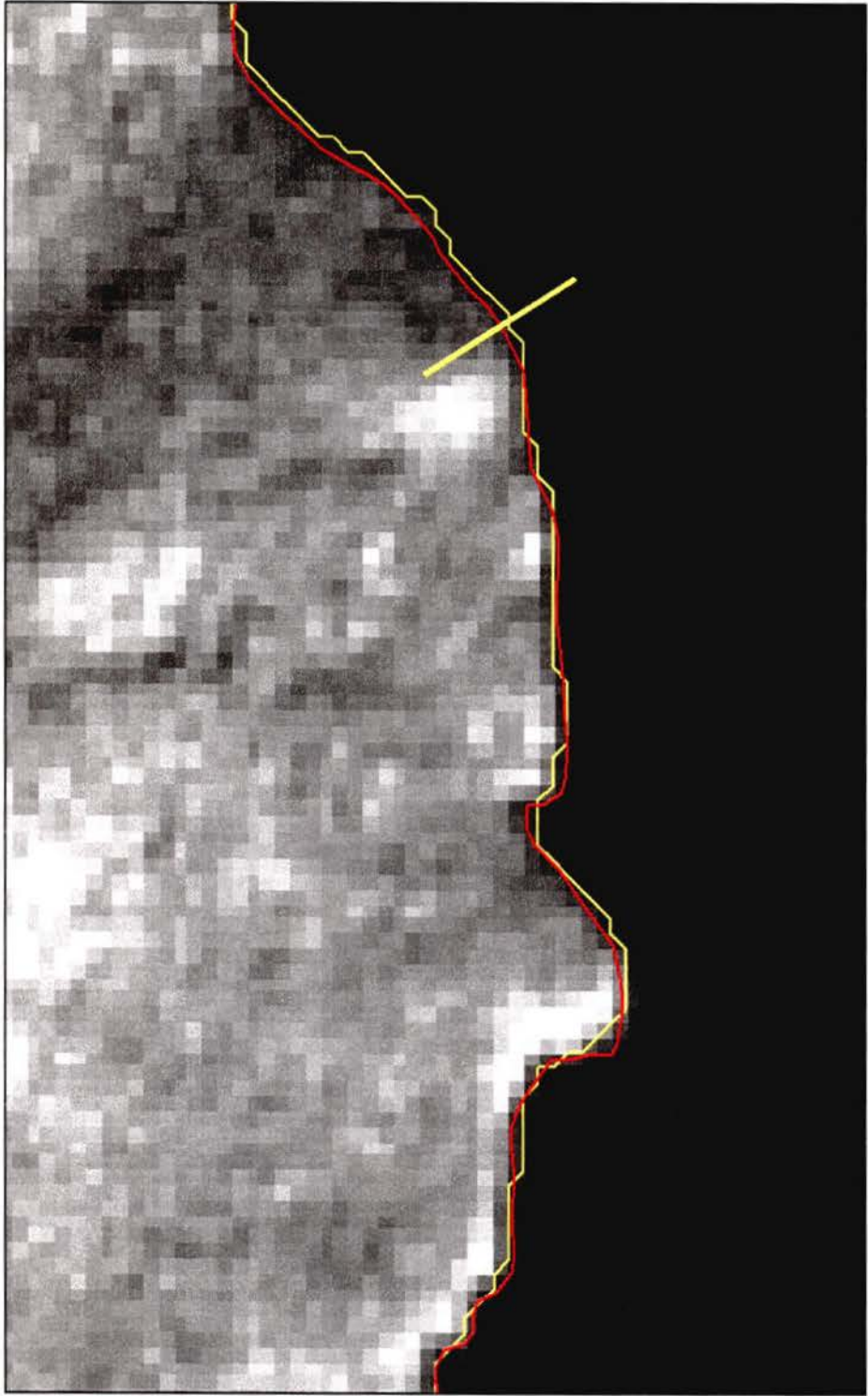


Figure 4.7 – Landsat 7 (15 m): Vector comparison. TRIM vectors are in red and shoreline extracted by the tracing algorithm is in yellow. Shoreline profile (in yellow) is perpendicular to the shoreline.

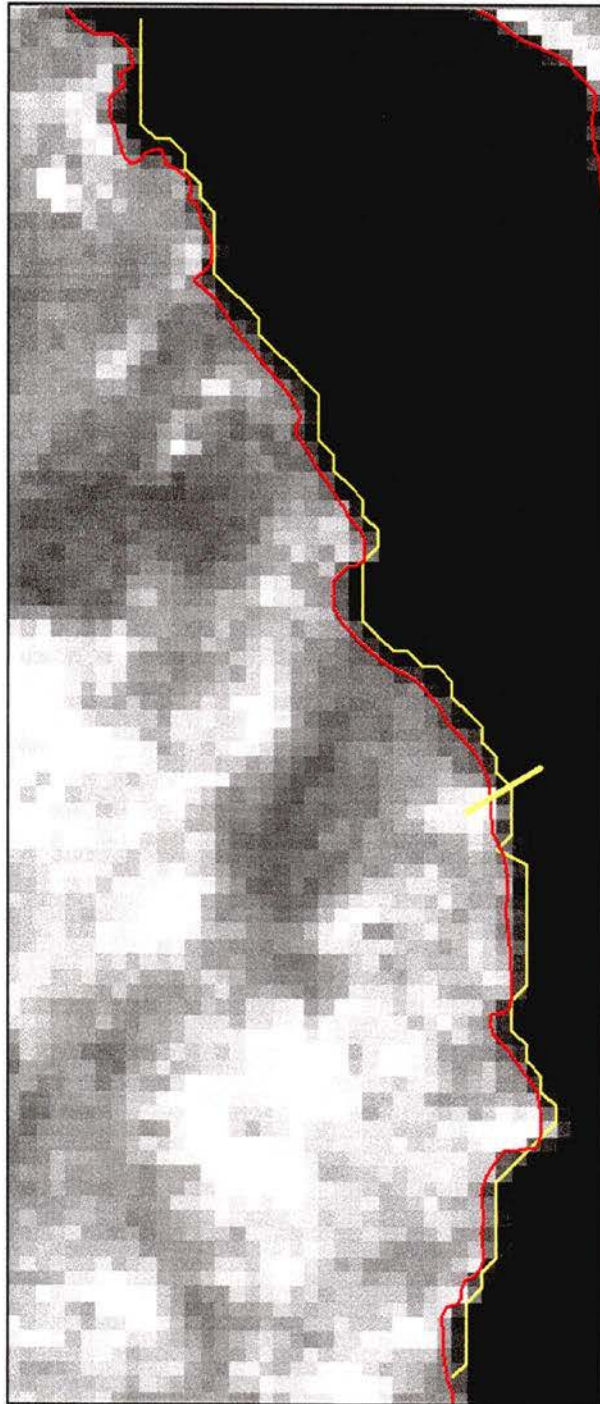


Figure 4.8 – SPOT 1 (20 m): Vector comparison. TRIM vectors are in red and shoreline extracted by the tracing algorithm is in yellow. Shoreline profile (in yellow) is perpendicular to the shoreline.

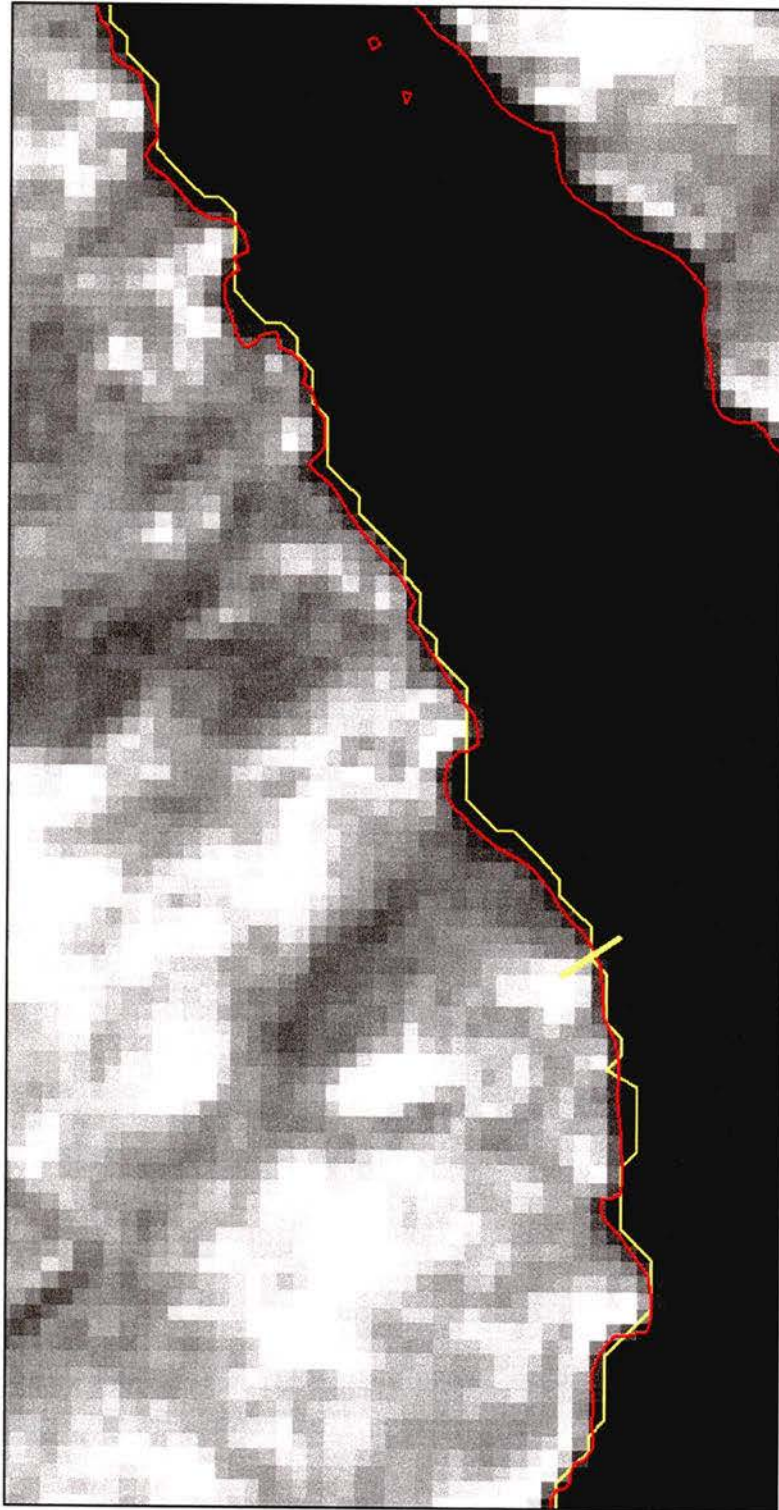


Figure 4.9 – Landsat 7 (30 m): Vector comparison. TRIM vectors are in red and shoreline extracted by the tracing algorithm is in yellow. Shoreline profile (in yellow) is perpendicular to the shoreline.

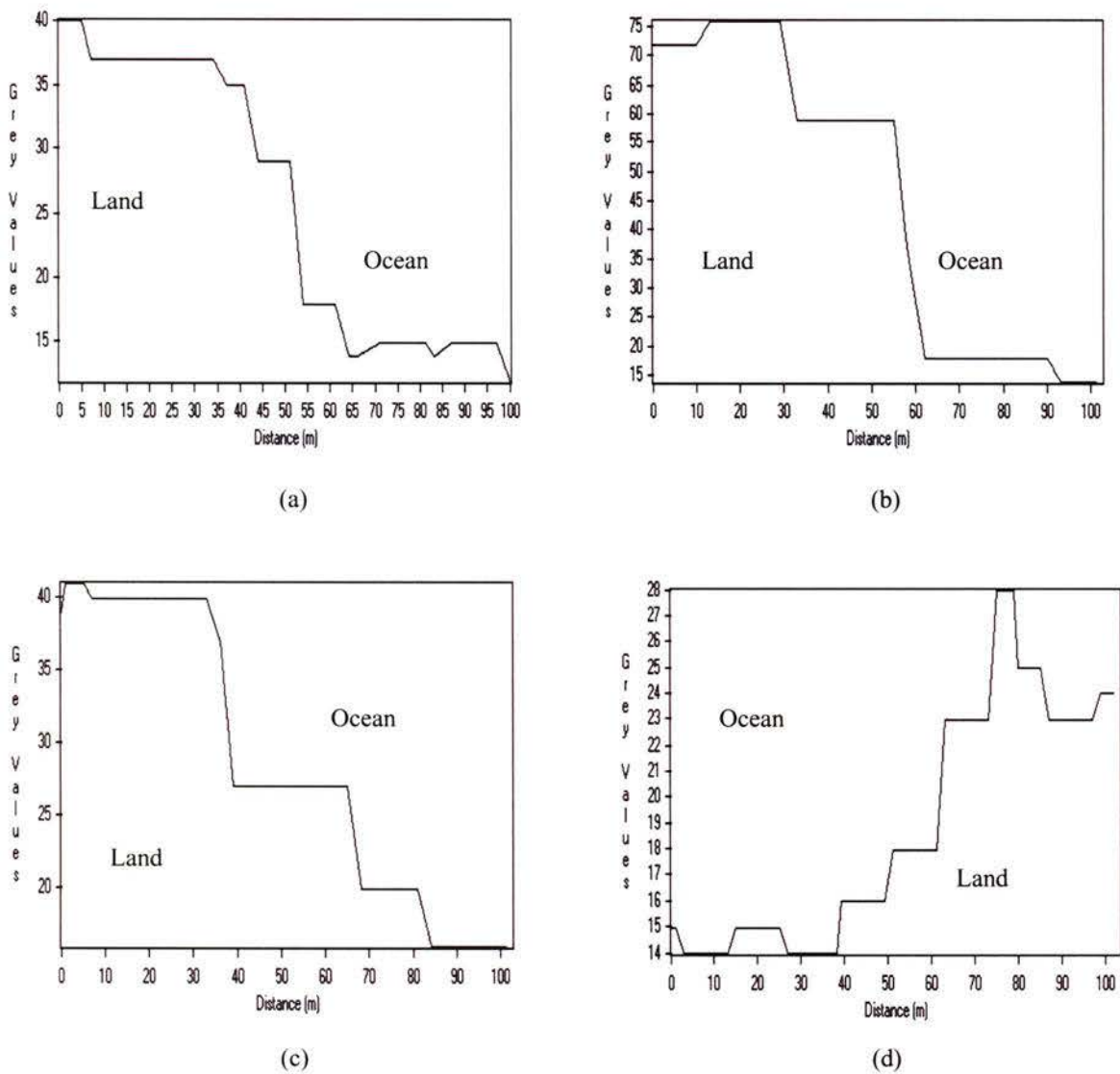


Figure 4.10 – Shoreline GLV profiles for coasts with vegetation line. (a) Landsat 7 (15 m); (b) Landsat 7 (30 m); (c) SPOT 1 and (d) SPOT 3.

4.2.1.2 Sandy Beaches

Just as the Roberts, Prewitt and Sobel filters worked well for detecting the edges of coastlines with vegetation (see Section 4.2.1.1), they also provided the best edges for coastlines with sandy beaches. Once edge detection was performed on the low resolution Landsat 7 (30 m) and SPOT 1 images, a smoothing filter was applied (Figures 4.11 to 4.14).

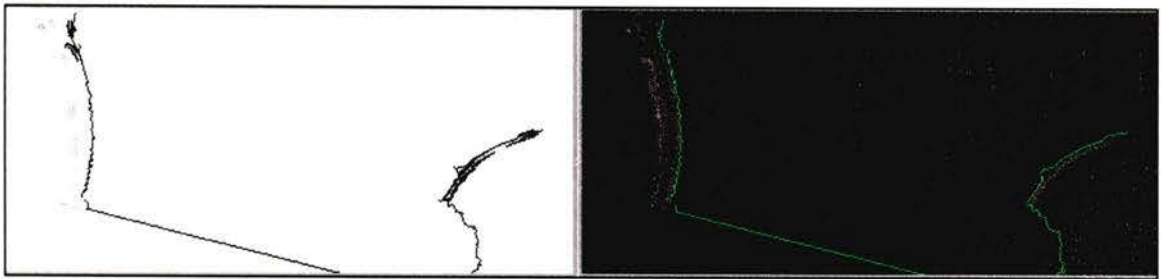


Figure 4.11 – SPOT 3 (10m). Left: shoreline (in black) detected in 1.1 s using Prewitt – Negate – 8 points. Right: original image and extracted shoreline in green.

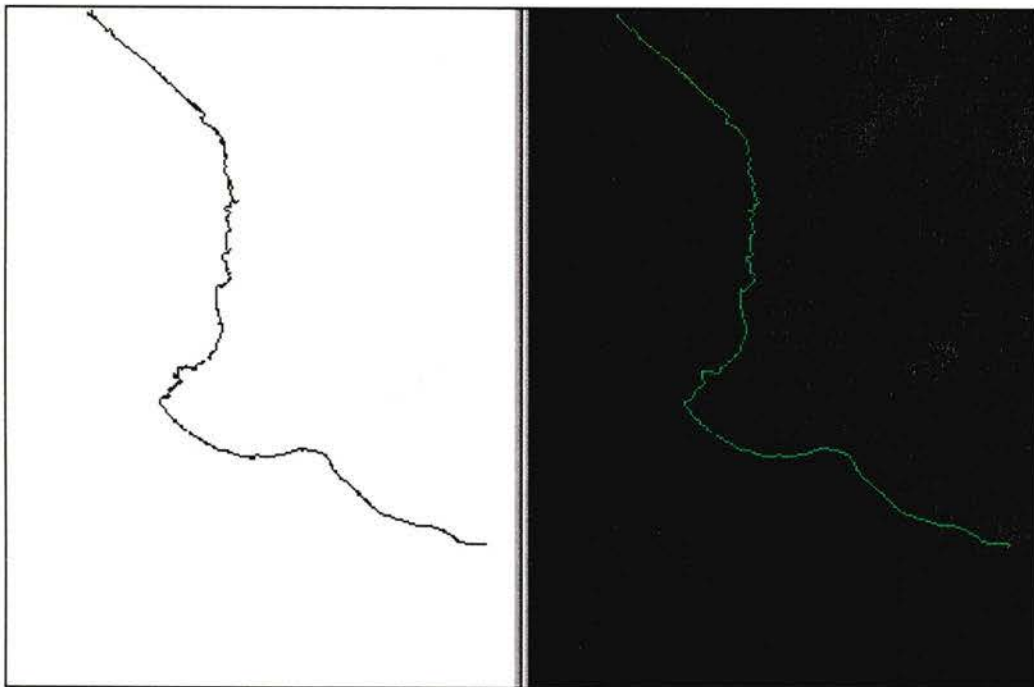


Figure 4.12 – Landsat 7 (15 m). Left: shoreline (in black) detected in 0.97 s using Prewitt – Negate – 6 points. Right: original image and extracted shoreline in green.

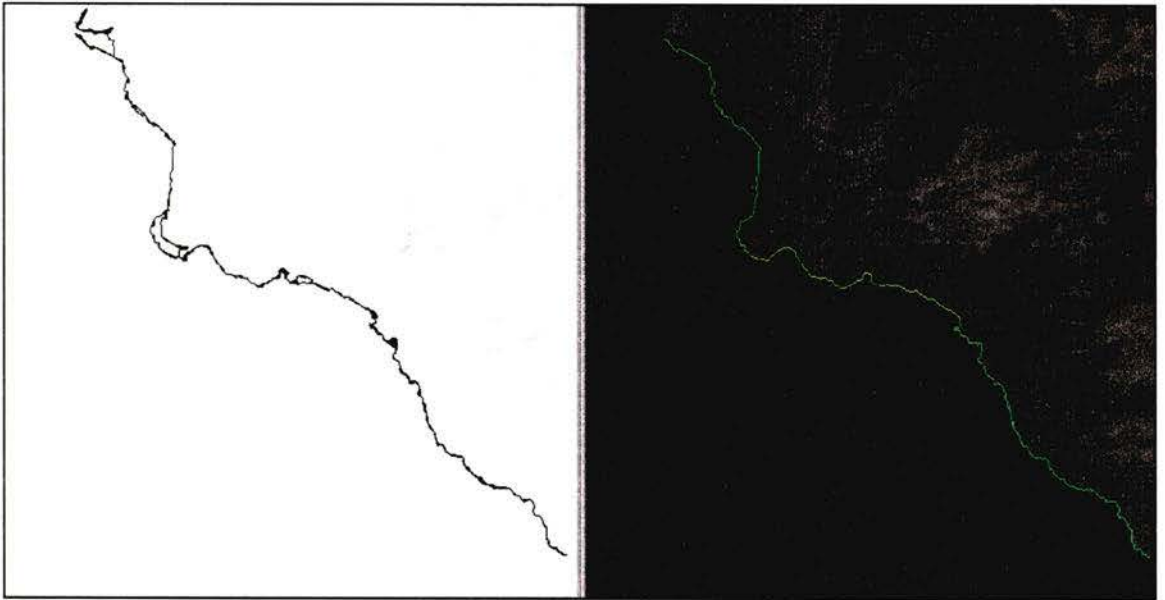


Figure 4.13 – SPOT 1 (20 m). Left: shoreline (in black) detected in 1.3 s using Prewitt – Negate – Average – 8 points. Right: original image and extracted shoreline in green.

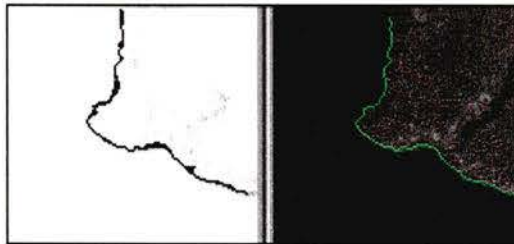


Figure 4.14 – Landsat 7 (30 m). Left: shoreline (in black) detected in 2.9 s using Roberts – Negate – Average - 2 points. Extracted shoreline is in green.

The comparison between extracted shoreline vectors and TRIM vectors can be seen in Figures 4.15 to 4.17. Although the extracted shoreline vectors loosely follow the TRIM vectors it is clear that in some places they do not indicate the exact same location for the shoreline. Maximum displacements range from 40 m on the SPOT 1 image to 60 m on both Landsat 7 images.

NB: The SPOT 3 image was not located in the same area as the other three images, and as a consequence, the positional accuracy of its vectors could not be assessed because there were no TRIM vectors available for the area.

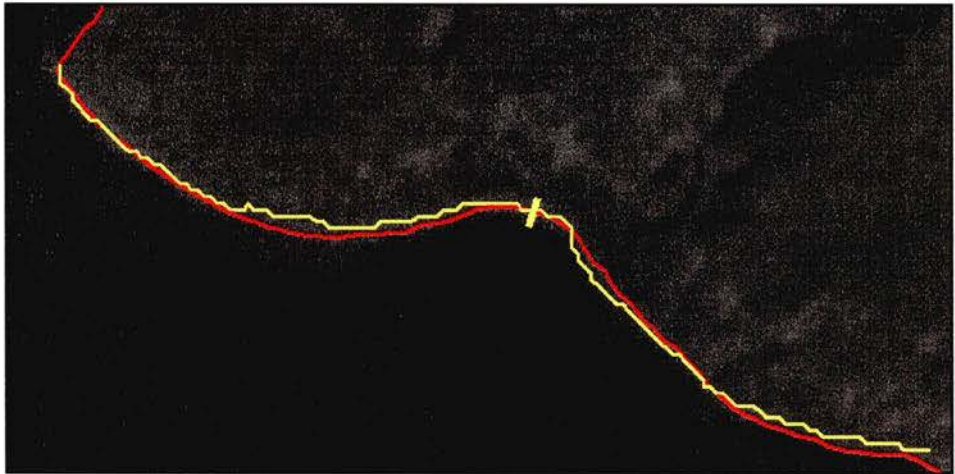


Figure 4.15 – Landsat 7 (15 m). Vector comparison. TRIM vectors are in red and the shoreline extracted by the tracing algorithm is in yellow. Shoreline profile (in yellow) is perpendicular to the shoreline

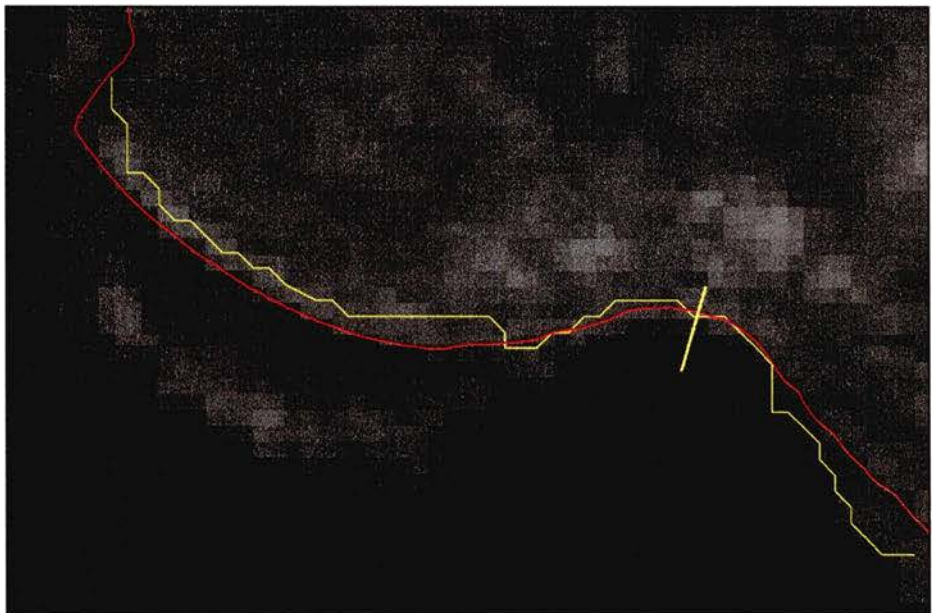


Figure 4.16 – SPOT 1 (20 m). Vector comparison. TRIM vectors are in red and the shoreline extracted by the tracing algorithm is in yellow. Shoreline profile (in yellow) is perpendicular to the shoreline.

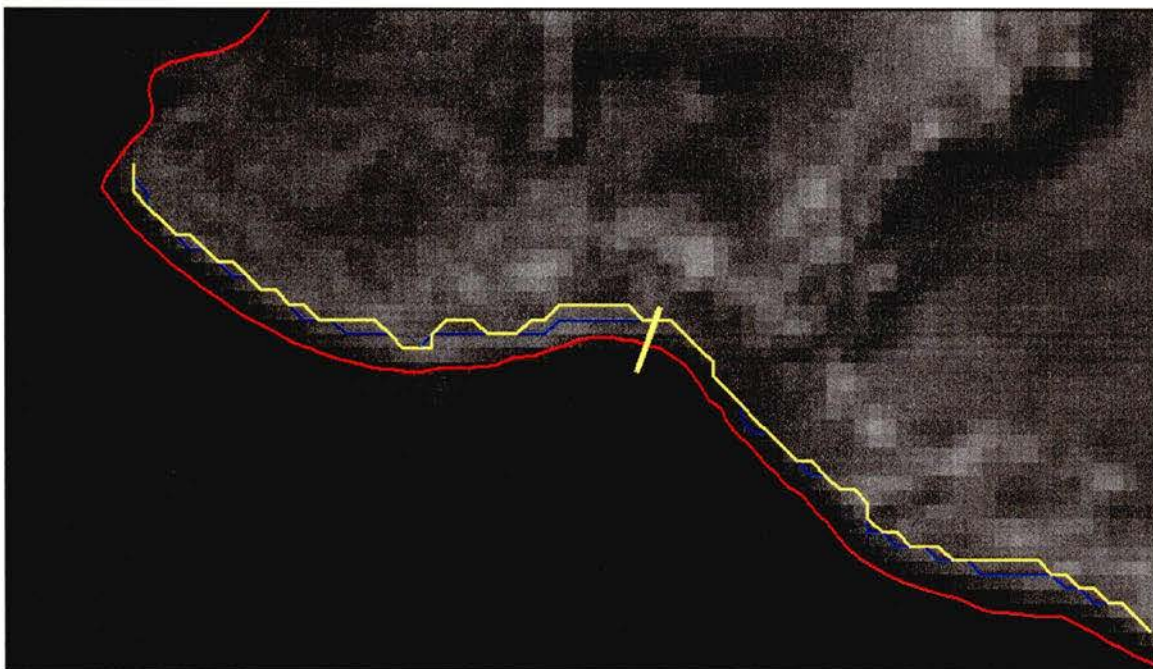
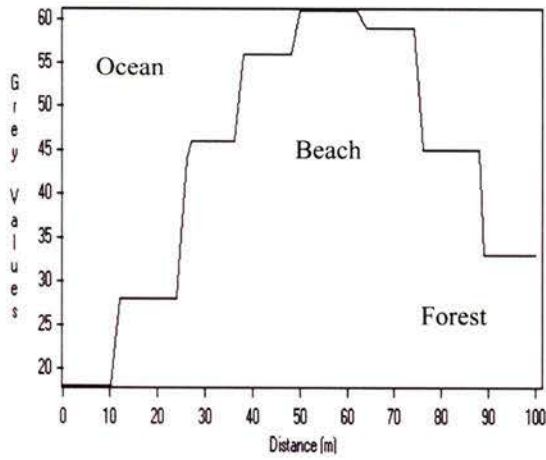
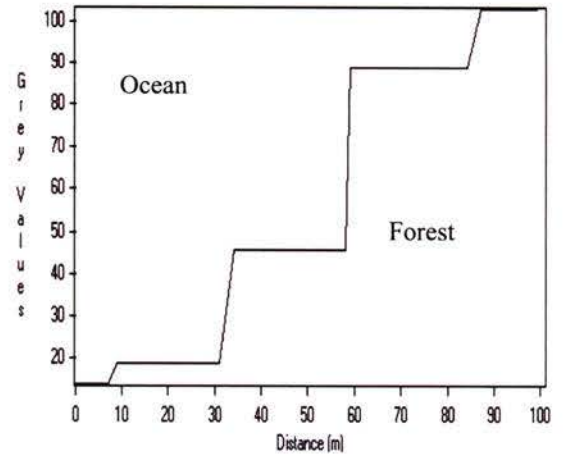


Figure 4.17 – Landsat 7 (30 m). Vector comparison. TRIM vectors are in red and the shorelines extracted by the tracing algorithm are in yellow (Roberts filter) and blue (Sobel filter). Shoreline profile (in yellow) is perpendicular to the shoreline.

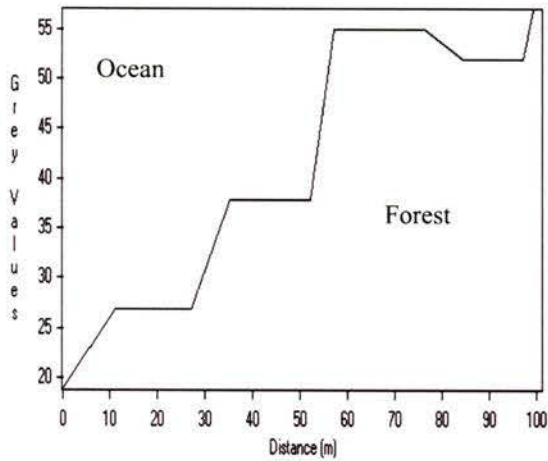
Edge profiles for the area (Figures 4.18a to 4.18d) show a clear contrast between land and ocean again, and particularly the edge between forest and beach on the SPOT 3 image.



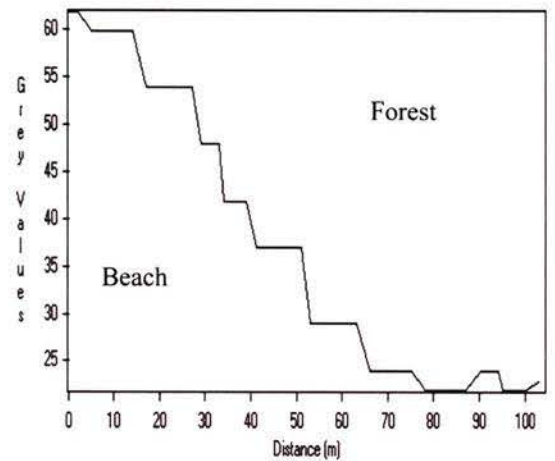
(a)



(b)



(c)



(d)

Figure 4.18 – Shoreline GLV profiles for sandy beaches. (a) Landsat 7 (15 m); (b) Landsat 7 (30 m); (c) SPOT 1 and (d) SPOT 3.

4.2.1.3 Mudflats/Shallow Areas

Mudflats offer a particular challenge to edge detection because the usual high contrast between land and sea is often replaced by a gradient of GLVs extending from the land to sea. Still, since low-resolution images do not usually provide much detail, the contrast is sufficiently high to allow the operators to find shoreline edges with relative accuracy (Figures 4.19 to 4.22).

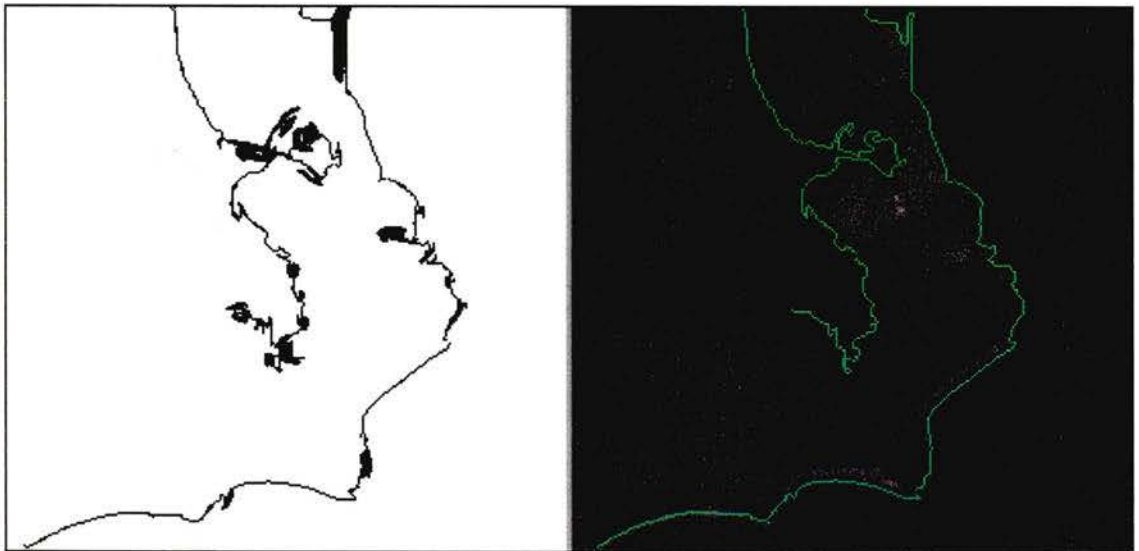


Figure 4.19 – SPOT 3 (10 m). Left: shoreline (in black) detected in 3.98 s using Equalisation Enhancement – Gaussian 7x7 – Sobel – Negate – Roberts – Negate – Average – 15 points. Right: original image and extracted shoreline in green.

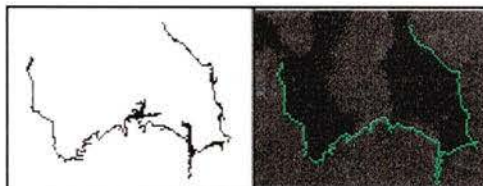


Figure 4.20 – Landsat 7 (15 m). Left: shoreline (in black) detected in 0.44 s using Roberts – Negate – 8 points. Right: original image and extracted shoreline in green.

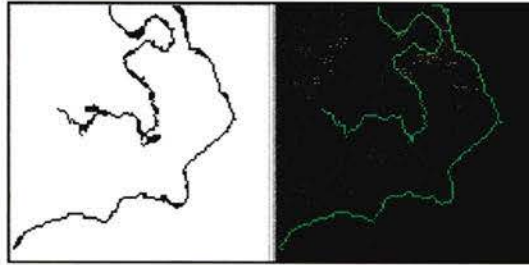


Figure 4.21 – SPOT 1 (20 m). Left: shoreline (in black) detected in 0.55 s using Roberts – Negate – Average - 12 points. Right: original image and extracted shoreline in green.

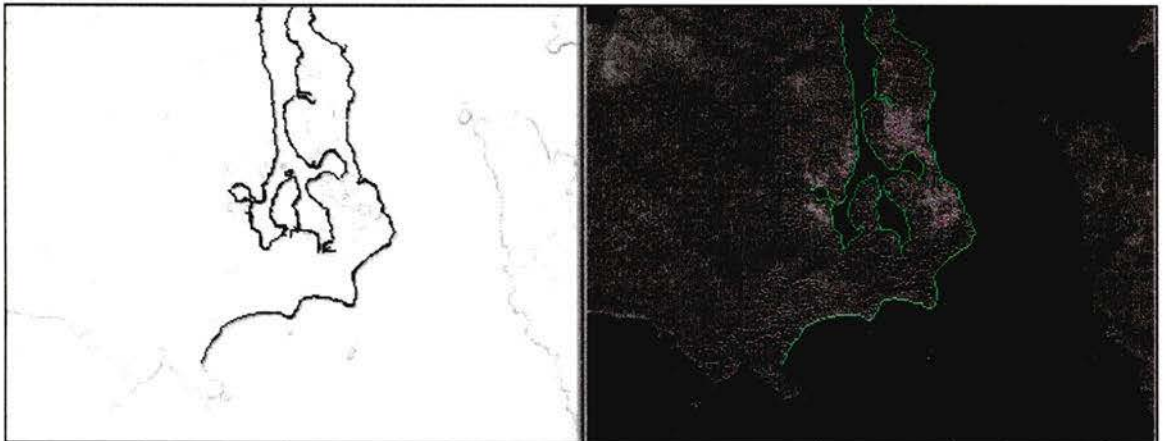


Figure 4.22 – Landsat 7 (30 m). Left: shoreline (in black) detected in 1.72 s using Roberts – Negate – 20 points. Right: original image and extracted shoreline in green.

The shorelines on Landsat 7 images were traced after processing them with the Roberts operator and negating them afterwards. The SPOT 1 image had to be smoothed to reduce the influence of jagged edges. The SPOT 3 image required still more smoothing operations because its higher resolution provided some finer detail. Edge operators did not detect strong edges because the original image had very low contrast. Consequently, its histogram was first stretched using an equalisation algorithm (Section 2.7.3.3). A Gaussian 7×7 filter (Template 2.39) was used to smooth the enhanced details so that the Sobel operator would be able to detect major edges. Still, too many edges were found,

thus forcing another edge-finding operator (Roberts) to filter the negated Sobel edge-detected image. Finally, this image was negated and averaged. This attempt to come up with the best FeIm seems to have paid off as seen in Figures 4.23 to 4.26.

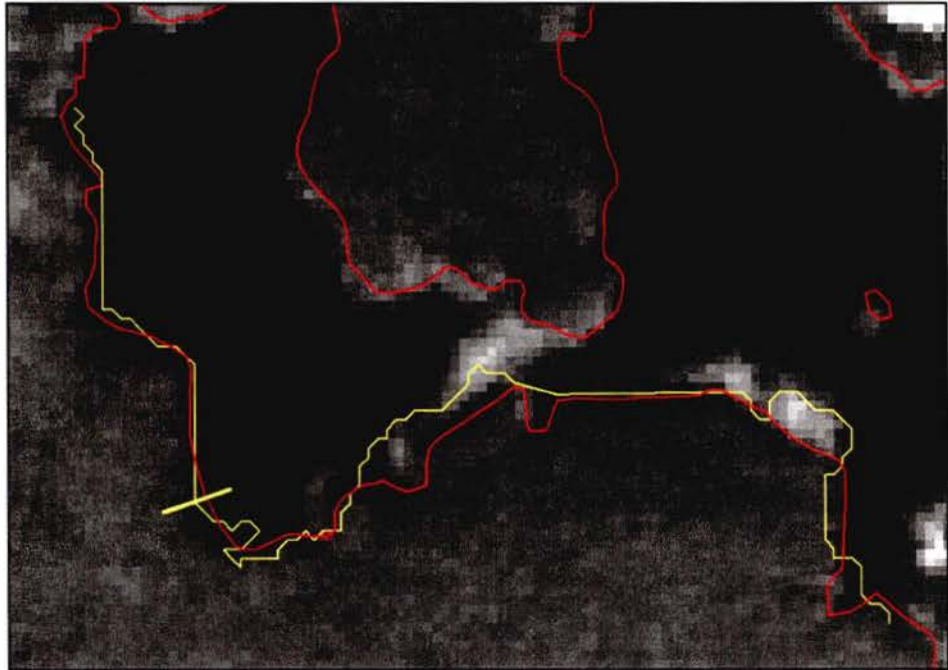


Figure 4.23 – SPOT 3 (10 m): Vector comparison. TRIM vectors are in red and the shoreline extracted by the tracing algorithm is in yellow. Shoreline profile (in yellow) is perpendicular to the shoreline.

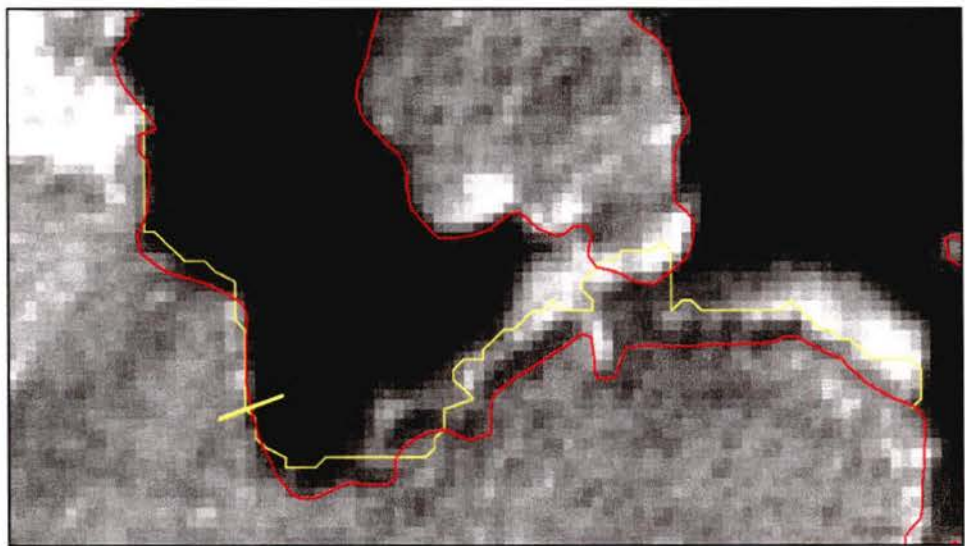


Figure 4.24 – Landsat 7 (15 m): Vector comparison. TRIM vectors are in red and shoreline extracted by the tracing algorithm is in yellow.

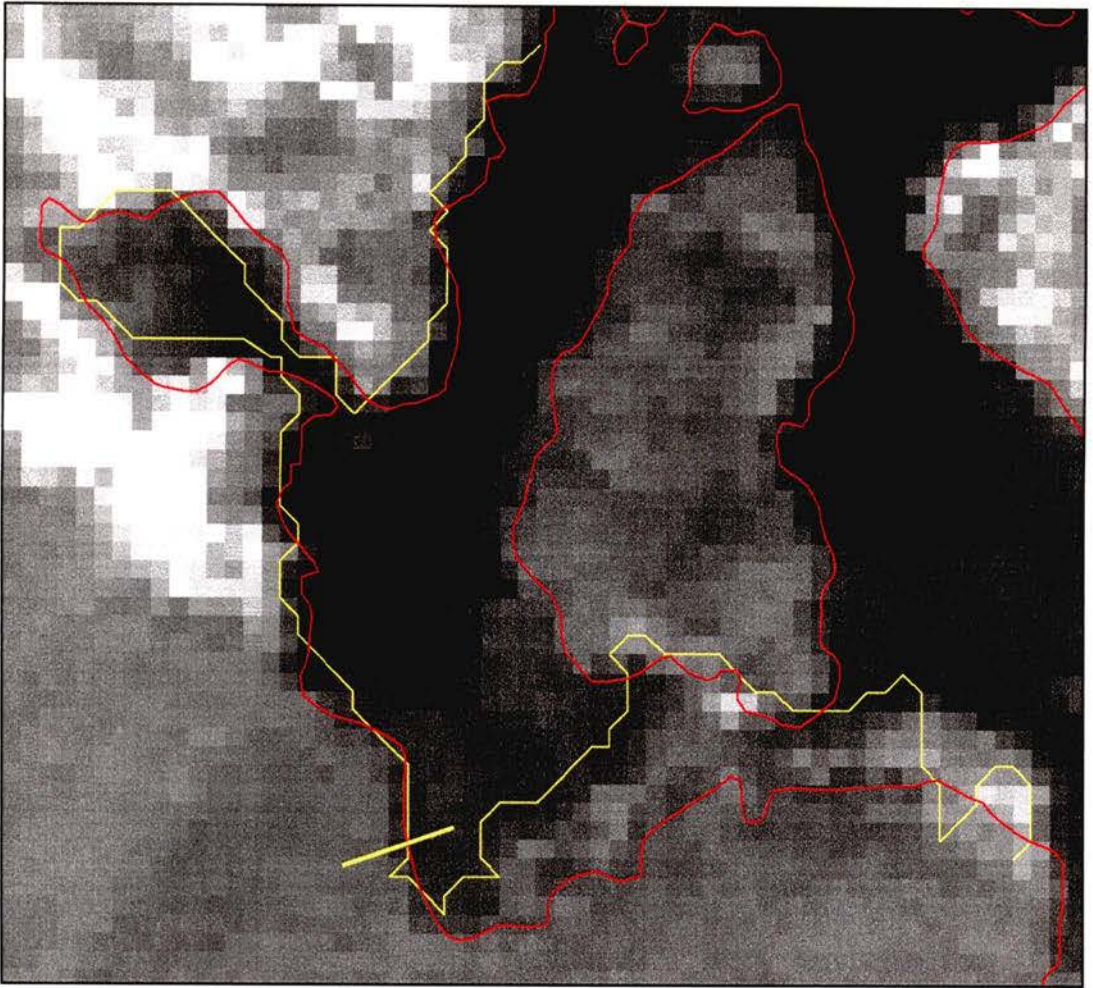


Figure 4.25 – SPOT 1 (20 m): Vector comparison. TRIM vectors are in red and the shoreline extracted by the tracing algorithm is in yellow. Shoreline profile (in yellow) is perpendicular to the shoreline.

The extracted shoreline vectors on the SPOT 3 image are the only ones that seem to have somewhat followed the difficult edge created by the mudflats, although they are still far from perfect when compared to the TRIM shoreline. The Landsat 7 (15 m) image yielded some vectors that overlapped the TRIM vectors but mainly on areas where the contrast between land and water was high. This can also be verified on the edge profiles (Figures 4.27a to 4.27d). Profiles for both Landsat 7 images show much larger amplitudes than those for the SPOT images where mudflats are more discernible, thus lowering the contrast between features.

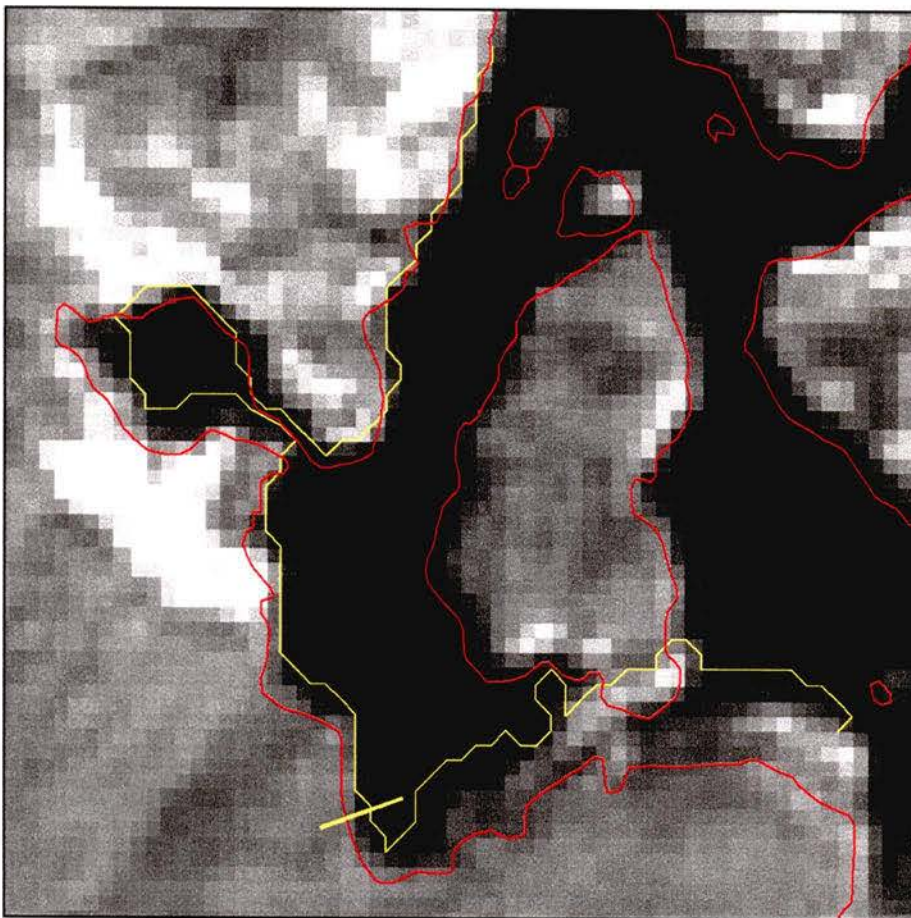
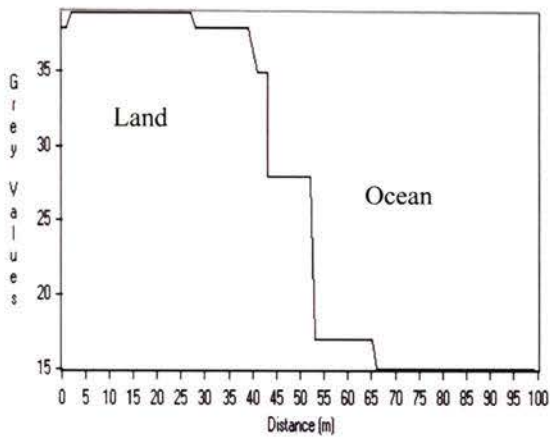
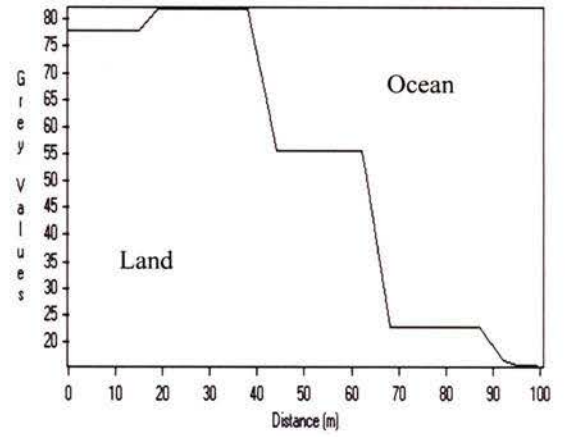


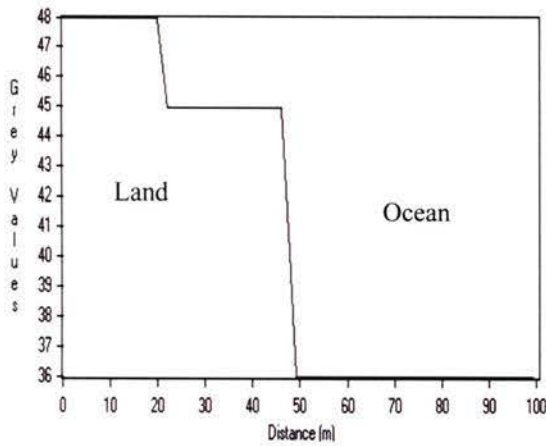
Figure 4.26 – Landsat 7 (30 m): Vector comparison. TRIM vectors are in red and the shoreline extracted by the tracing algorithm is in yellow. Shoreline profile (in yellow) is perpendicular to the shoreline.



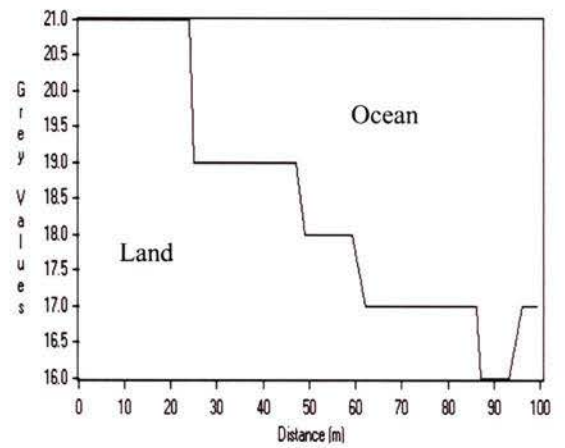
(a)



(b)



(c)



(d)

Figure 4.27 – Shoreline GLV profiles for mudflats/shallow areas. (a) Landsat 7 (15 m); (b) Landsat 7 (30 m); (c) SPOT 1 and (d) SPOT 3.

4.2.1.4 Rocky Coasts

Similarly to the images containing coasts with vegetation lines (Section 4.2.1.1), the rocky coastline images required only an edge operator and a smoothing filter to yield the best shoreline (Figures 4.28 to 4.31). The exception was the SPOT 3 image, which required two smoothing operators (Figure 4.28).

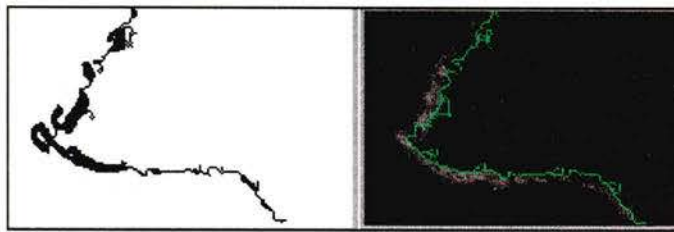


Figure 4.28 – SPOT 3 (10 m). Left: shoreline (in black) detected in 0.63 s using Gaussian 7x7 – Roberts – Negate – Average – 9 points. Right: original image and extracted shoreline in green.

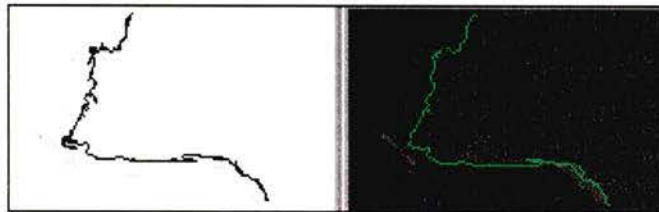


Figure 4.29 – Landsat 7 (15 m). Left: shoreline (in black) detected in 0.33 s using Sobel – Negate – 8 points. Right: original image and extracted shoreline in green.

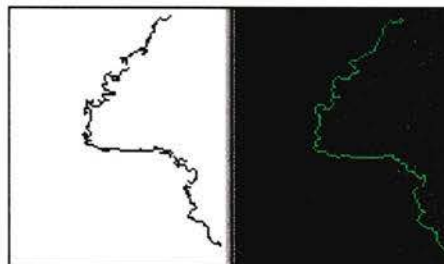


Figure 4.30 – SPOT 1 (20 m). Left: shoreline (in black) detected in 0.20 s using Roberts – Negate – 7 points – 0.20 s. Right: original image and extracted shoreline in green.

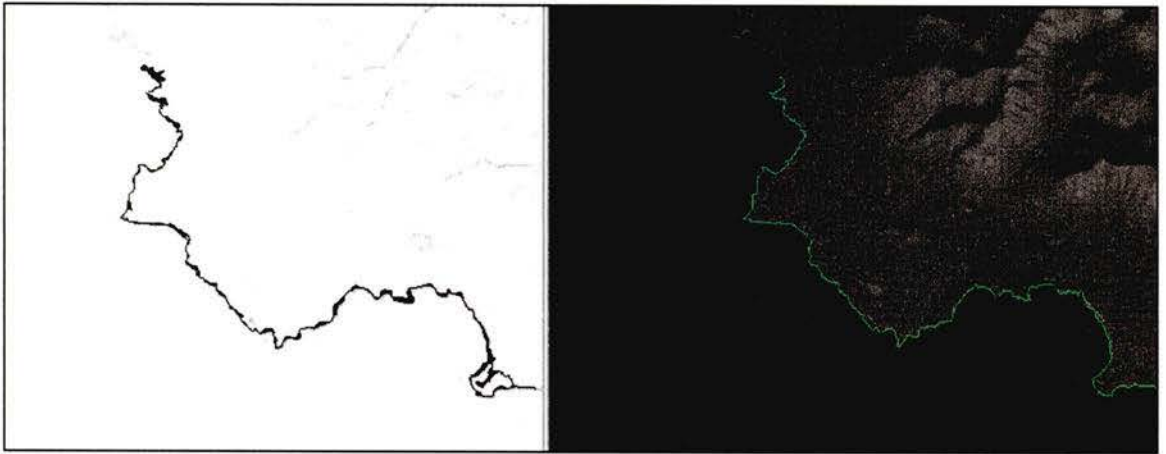


Fig 4.31 – Landsat 7 (30 m). Left: shoreline (in black) detected in 1.81 s using Prewitt – Negate – Average – 10 points. Right: original image and extracted shoreline in green.

A closer inspection shows that the location of extracted shoreline vectors on both SPOT 3 and Landsat 7 (15 m) images is generally very close to the TRIM shoreline vectors (Figures 4.32 and 4.33). The vectors extracted from the SPOT 1 and Landsat 7 (30 m) images are not nearly as similar to TRIM vectors (Figures 4.34 and 4.35). Still, all four images show areas where the difference between vectors is quite large (100 m on the SPOT 3, 80 m on the SPOT 1, 60 m on the Landsat 7 (15 m) and 90 m on the Landsat 7 (30 m)).

Edge profiles for these areas provide some interesting information about the sensor's imaging nature regarding time of acquisition (Figures 4.36a to 4.36d). The Landsat 7 (15 m) profile shows two peaks instead of only one normally expected for this area. This anomalous peak was created because the profile end touched some land features located on the water, which were probably emerged due to tidal movement. Apart from that anomaly, both Landsat 7 profiles show a clear distinction between land

and water, with the Landsat 7 (30 m) GLV amplitudes being much higher than those of Landsat 7 (15 m). This also seems to be the trend throughout the profiles analysed so far. Profiles for both SPOT images show two different responses for the same features. On the SPOT 1 image, forest GLVs are higher than forest GLVs found on the SPOT 3 image. By looking at the images it is possible to verify that the forest pixels on the SPOT 1 image have higher GLVs than pixels further down the profile end, where rocks and water features are located. On the other hand, forest pixels on the SPOT 3 image have much lower GLVs than pixels showing the rocky coast. This reversed behaviour is typical of different acquisition times and the related tidal movements (Jensen, 1996).

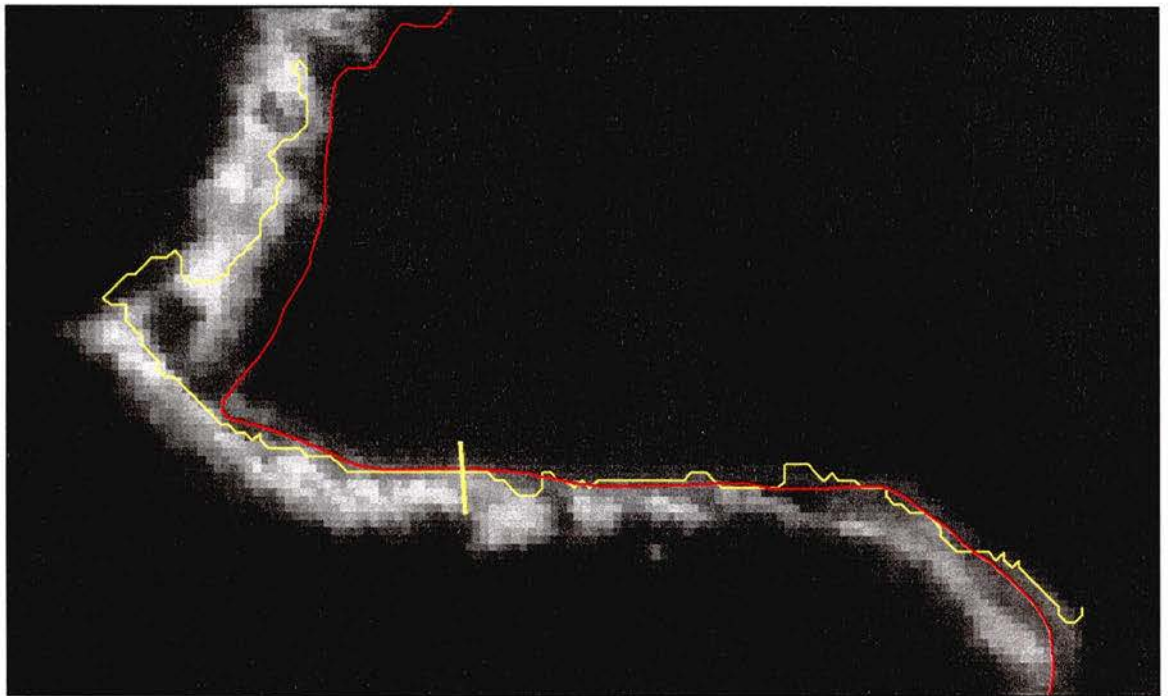


Figure 4.32 – SPOT 3 (10 m): Vector comparison. TRIM vectors are in red and the shoreline extracted by the tracing algorithm is in yellow. Shoreline profile (in yellow) is perpendicular to the shoreline.

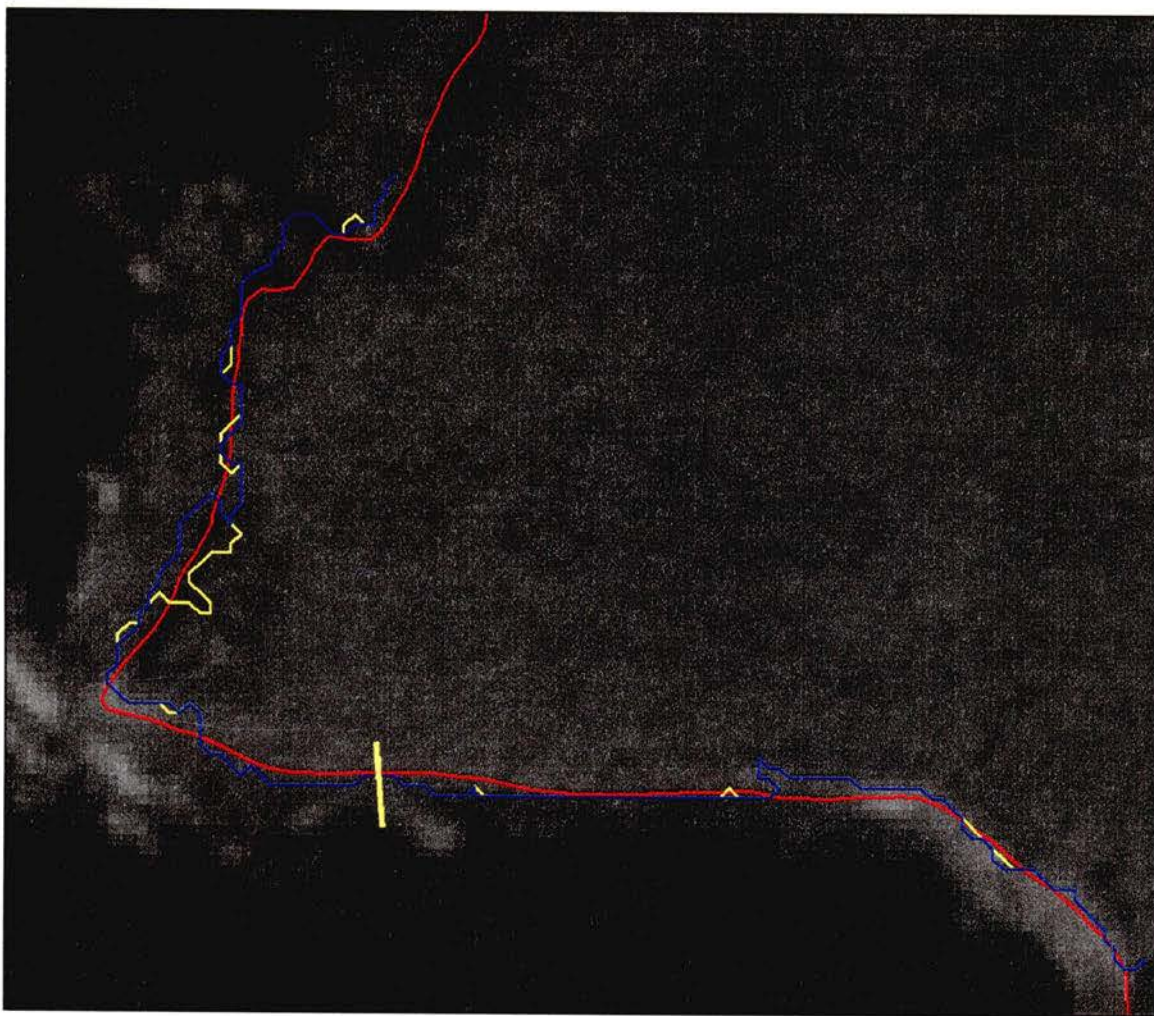


Figure 4.33 – Landsat 7 (15 m). Vector comparison. TRIM vectors are in red and shorelines extracted by the tracing algorithm are in yellow (Sobel) and blue (Frei-Chen). Shoreline profile (in yellow) is perpendicular to the shoreline.



Figure 4.34 – SPOT 1 (20 m). Vector comparison. TRIM vectors are in red and the shoreline extracted by the tracing algorithm is in yellow. Shoreline profile (in yellow) is perpendicular to the shoreline.

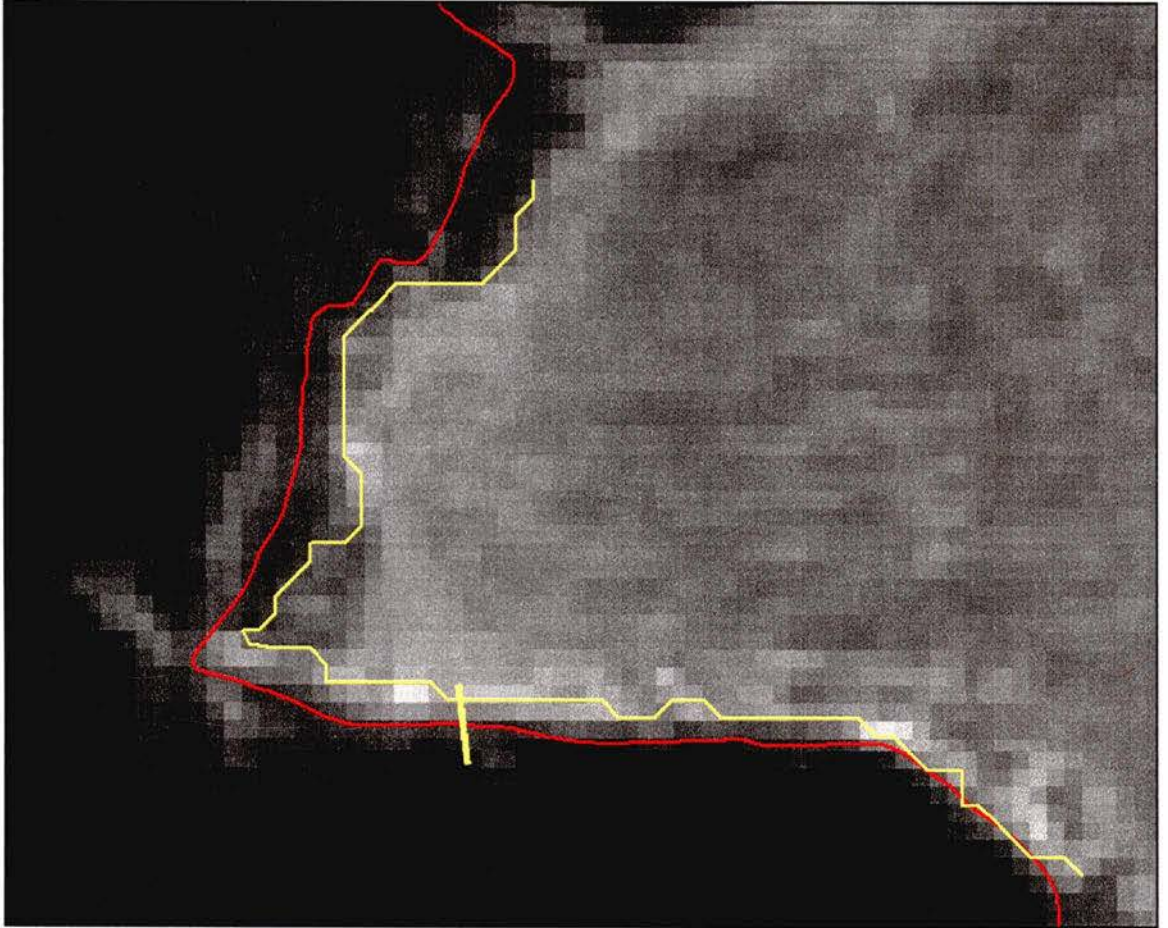
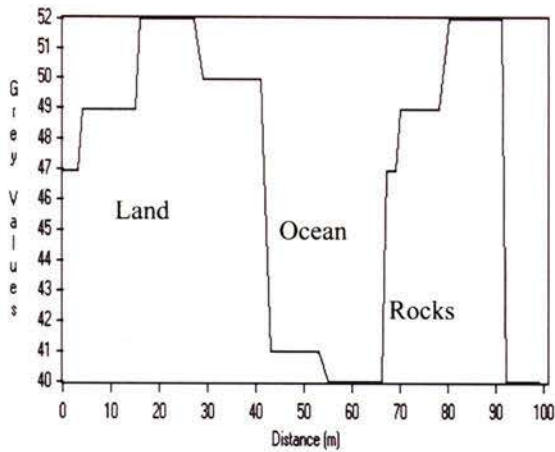
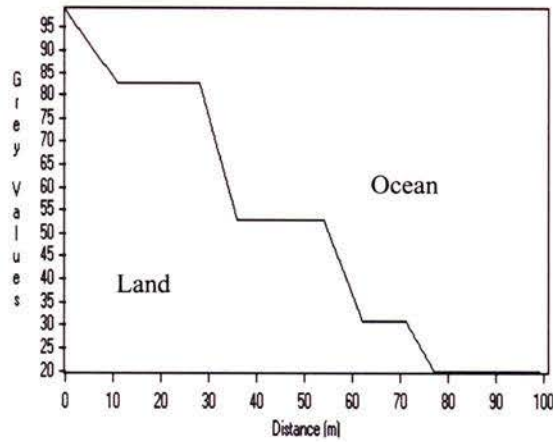


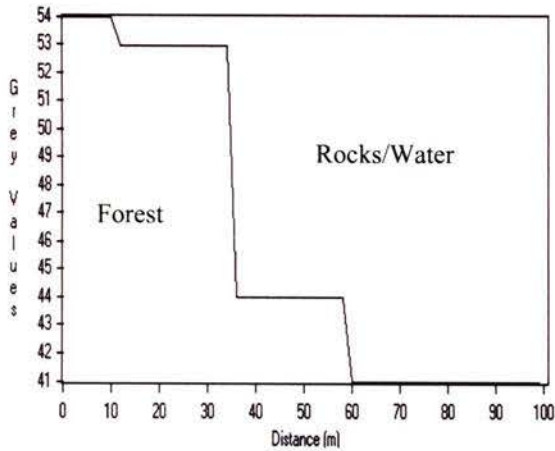
Figure 4.35 – Landsat 7 (30 m). Vector comparison. TRIM vectors are in red and the shoreline extracted by the tracing algorithm is in yellow. Shoreline profile (in yellow) is perpendicular to the shoreline.



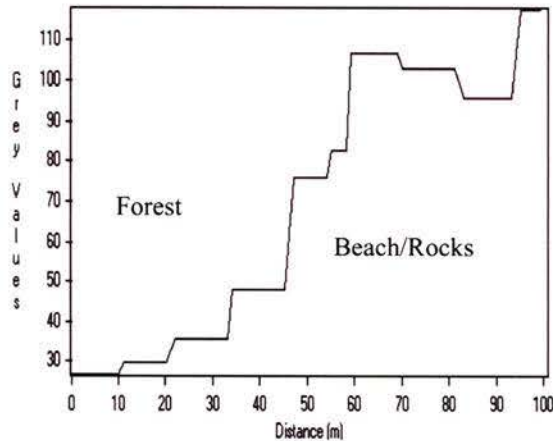
(a)



(b)



(c)



(d)

Figure 4.36 – Shoreline GLV profiles for rocky coasts. (a) Landsat 7 (15 m); (b) Landsat 7 (30 m); (c) SPOT 1 and (d) SPOT 3.

4.2.2 High-Resolution Images

Aerial orthophotos and the KFA-1000 photograph often required a more complex scheme for creating their respective FeIm surfaces. It was thought that the level of detail provided by these high-spatial resolution images would demand not only a different approach but also an increased number of steps to process the original images and select the correct placement of seed-points. Nevertheless, this was not always the case and FeIm could often be created using smoothing and edge operators similar to those used on the low-resolution images (Table 8). The difference was that for the high-resolution images the smoothing operators were more complex, thus requiring longer computation time, and two different edge detectors were used.

Table 8

FeIm Rationale for High-resolution Images

	Coasts with Vegetation Line	Sandy Beaches	Mudflats/Shallow Areas	Rocky Coasts
Aerial Orthophotos	Gaussian 7x7- Prewitt-Negate- Roberts-Negate- Average	Gaussian 7x7- Sobel-Negate- Roberts-Negate	Gaussian 7x7- Prewitt-Negate- Roberts-Negate- Average	Diamond 7x7- Gaussian 7x7- Prewitt-Negate- Roberts-Negate- Average
KFA-1000	Prewitt-Negate- Average	Gaussian 3x3- Roberts-Negate	Gaussian 7x7- Prewitt-Negate	Gaussian 7x7- Prewitt-Negate- Average

Contrast between land and water was lower in the aerial orthophotos than in the KFA images. In addition to that, the shadows of trees were a predominant problem in the aerial images. Shadows indicate areas without any “valid” shoreline data; in other words, there is no way of determining where and of what type the “true” shoreline is when all that can be seen are shadows (Figure 4.37).

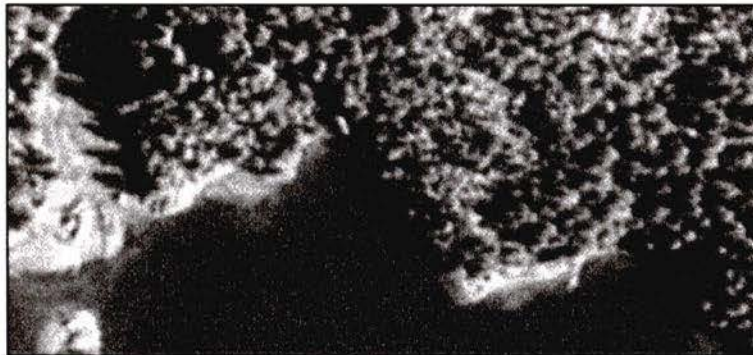


Figure 4.37 – Aerial Orthophoto. Example of shadows caused by trees.

As suggested by Zahn (1969), smoothing the images helped reduce the number of edge elements searched by the algorithm. However, because of the high spatial resolution of the aerial images, simple averaging was not enough to smooth out fine detail, including shadows. Median and Gaussian filters seemed to blur the images to the point where vegetation features were smooth enough to be considered an homogeneous feature and their boundary could be traced as an edge by the algorithm. However, these filters are much slower than the edge operators. This method works in a fashion similar to extracting edge information after reducing the original resolution, as suggested by Kelly (1971).

4.2.2.1 Coasts with Vegetation Line

The delineation of the shoreline on the KFA-1000 photograph was facilitated by the high contrast between land and water, thus requiring only one edge operator and one smoothing filter (Figure 4.38).

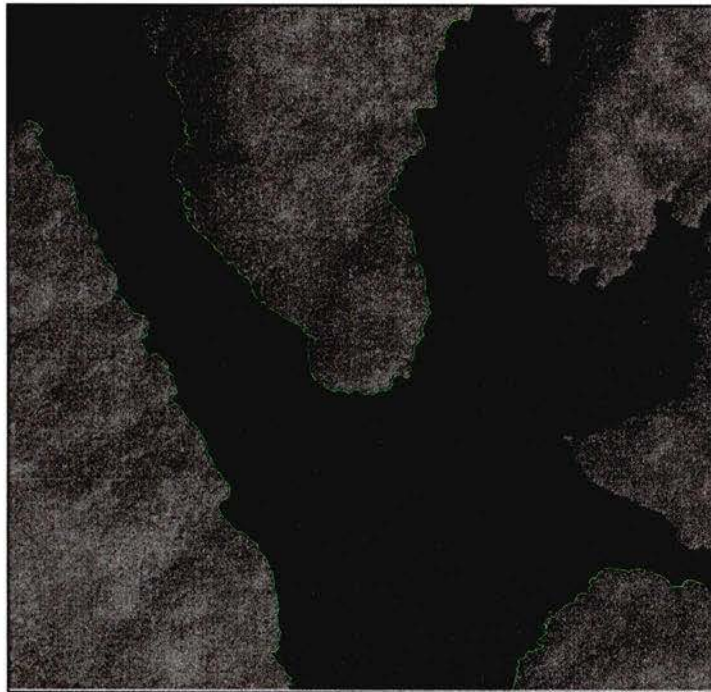


Figure 4.38 – KFA-1000. Shoreline (in green) extracted in 6.4 s using Prewitt – Negate – Average – 10 points.

A closer examination shows that the extracted vectors do a better job of delineating the edge of contact between land and water than the TRIM vectors, most of which are displaced inland (Figure 4.39).

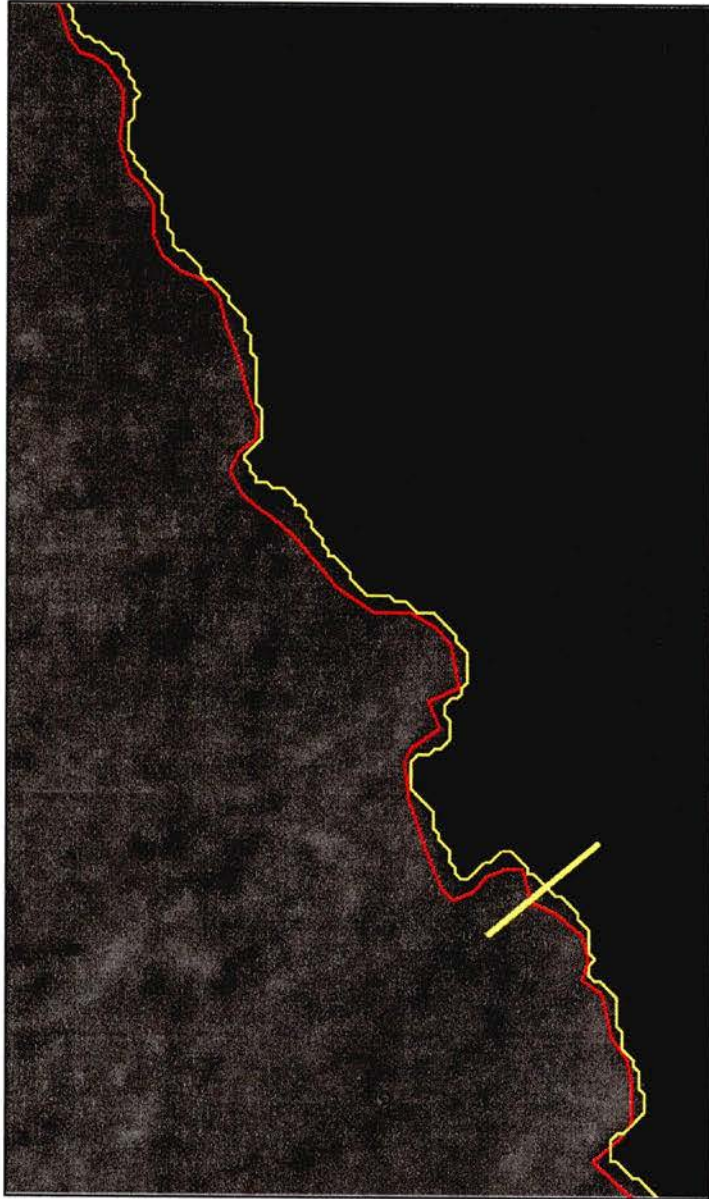


Figure 4.39 – KFA-1000. Vector Comparison. TRIM vectors are in red and the shoreline extracted by the tracing algorithm is in yellow. Shoreline profile (in yellow) is perpendicular to the shoreline.

The edge profile perfectly reflects the high contrast between land and water showing an abrupt drop in 70 GLVs over a distance of only 10 m (Figure 4.40).

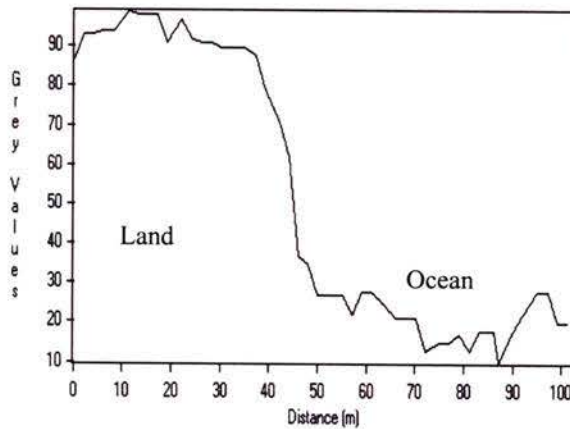


Figure 4.40 – KFA-1000 – Shoreline GLV profile for coasts with vegetation line.

The high resolution of the aerial photograph poses a greater challenge for the delineation of the shoreline in comparison to the other images used in this study. Figure 4.41 shows how much detail is available on the orthophotos, and also the extracted shoreline. Shadows are a constant problem at this resolution and also different acquisition angles make trees “bend” over the shoreline at some places. TRIM vectors show a very generalised shoreline whereas the extracted vectors attempt to represent every single feature along the shoreline (Figure 4.42).

Furthermore, land and water GLVs are not that different, as Figure 4.43 shows. The high number of peaks is typical of high-resolution images. Land features are constantly permeated by shadows, which cause a drop in the GLVs, as depicted by the high peaks on the left being interrupted by troughs on the profile.

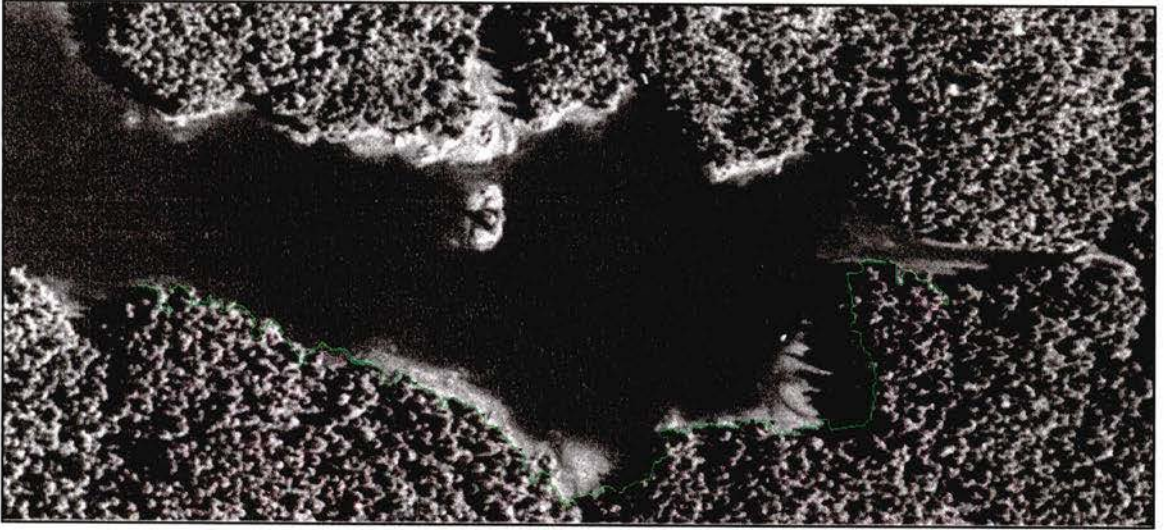


Figure 4.41 – Aerial Orthophoto. Shoreline (in green) extracted in 9.63 s using Gaussian 7x7 – Prewitt – Negate – Roberts – Negate – Average – 10 points.

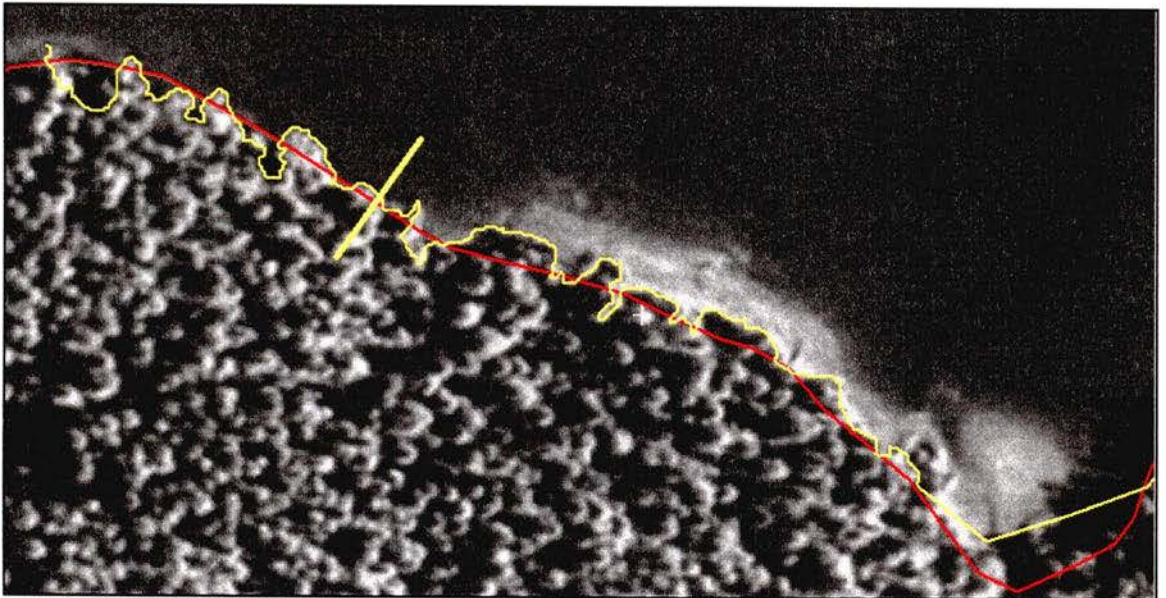


Figure 4.42 – Aerial Orthophoto. Vector Comparison. TRIM vectors are in red and the shoreline extracted by the tracing algorithm is in yellow. Shoreline profile (in yellow) is perpendicular to the shoreline.

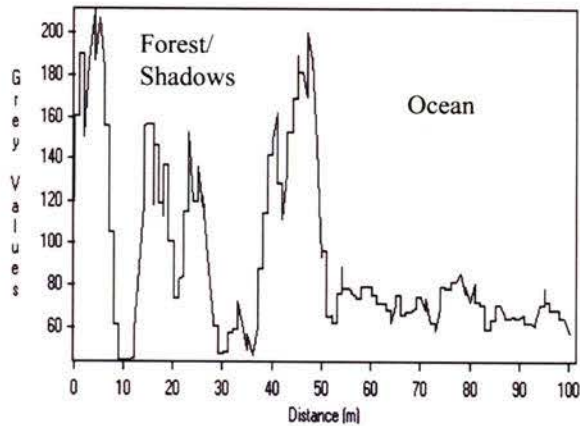


Figure 4.43 – Aerial Orthophoto. Shoreline GLV profile for coasts with vegetation line.

4.2.2.2 Sandy Beaches

Smoothing the KFA-1000 photograph before running the edge operator yielded better results than expected. A 3 x 3 Gaussian filter provided a more subtle smoothing than the simple average filter and the final shoreline was detected quite quickly due to the efficient Roberts edge detector (Figure 4.44).

Extracted shoreline vectors delineate the vegetation line more accurately than the TRIM vectors, however it is possible to see that the evaluation was misled in some areas. Still, there is some overlap between the two along certain parts of the shoreline (Figure 4.45).



Figure 4.44 – KFA-1000. Shoreline (in green) detected in 2.7 s using Gaussian 3x3 – Roberts – Negate – 10 points.

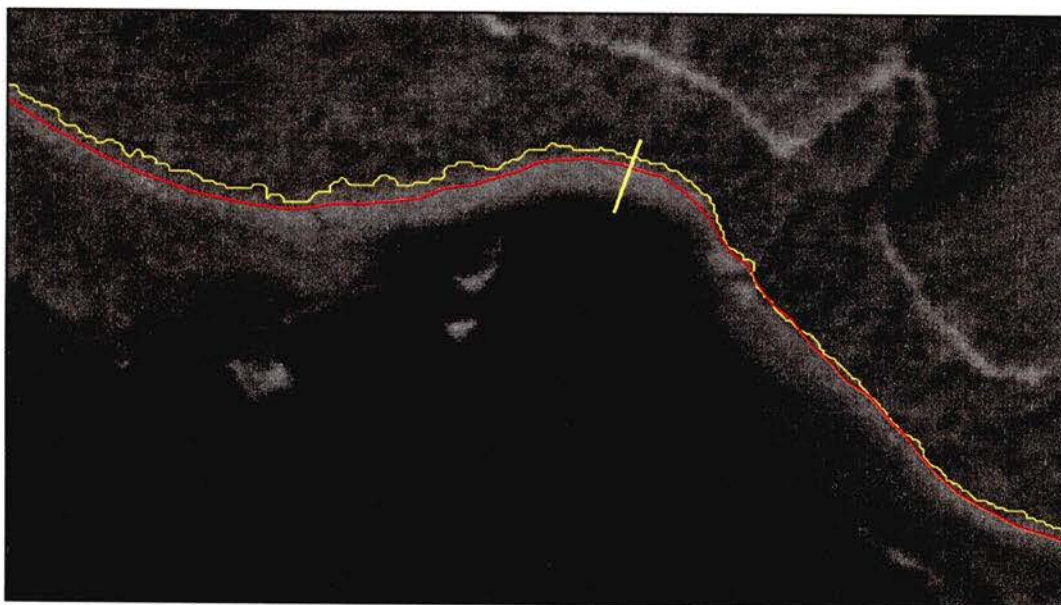


Figure 4.45 – KFA-1000. Vector comparison. TRIM vectors are in red and the shoreline extracted by the tracing algorithm is in yellow. Shoreline profile (in yellow) is perpendicular to the shoreline.

Beach pixels create the high and extensive peak on the edge profile whilst forest influence is depicted by somewhat lower GLVs. The GLVs of water pixels are much lower than those of any other features on the image (Figure 4.46).

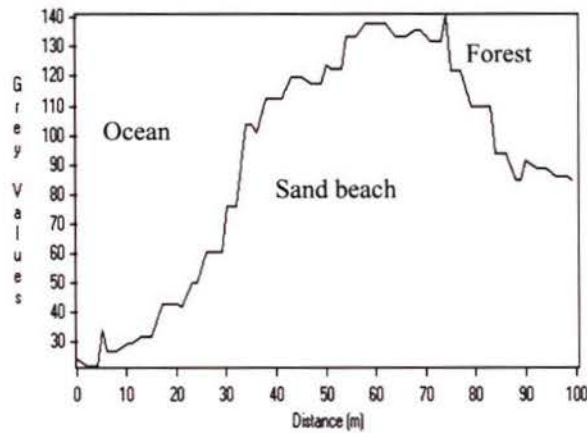


Figure 4.46 – KFA-1000. Shoreline GLV profile for sandy beaches.

The operators used on the aerial photograph allowed the tracing algorithm to avoid tree shadows in some areas but it also missed some trees in other areas (Figure 4.47). Overall it had a relative good performance given that it overlaps with the TRIM vectors in some areas and even detects the vegetation line more accurately than TRIM in others. The shoreline profile of the aerial orthophoto is not very different from the KFA-1000 profile, with beach pixels creating the massive middle peak and water and forest pixels having equivalent GLVs (Figure 4.48).

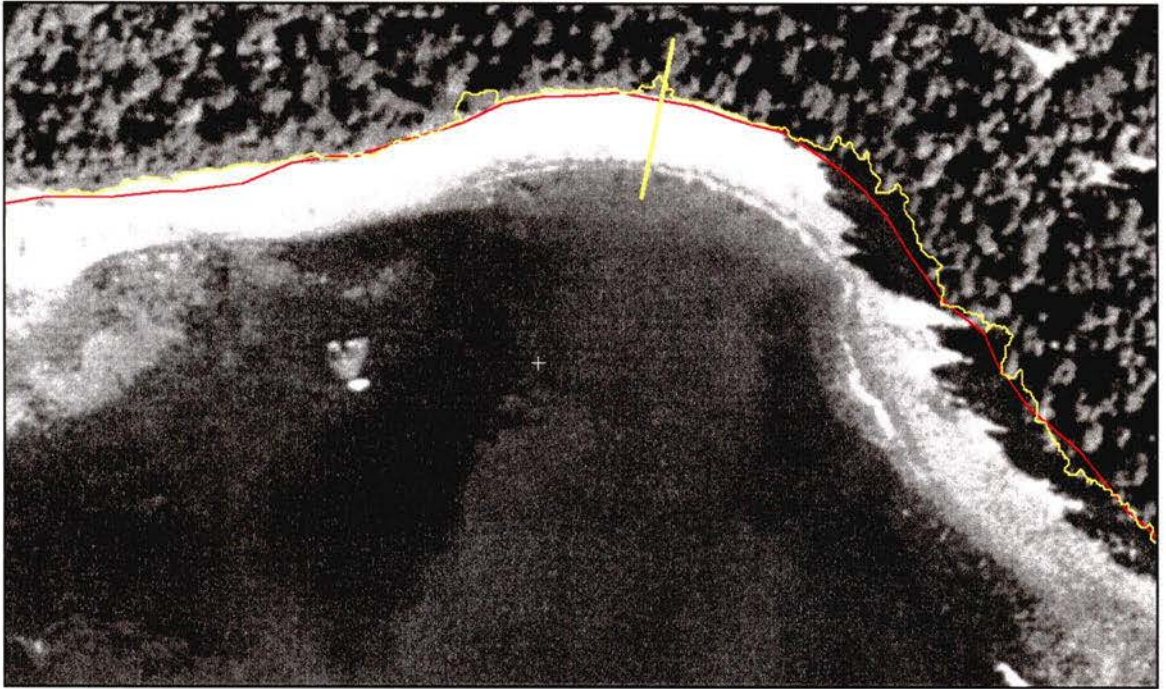


Figure 4.47 – Aerial Orthophoto. Shoreline (in yellow) extracted in 6.9 s using Gaussian 7x7 – Sobel – Negate – Roberts – Negate – 14 points. TRIM vectors are in red. Shoreline profile (in yellow) is perpendicular to the shoreline.

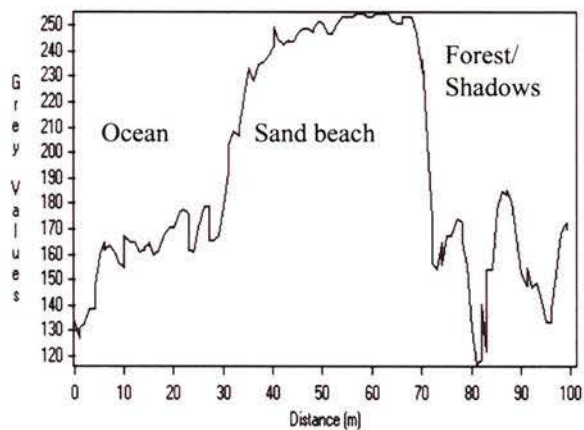


Figure 4.48 – Aerial Orthophoto. Shoreline GLV profile for sandy beaches.

4.2.2.3 Mudflats/Shallow Areas

The time required to extract the shoreline from the KFA-1000 photograph on Figure 4.49 was due to the large number of points used to guide the tracing algorithm over a shoreline approximately six kilometres long.

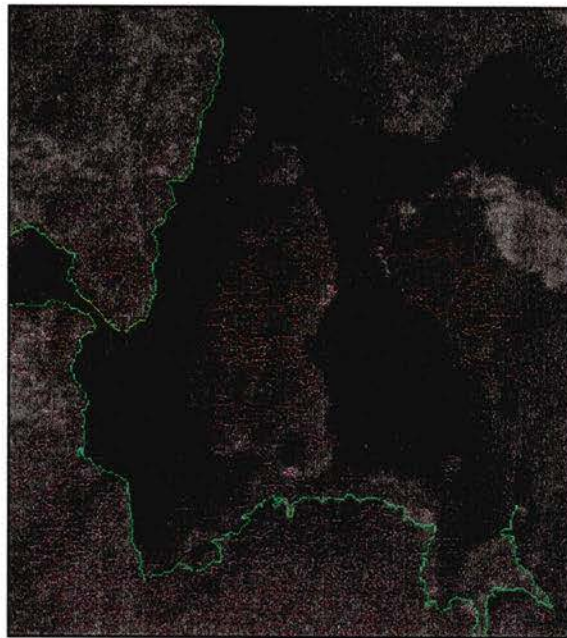


Figure 4.49 – KFA-1000. Shoreline (in green) extracted in 2.10 s using Gaussian 7x7 – Prewitt – Negate – 26 points.

By zooming in on the image it is possible to see that by tracing the vegetation line the algorithm defined the shoreline more effectively than the TRIM vectors (Figure 4.50). The shoreline profile on Figure 4.51a clearly shows the edge of contact between land and water whereas Figure 4.51b shows the profile extending from the forest over to the beginnings of the mudflat.

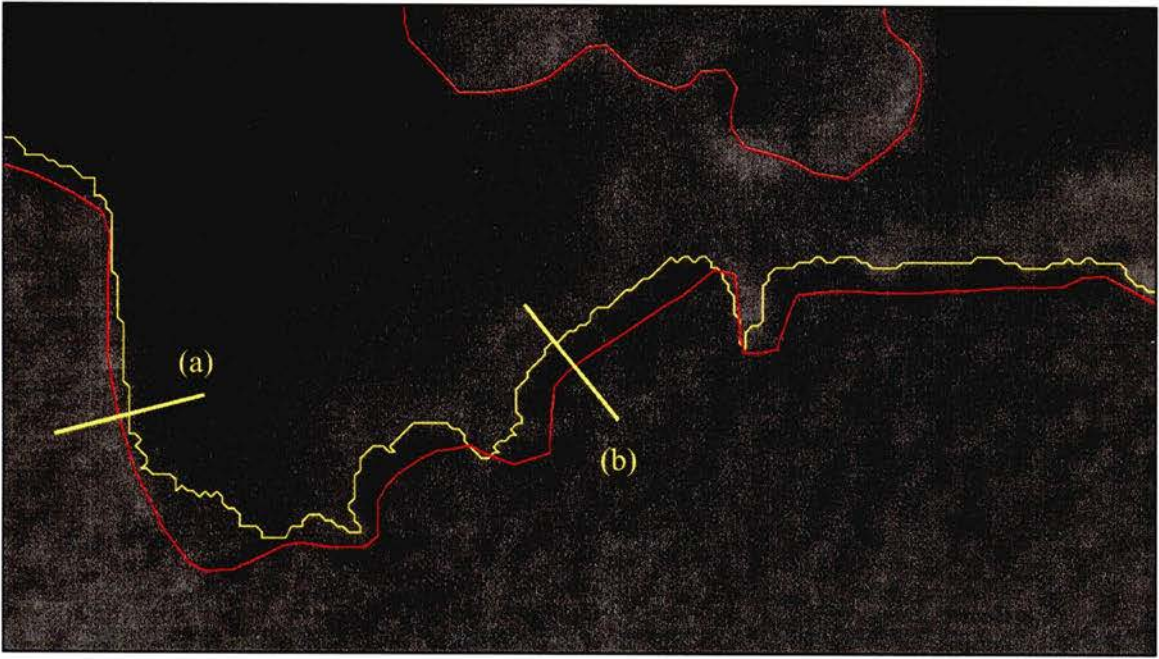
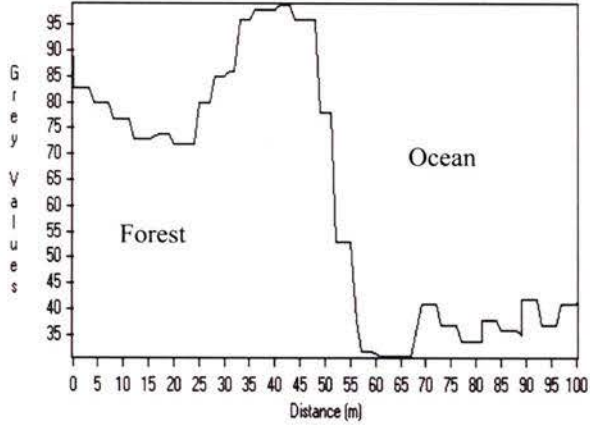
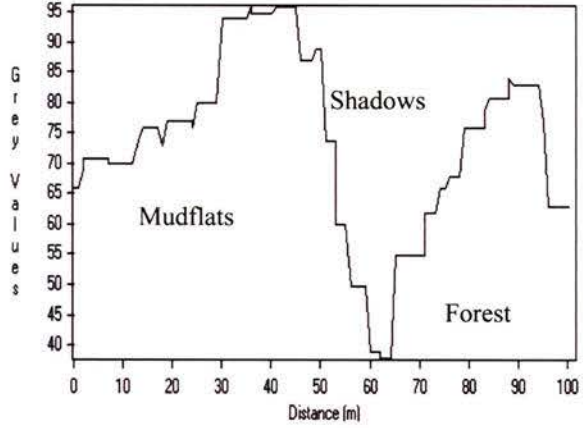


Figure 4.50 – KFA-1000. Vector comparison and two perpendicular profiles (a) and (b) of the shoreline. TRIM vectors are in red and the shoreline extracted by the tracing algorithm is in yellow.



(a)



(b)

Figure 4.51 – KFA-1000. Shoreline GLV profiles for mudflats/shallow areas.

The shoreline extracted from the aerial photograph depicts the shoreline more accurately than the TRIM vectors and successfully avoided tree shadows on many areas (Figure 4.52) even though, once again, excessive detail provided by the extracted shorelines is contrasted by the overly-generalised TRIM shoreline. The shoreline profile does not provide an instantly clear insight on the characteristics of land, water and mudflat features (Figure 4.53). The algorithm exhausted the allotted searching time of 10 seconds and required 14 seed-points, which is quite a large number of points considering the short (two-kilometre long) shoreline.

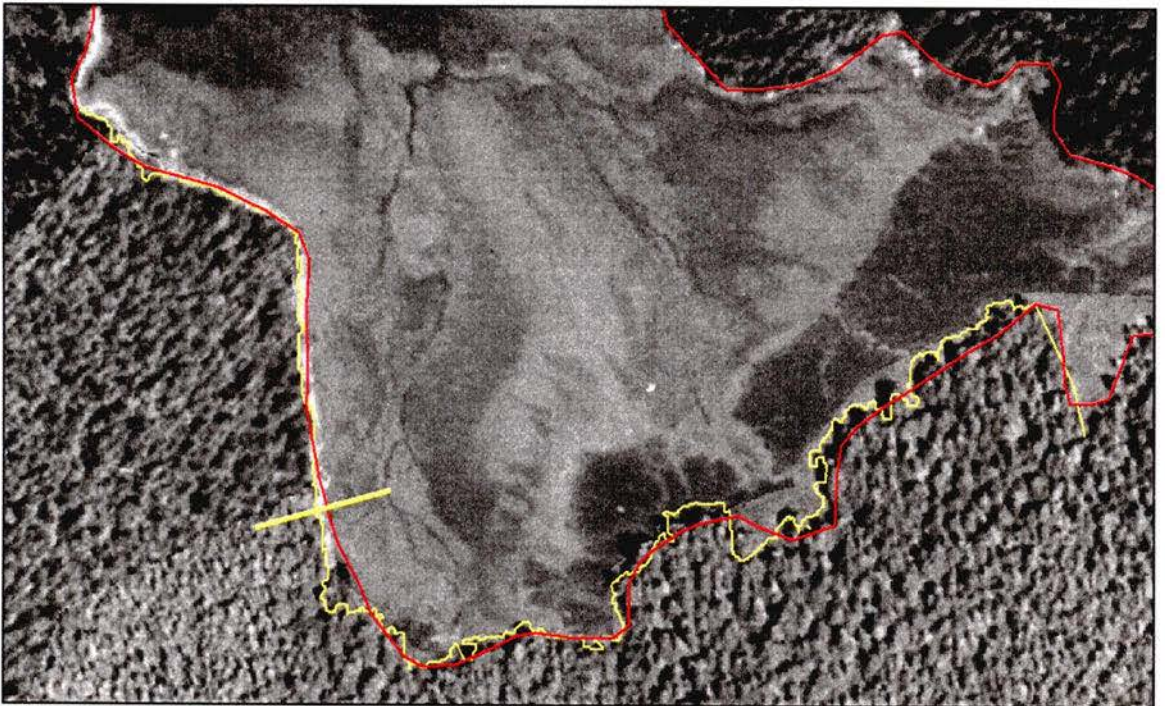


Figure 4.52 – Aerial Orthophoto. Shoreline (in yellow) extracted in 10 s using Gaussian 7x7 – Prewitt – Negate – Roberts – Negate – Average – 14 points. TRIM vectors are in red. Shoreline profile (in yellow) is perpendicular to the shoreline.

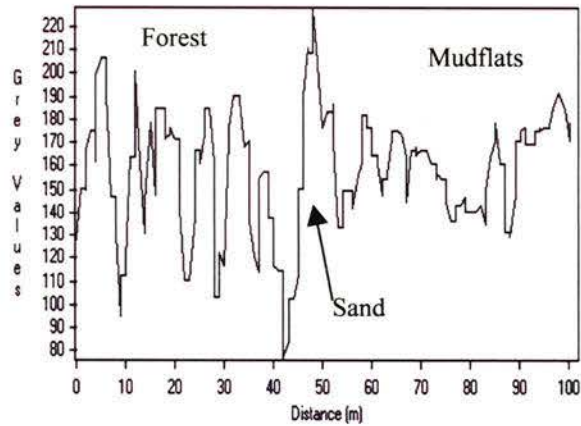


Figure 4.53 – Aerial Orthophoto. Shoreline GLV profile for mudflats/shallow areas.

4.2.2.4 Rocky Coasts

High contrast between forest and rocky ground along rocky coasts allowed for a very fast shoreline extraction by the tracing algorithm on the KFA-1000 photograph (Figure 4.54). It can be seen that the extracted shorelines outperform the TRIM vectors at the top of the image where they seem to follow the forest line more accurately. The shoreline profile clearly shows the forest and rock GLVs separated by sharp edges. A trough separating rock GLVs might be an indication of different slopes influencing brightness (Figure 4.55).

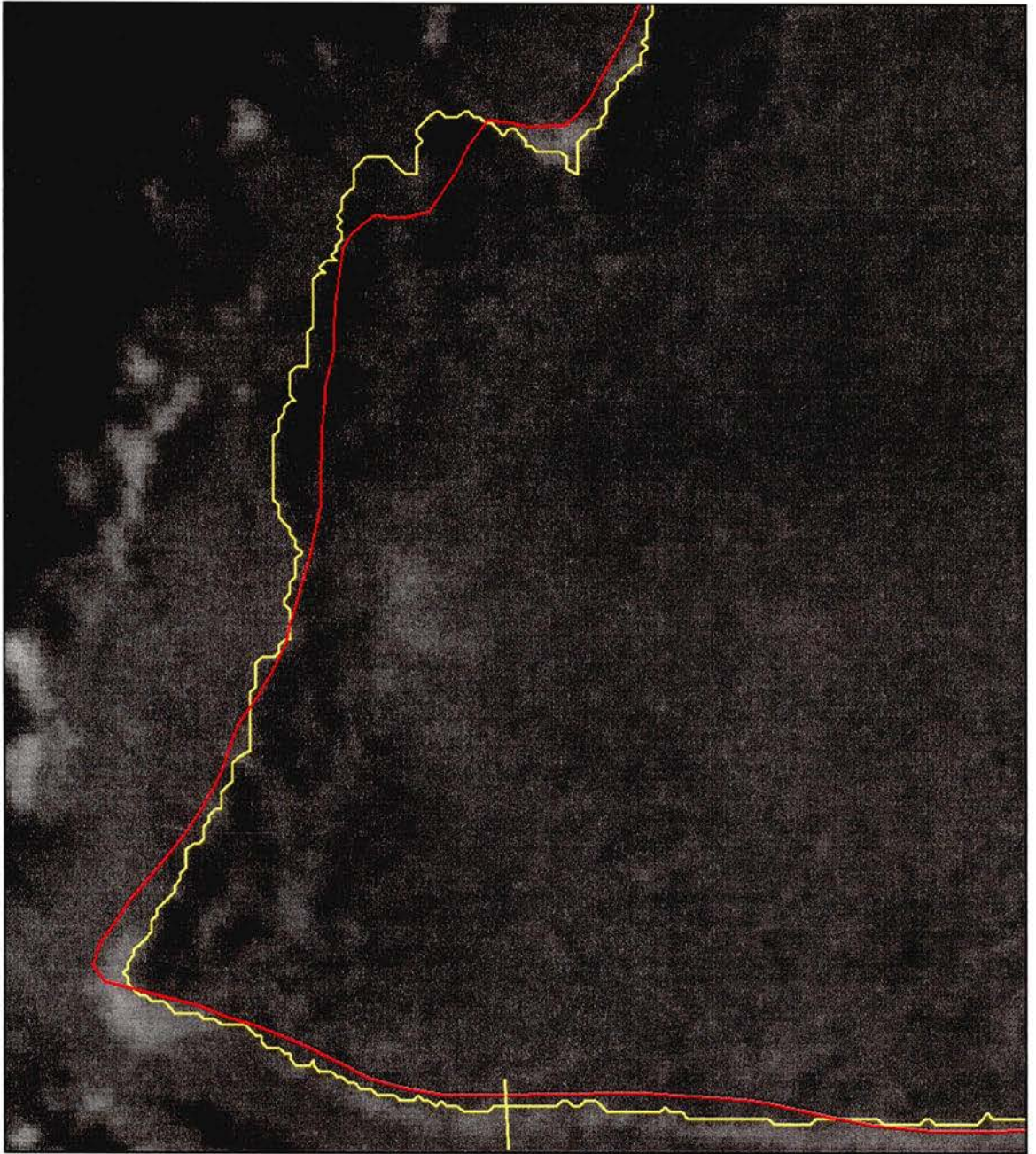


Figure 4.54 – KFA-1000. Shoreline (in yellow) extracted in 0.71 s using Gaussian 7x7 – Prewitt – Negate – Average – 11 points. Shoreline profile (in yellow) is perpendicular to the shoreline.

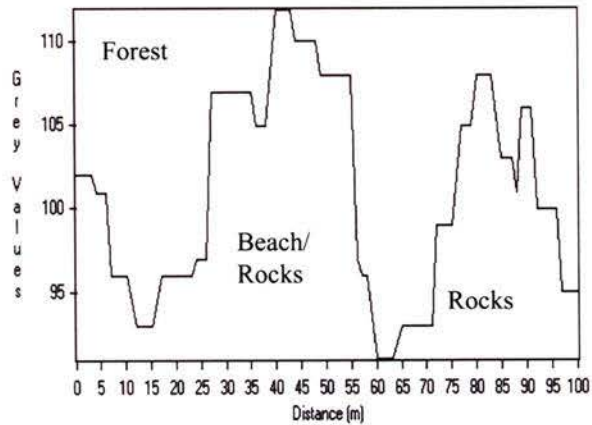


Figure 4.55 – KFA-1000. Shoreline GLV profile for rocky coasts.

Extracting the shoreline from the aerial orthophoto was a challenge for the tracing algorithm. A long and accurate shoreline was never achieved mainly because the quality of FeIm was never high enough to allow for a reasonable search (Figure 4.56). FeIm was difficult to create due to the low contrast between ground features. The same edge and smoothing operators used on the aerial photograph of mudflats were tried on the rocky coast, preceded by another, previously unused, smoothing operator. A median filter of the diamond shape (Template 2.36) was used to smooth forest features. Individual trees were therefore amalgamated with neighbouring trees, allowing the tracing algorithm to analyse them as a whole, and reducing the occurrence of intervening shadow. Forest GLVs are usually interspersed by shadow GLVs creating narrow peaks whereas rock GLVs are more consistent and create a wider peak (Figure 4.57).

Figure 4.56 also shows how the positioning of seed-points influences the tracing algorithm thereby creating different outputs (in blue and yellow).

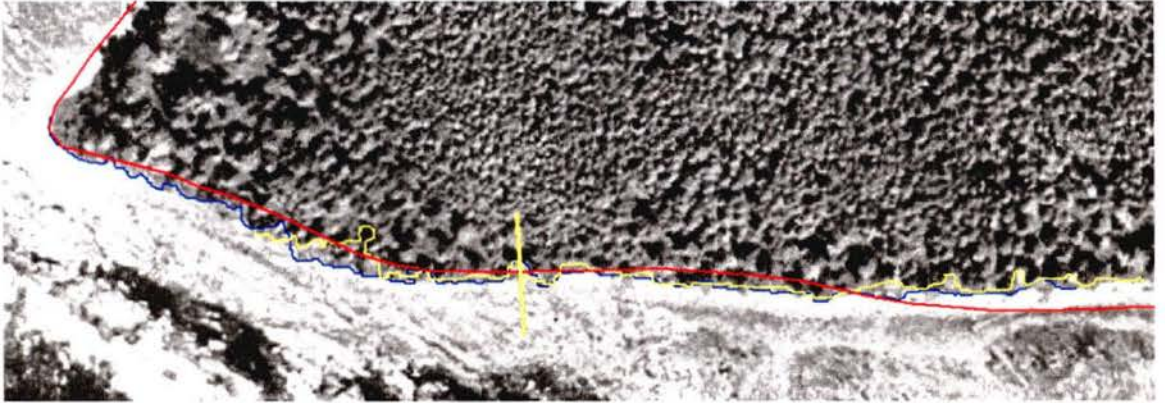


Figure 4.56 – Aerial Orthophoto. Shoreline extracted in 3.24 s (yellow) and 4.2 s (blue) using Diamond 7x7 - Gaussian 7x7 – Prewitt – Negate – Roberts – Negate – Average – 11 points and 10 points, respectively. TRIM vectors are in red. Difference in extracted shorelines is due to different seed-points. Shoreline profile (in yellow) is perpendicular to the shoreline.

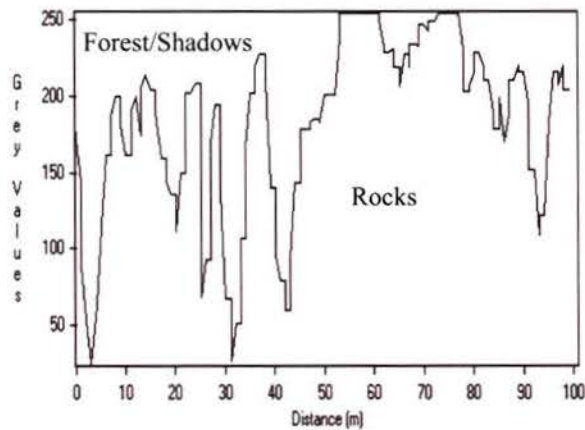


Figure 4.57 – Aerial Orthophoto. Shoreline GLV profile for rocky coasts.

5. Discussion

5.1 Digital Image Processing Techniques

The techniques presented in this thesis have been used in a wide range of scientific research since early 1960's. For simplification, they are often categorised into broad types of computer-assisted procedures:

- Image rectification/restoration;
- Image enhancement;
- Image classification;
- Data merging/GIS integration; and
- Modelling.

As described previously, all datasets were corrected for distortions introduced by the digital image acquisition process and were then referenced to the same projection and referencing system datum, Universal Transverse Mercator (UTM) and North American Datum (NAD83), respectively.

The tracing algorithm used in this project requires that images be processed to show only the most likely occurrence of shoreline features. This was achieved by applying different digital enhancements to the dataset histograms and also by filtering the

spatial domain of the datasets. Several attempts were made with different enhancement and filtering combinations in order to test the performance of the tracing algorithm. The images have very different characteristics and therefore their spatial resolutions and coast types were the deciding factors utilised when choosing which methods to use for creating the Feature Image FeIm.

The methods employed when generalising line features are intimately connected with the scale of the dataset (Heywood *et al.*, 1998). Unlike paper maps, digital images can be zoomed in and out, thus changing the scale. It is the scale of the dataset that determines the number of points necessary to accurately trace the shoreline. Choosing the right scale can become quite a dilemma when digitising. Simply zooming in on an image does not necessarily translate into features becoming easier to discern. The ability to differentiate between features depends on the spatial resolution of the sensor that captured the image in question. The one advantage to changing the scale is that zooming in or out will change the user's ability to make out individual pixels, which might, in some cases, help with positioning of the shoreline. The disadvantage of increasing the scale is that when a user zooms in a smaller area of the image will be evaluated each time, thus increasing the overall time spent digitising the shoreline.

For the reasons listed above, all images were subsequently analysed at full resolution. In other words, the display window was large enough to accommodate the full size of the image subsets, avoiding decimated (zoomed out) or resampled (zoomed in) images.

The tracing algorithm used in this study (Section 3.5.7 and Appendix A) does not take into account pixel size, rather it only evaluates the GLV costs of individual edge elements. This is an advantage for two reasons. Firstly, the algorithm has one less variable to evaluate, which has the potential to decrease processing times. Secondly, the algorithm did not need to be modified for each of the images with different pixel sizes (resolutions) that were used in this project. Still, the smaller the pixel size, the more edge elements to be evaluated and the longer it could take to perform the search. In general, time is more of a factor of the quality of FeIm than of the number of pixels itself.

Finally, as demonstrated in the previous chapter, images whose contrast between features was high allowed the algorithm to trace visually-accurate shorelines well under the time limit of 10 seconds. Aerial orthophotos with poor feature contrast exhausted the search time, but despite this, their shorelines were well delineated, although somewhat shorter in length.

5.2 Low-Resolution Images

When examining the low-resolution images, one can see why the tracing algorithm performs well on them; the shoreline is clearly apparent and the contrast between land and water is very high (Figures 4.4 to 4.9). Tracing the shoreline manually would also be an option, but depending on the size of the image and shoreline extent, manual tracing could be very cumbersome. Furthermore, although the digitisation resolution of an image influences what the algorithm considers as shoreline and where the

algorithm chooses to locate shoreline vectors, the results for the low-resolution images were more than detailed enough given their scales and spatial resolutions.

In general, low-resolution images require a very simple procedure for tracing the shoreline, as outlined in Table 7. Although images with a spatial resolution smaller than 10 m required a combination of different edge and smoothing operators, only one edge operator was needed to extract the shoreline vectors of images with a spatial resolution larger or equal to 15 m.

5.3 High-Resolution Images

Initially it was thought that the high level of detail provided by high-resolution images would be more of an obstacle to the tracing algorithm than it turned out to be. This can be explained, in part, by the spatial differences between aerial orthophotos and the KFA-1000 photograph. The spatial resolution of the KFA-1000 photograph is four times coarser, thus favouring the reflectance characteristics of groups of objects as a whole and thereby increasing the contrast between features. On the other hand, the high resolution of the aerial orthophotos picked up details that were not necessary for defining the shoreline. In fact, the level of detail often hindered the performance of the tracing algorithm. Shadows cast over portions of the shoreline by individual trees that were visible on the aerial photographs caused some problems. Since Felm could not distinguish between shadows and the shoreline beneath, the shadows were mistakenly traced by the algorithm.

Table 8 illustrates complex procedures involving at least two different edge and smoothing operators for extracting shorelines on the aerial orthophotos, whereas only one edge and one smoothing operators are required for the same job on the KFA-1000 photographs.

5.4 Overall Considerations

When compared with TRIM vectors, the results of the research described in this thesis show that it is possible to obtain accurate shoreline vectors by guiding the tracing algorithm through the use of seed-points. Better and more accurate shorelines could have been generated had the seeding of points been performed when zoomed in on the images. Unfortunately, due to software limitations, seed-points for this study had to be collected on the full resolution images without any image magnification. It can be assumed, due to this lack of magnification, that in some situations the seed-points were not placed exactly on the shoreline. The ability to zoom in and place seed points exactly on the shoreline would not only improve the quality of the shoreline traced by the algorithm, it could also potentially reduce the number of required seed-points, thereby reducing the amount of user-time spent seeding the coastline. Table 5 provides some information regarding the number of seed-points and search time. Nevertheless, by using more seed-points, search time is reduced because the algorithm makes use of the extra seed-points to minimise search time, thus maximising the quality of the extracted shoreline.

An example of how changing the position of seed-points influences the behaviour of the tracing algorithm can be seen on Figure 4.56. By using the same FeIm with

different seed-points, a different shoreline was traced. This variation is due to the localised search conducted by the tracing algorithm. Changes in seed-point position greater than one pixel in any direction would presumably yield different paths if the seed-points fell on pixels with different GLVs. This is due to the different costs associated with the different GLVs. Therefore, a thorough knowledge of the image characteristics is extremely important for placing the seed-points. Not only are the characteristics of the original image important, but most of all, FeIm must be able to provide enough edge information in order to aid the image analyst, along with comparisons with the original image, to place seed-points that will minimise search time and maximise the results.

In some cases the vectors extracted by the tracing algorithm clearly define a more obvious shoreline than those depicted by TRIM vectors. (Figures 4.8, 4.9, 4.17, 4.42, 4.45 and 4.50). Three factors are responsible for this. Firstly, the geometric correction used on the images might not have been as good as expected in some areas (Figure 4.54), this resulting in shifted TRIM vectors when displayed on the georeferenced images. The fact that the KFA-1000 photograph was first geocoded using TRIM vectors and was then used to geocode all of the other images could have added some extra distortion. Also, areas close to the edges of the images usually have a higher distortion, adding to the fact that sometimes it was difficult to collect points accurately on all of the images. Secondly, the TRIM vectors were originally collected through the manual digitalisation of 1:60,000 aerial orthophotos and later scaled to a scale of 1:20,000. When these vectors are overlaid on high-resolution images, such as the images used in this study, it is possible to see how generalised those vectors are. Collecting ground control points on an image by using shoreline vectors as the georeferenced dataset might not give the best results due to the

generalised vectors that could attenuate features seen on the imagery. That is the reason for the differences in vector resolution between the extracted shorelines and the TRIM shorelines. Lastly, TRIM shoreline vectors are not delineated very precisely (J. E. Harper, personal communication, July, 2002).

Differences in time of acquisition and sensor characteristics could also influence the comparison of shoreline vectors. Figures 4.32 and 4.34 show the same rocky coast and there are noticeable differences between the two images. These were most likely caused by different imaging angles or different oceanographic conditions such as waves or tides. One common cause of error that was not a factor in this study was the shift of edge pixels that usually occurs with the use of consecutive edge operators (Lyon, 1999). Analysis of the extracted shorelines compared with TRIM vectors on the high- and low-resolution images did not indicate major shoreline displacements (Figures 4.39 to 4.56).

The Roberts, Sobel and Prewitt edge operators (Section 2.7.4) were the most efficient among those used in this study. Roberts is the simplest and fastest edge operator, and it yielded good edge images, however it was not able to match the better tracing performance of the other two (Tables 7 and 8). By employing a larger kernel, Sobel and Prewitt operators make use of more GLVs in order to evaluate edges. The slight increase in computational time in comparison to the Roberts operator is not always evident and could be compensated for by increasing the number of seed-points (Table 5). Figures 4.17 and 4.33 illustrate the performance of two different edge detectors using the same seed-points for the shoreline tracing. Both edge operators have a similar performance, detecting the shoreline quite accurately, however it is evident that at some segments of the shoreline they yielded different results.

In summary, aerial orthophotos required one smoothing operator and two edge detectors, while the KFA-1000 photograph required only one smoothing operator and one edge detector. Low-resolution images, on the other hand, required only one set of smoothing operator and one edge detector, with the exception of the SPOT 3 images, which occasionally required a different rationale. High level of detail on high-resolution images decreased the contrast between ground features, making it difficult for the tracing algorithm to extract shorelines, whereas high contrast on low-resolution images allows for a more consistent and easier shoreline extraction.

6. Conclusion

6.1 Overall Assessment

This study has successfully developed procedures for improving the extraction of shorelines from remotely-sensed imagery. More specifically, the techniques presented here outline a semi-automatic approach to tracing shorelines by using simple and accessible algorithms. These algorithms have been tested in the literature and have proven their reliability by showing consistent results (Duda *et al.*, 1973; Jain, 1989; Schalkoff, 1989; Lyon, 1999).

The basic requirement for beginning the shoreline extraction process is the enhancement of edge information contained in the image datasets being used. Processing the images using edge operators extracts the edge information, indicating the virtual position of the shoreline. Accurate edge detection relies on the user's choice of the correct edge operator, which in turn depends on the characteristics of the image and the coast type.

Remotely-sensed images have very different characteristics depending on the sensors used to acquire them and also on the features contained in the image. The images used in this study were grouped into two main types: low spatial resolution and high spatial resolution. Low-resolution images were acquired by three different sensors: Landsat 7, SPOT 1 and SPOT 3. These sensors provided both panchromatic and multi-

spectral data. Panchromatic image data have a finer spatial resolution than multi-spectral data, thus enabling a more detailed visualisation of features.

Different coast types can be found around Clayoquot Sound in British Columbia, the area in which the datasets were collected. From these coast types, the most predominant were selected for use in this study. These include: coasts with vegetation line, sandy beaches, mudflats or shallow areas, and rocky coasts. The reason for using different coast types was the challenges their unique characteristics would pose for the tracing algorithm. Some coasts present a high contrast between land and water features, as seen on Figures 4.4 to 4.9. On the other hand, the contrast between features is much lower on other coast types, such as mudflats, and it is difficult to visually differentiate ground features (Figures 4.25 and 4.50).

It is the high contrast that allows the edge operators to detect the most promising edges. But in some images, especially the aerial orthophotos, the contrast is not strong enough and a more complex methodology is required for tracing the shoreline. Smoothing filters are therefore applied prior to the edge operators and a set of two edge operators are then used to extract edges. In some cases, another smoothing operator is passed on the images to further attenuate random noise, thus maximising the smoothness of the edges. In this respect, the KFA-1000 photograph behaves in a similar way to the low-resolution images by requiring only one edge operator (Tables 7 and 8).

The algorithm used for tracing the shoreline was chosen for its simplicity and ease of use, and its ability to be fully customised to the user's needs according to the datasets and methodologies available. Dijkstra's A* algorithm modified by Martelli (1972, 1976)

and implemented by Lyon (1999) makes good use of heuristics through seed-points used to guide the search for the best shoreline. At least two seed-points are required, namely start and end nodes, for the search to be carried out. If more than two seed-points are used, the extracted shoreline must pass through these intermediate nodes. The main advantage of this manual method over automatic extraction algorithms, such as classification methods, is its ability to evaluate only the most promising edges. This local evaluation reduces the amount of time required for processing the image by allowing the user to provide important information regarding the approximate location of the shoreline. Hence, the user needs to have a reasonable knowledge of the shoreline in both the original image and FeIm in order to locate seed-points accordingly.

6.2 Final Remarks

Remotely-sensed data provide broad and reliable information on coasts anywhere in the world at any time. This enables the creation of more accurate and more frequently-updated maps and charts of these areas with significantly lower costs than techniques employed 30 or 40 years ago. This in turn translates into opportunities for improved management and conservation of coastal environments.

Semi-automated shoreline delineation techniques using remotely-sensed imagery have been found to be efficient and fast. This is because they incorporate the input of an

analyst who guides the tracing algorithm by inputting seed-points along the most-likely location of the shoreline.

The accurate determination of shoreline position using remote sensing techniques is a difficult task that must be undertaken carefully. When evaluating the position of the shoreline the user must consider the physical, geological, biological and chemical aspects involved in its evolution.

7. References

- Bandat, H. F. (1962). *Aerogeology*. Gulf Publishing Co., Texas. 350 p.
- Bo, G.; Dellepiane, S.; De Laurentiis, R. (2001). Coastline Extraction in Remotely Sensed Images by Means of Texture Features Analysis. **In:** *IEEE (2001) International Geoscience and Remote Sensing Symposium (IGARSS 2001)*, University of New South Wales, Sydney, Australia, 9-13 July 2001.
- Cameron, H. L. (1965). Coastal Studies by Sequential Air Photography. **In:** El-Ashry, M. T. (ed.) (1977). *Air Photography and Coastal Problems*. Benchmark Papers in Geology, 38:168-177. Dowden, Hutchinson & Ross, Inc., Pennsylvania.
- Canny, J. (1986). A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8 (6): 679-698.
- Castleman, K. R. (1996). *Digital Image Processing*. Prentice Hall, Englewood Cliffs, New Jersey. 667 p.
- CCRS (2002). Canada Centre for Remote Sensing. Glossary Database web-site: <http://dweb.ccrs.nrcan.gc.ca/ccrs/db/glossary/gloste.cfm?Language=English&GlosID=412> .
- Chen, H. S. (1985). *Space Remote Sensing Systems – An Introduction*. Academic Press, Inc., Florida. 257 p.
- CHS. (1999). *Canadian Hydrographic Service* web-site: <http://www.ios.bc.ca/ios/chs/tides/datums/Datums.html> .
- Clark, J. R. (1994). Integrated Management of Coastal Zones. *FAO Fisheries Technical Paper No. 327*. Food and Agriculture Organisation of the United Nations, Rome. 175 p.
- Collins, S. M.; Skorton, D. J.; Geiser, E. A.; Nichols, J. A.; Conetta, D. A.; Pandian, N. G.; Kerber, R. E. (1984). Computer-Assisted Edge Detection in Two-Dimensional Echocardiography: Comparison with Anatomic Data. *The American Journal of Cardiology*, 53: 1380-1387.
- Couloigner, I. and Ranchin, T. (2000). Mapping of Urban Areas: A Multiresolution Modeling Approach for Semi-Automatic Extraction of Streets. *Photogrammetric Engineering & Remote Sensing*, 66 (7): 867-874.
- Curran, P. J. (1985). *Principles of Remote Sensing*. Longman Group Ltd., London. 274 p.

- Davies, E. R. (1990). *Machine Vision*. Academic Press, Cambridge. 547 p.
- Davis, R. A. (1996). *Coasts*. Prentice-Hall, Inc., New Jersey. 274 p.
- Deriche, R. (1987). Using Canny's Criteria to Derive a Recursively Implemented Edge Detector. *International Journal of Computer Vision*, 1 (2): 167-187.
- Dietz, R. S. (1947). Aerial Photographs in the Geological Study of Shore Features and Processes. **In:** El-Ashry, M. T. (ed.) (1977). *Air Photography and Coastal Problems*. Benchmark Papers in Geology, 38: 23-31. Dowden, Hutchinson & Ross, Inc., Pennsylvania.
- Dijkstra, E. W. (1959). A Note on Two Problems in Connexion with Graphs. *Numerische Mathematik*, 1: 269-271.
- Dillabaugh, C. R. (2000). *Multi-Resolution Extraction of River Networks from Digital Satellite Imagery*. Master of Science Thesis, University of Victoria, Victoria, BC. 201 p.
- Duda, R. O. and Hart, P. E. (1973). *Pattern Classification and Scene Analysis*. John Wiley & Sons, New York. 482 p.
- El-Ashry, M. T. (ed.) (1977). *Air Photography and Coastal Problems*. Benchmark Papers in Geology, v. 38. Dowden, Hutchinson & Ross, Inc., Pennsylvania. 425 p.
- El-Ashry, M. T. and Wanless, H. R. (1967). Shoreline Features and Their Changes. **In:** El-Ashry, M. T. (ed.) (1977). *Air Photography and Coastal Problems*. Benchmark Papers in Geology, 38: 178-183. Dowden, Hutchinson & Ross, Inc., Pennsylvania.
- ENVI (2000). *The Environment for Visualizing Images – User's Guide - Version 3.4*. Research Systems, Inc. Boulder, Colorado. 930 p.
- Fisher, R.; Perkins, S.; Walker, A.; Wolfart, E. (2002). Image Processing Learning Resources – HIPR2 web-site: <http://www.dai.ed.ac.uk/HIPR2/> .
- Fischler, M. A.; Tenenbaum, J. M.; Wolf, H. C. (1981). Detection of Roads and Linear Structures in Low-Resolution Aerial Imagery Using a Multisource Knowledge Integration Technique. *Computer Graphics and Image Processing*, 15: 201-223.
- Forrester, W. D. (1983). *Canadian Tidal Manual*. Department of Fisheries and Oceans, Canadian Hydrographic Service, Ottawa, Ontario. 138 p.
- Fortier, M.-F. A.; Ziou, D.; Armenakis, C.; Wang, S. (2001). Automated Correction and Updating of Road Databases from High-Resolution Imagery. *Canadian Journal of Remote Sensing*, 27 (1): 76-89.

- Frei, W. and Chen, C.-C. (1977). Fast Boundary Detection: A Generalization and a New Algorithm. *IEEE Transactions on Computers*, C26 (10): 988-998.
- Furmanczyk, K. and Musielak, S. (1995). Remote Sensing of the Polish Coasts Morphology. **In:** Dally W. R. and Zeidler, R. B. (eds.). *Coastal Dynamics '95. Proceedings of the International Conference on Coastal Research in Terms of Large Scale Experiments*. Sept. 4-8, 1995. pp.1018-1023.
- Galbiati, L. J. (1990). *Machine Vision and Digital Image Processing Fundamentals*. Prentice-Hall, Inc., New Jersey. 164 p.
- Getis, A.; Getis, J.; Fellmann, J. (1991). *Introduction to Geography*, 3rd ed. Wm. C. Brown Publishers, Iowa. 469 p.
- Gierloff-Emden, H. G. (1977). *Orbital Remote Sensing of Coastal and Offshore Environments – A Manual of Interpretation*. Walter de Gruyter & Co., Berlin. 176 p.
- Gong, P. and Wang, J. (1996). Road Network Extraction from Airborne Digital Camera Data. *Proceedings of the Second International Airborne Remote Sensing Conference and Exhibition*, San Francisco, California, USA, 24-27 June 1996. pp. III387-III395.
- Greanias, E. C.; Meagher, P. F.; Norman, R. J.; Essinger, P. (1963). The Recognition of Handwritten Numerals by Contour Analysis. *IBM Journal of Research and Development*, 7: 14-21.
- Gross, M. G. (1972). *Oceanography: A View of the Earth*. Prentice-Hall, Inc., New Jersey. 497 p.
- Heywodd, I; Cornelius, S; Carver, S. (1998). *An Introduction to Geographical Information Systems*. Addison Wesley Longman Limited, New York. 279 p.
- Howard, A. D. (1939). Hurricane Modification of the Offshore Bar off Long Island, New York. **In:** El-Ashry, M. T. (ed.) (1977). *Air Photography and Coastal Problems*. Benchmark Papers in Geology, 38: 134-149. Dowden, Hutchinson & Ross, Inc., Pennsylvania.
- Hui, A.; Liew, S. C.; Kwoh, L. K. (2001). Extraction of Linear Features in Multispectral Imagery. **In:** *IEEE 2001 International Geoscience and Remote Sensing Symposium. (IGARSS 2001)*, University of New South Wales, Sydney, Australia, 9-13 July 2001.
- Jain, A. K. (1989). *Fundamentals of Digital Image Processing*. Prentice-Hall, Inc., New Jersey. 569 p.
- Jensen, J. R. (1996). *Introductory Digital Image Processing: A Remote Sensing Perspective*, 2nd ed. Prentice-Hall, Inc., New Jersey. 318 p.

- Keating, P. N.; Sawatari, T.; Zilinskas, G. (1979). Signal Processing in Acoustic Imaging. *Proceedings of the Institute of Electrical and Electronics Engineers*, 67: 496-510.
- Kelly, M. D. (1971). Edge Detection in Pictures by Computer Using Planning. **In:** Meltzer, B and Michie, D. (eds.). *Machine Intelligence 6*. American Elsevier Publishing Company, Inc. New York. pp. 397-409.
- Kirsch, R. A. (1971). Computer Determination of the Constituent Structure of Biological Images. *Computers and Biomedical Research*, 4 (3): 315-328.
- Kisslo, J; von Ramm, O. T.; Thurstone, F. L (1976). Cardiac Imaging Using a Phased Array Ultrasound System. II. Clinical Technique and Application. *Circulation* 53: 262-267.
- KODAK (2002). Sensitometric and Image Structure Data web-site: <http://www.kodak.ch/country/US/en/motion/support/h1/structureP.shtml> .
- Kruskal Jr, J. B. (1956). On the Shortest Spanning Subtree of a Graph and the Travelling Salesman Problem. *Proceedings of the American Mathematical Society*, 7: 48-50.
- Laing, D. (1995). *The Earth System – An Introduction to Earth Science*. Wm. C. Brown Publishers, Iowa. 590 p.
- Ledley, R. S. (1964). High-speed Automatic Analysis of Biomedical Pictures. *Science*, 146: 216-223.
- Lillesand, T. M and Kiefer, R. W. (1994). *Remote Sensing and Image Interpretation*, 3rd ed. John Wiley & Sons, Inc., New York. 750 p.
- Lyon, D. (1999). *Image Processing in Java*. Prentice Hall PTR, New Jersey. 532 p.
- Lucke, J. B. (1934). A Study of Barnegat Inlet. **In:** El-Ashry, M. T. (ed.) (1977). *Air Photography and Coastal Problems*. Benchmark Papers in Geology, 38: 119-133. Dowden, Hutchinson & Ross, Inc., Pennsylvania.
- MacDonald, H. C.; Lewis, A. J.; Wing, R. S. (1971). Mapping and Landform Analysis of Coastal Regions with Radar. **In:** El-Ashry, M. T. (ed.) (1977). *Air Photography and Coastal Problems*. Benchmark Papers in Geology, 38: 86-98. Dowden, Hutchinson & Ross, Inc., Pennsylvania.
- Magoon, O. T. (1973). Use of Earth Resources Technology Satellite (ERTS-1) in Coastal Studies. **In:** El-Ashry, M. T. (ed.) (1977). *Air Photography and Coastal Problems*. Benchmark Papers in Geology, 38: 99-107. Dowden, Hutchinson & Ross, Inc., Pennsylvania.

- Marr, D and Hildreth, E. (1980). Theory of Edge Detection. *Proceedings of the Royal Society of London*, B (207): 187-217.
- Martelli, A. (1972). Edge Detection Using Heuristic Search Methods. *Computer Graphics and Image Processing*, 1: 169-182.
- Martelli, A. (1976). An Application of Heuristic Search Methods to Edge and Contour Detection. *Communications of the ACM*, 19 (2): 73-83.
- Mason, S. J and Clemens, J. K. (1968). Character Recognition in an Experimental Reading Machine for the Blind. **In:** Kolers, P. A. (ed). *Recognizing Patterns: Studies in Living and Automatic Systems*. MIT Press. pp. 156-167.
- Moffitt, F. H. (1969). History of Shore Growth from Aerial Photographs. **In:** El-Ashry, M. T. (ed.) (1977). *Air Photography and Coastal Problems*. Benchmark Papers in Geology, 38: 51-55. Dowden, Hutchinson & Ross, Inc., Pennsylvania.
- Moik, J. G. (1980). *Digital Processing of Remotely Sensed Images*. NASA, Washington, D.C. 330 p.
- Mollard, J. D. (1960). *Airphoto Analysis and Interpretation. Short Course*. University of Alberta. 70 p.
- NGS (1999). *National Geodetic Survey* web-site: <http://www.ngs.noaa.gov/> .
- Nichols, S. E. (1983). *Tidal Boundary Delimitation*. UNB Technical Report #103. Department of Survey Engineering, University of New Brunswick. 202 p.
- Nilsson, N. J. (1971). *Problem-solving Methods in Artificial Intelligence*. McGraw-Hill, New York. 255 p.
- Nilsson, N. J. (1980). *Principles of Artificial Intelligence*. Tioga Publishing Co., Palo Alto, California. 476 p.
- NOAA (2001). *National Oceanic and Atmospheric Administration* web-site: <http://www.csc.noaa.gov/shoreline/term.html> .
- NOS (1999). *NOS Hydrographic Surveys Specifications and Deliverables*. NOAA, U.S. Department of Commerce, Washington, D.C. Web-site: <http://chartmaker.ncd.noaa.gov/hsd/specs/specs.htm> .
- O'Reilly, C. (2000). Defining the Coastal Zone from a Hydrographic Perspective. *Backscatter*, 11 (2): 20-25.
- Onana, V. P.; Ngonu, J. M.; Trebossen, H.; Rudant, J. P.; Tonye, E. (2001). Coastline Detection in SAR Images Using Texture Analysis in Textural or Geometrical Multi-resolution. **In:** *IEEE 2001 International Geoscience and Remote Sensing*

Symposium. (IGARSS 2001), University of New South Wales, Sydney, Australia, 9-13 July 2001.

- PCI (1997). *Using PCI Software – Version 6.2*. Volumes 1 and 2. PCI Geomatics, Inc. Richmond Hill, Ontario. 551 p.
- Pingle, K. K. (1969). Visual Perception by a Computer. **In:** Grasselli, A (ed.). *Automatic Interpretation and Classification of Images*. Academic Press, New York. pp. 277-284.
- Prewitt, J. M. S. (1970). Object Enhancement and Extraction. *Symposium on Psychopictorics*, Arlington, Virginia, 1969: 75-149.
- Province of British Columbia. (1993). Web-site: <http://www.for.gov.bc.ca/het/Clayquot/t-toc.htm> .
- Richards, J. A. (1986). *Remote Sensing Digital Image Analysis*. Springer-Verlag, Berlin. 281 p.
- Roberts, L.G. (1965). Machine Perception of Three-Dimensional Solids. **In:** Tippet, J.T., (ed.), *Optical and Electro-Optical Information Processing*. MIT Press, Cambridge, MA. pp. 159-197.
- Robinson, G. S. (1977). Edge Detection by Compass Gradient Masks. *Computer Graphics and Image Processing*, 6 (5): 492-501.
- RSI (2002). *RADARSAT International* web-site: <http://www.rsi.ca> .
- Russ, J. C. (1992). *The Image Processing Handbook*. CRC Press, Inc., Boca Raton, Florida. 445 p.
- Schalkoff, R. J. (1989). *Digital Image Processing and Computer Vision*. John Wiley & Sons, Inc., New York. 489 p.
- Schiller, N. B. and Silverman, N. H. (1978). Two-dimensional Ultrasonic Cardiac Imaging. **In:** Kleid, J. J. and Arvan, S. B. (eds.). *Echocardiography: Interpretation and Diagnosis*. Appleton-Century-Crofts, New York. pp 365-421.
- Schowengerdt, R. A. (1983). *Techniques for Image Processing and Classification in Remote Sensing*. Academic Press, New York. 249 p.
- Sebert, L.M., and Munro M. R. (1972). *Dimensions and Areas of Maps of the National Topographic System of Canada*. Technical Report 72-1. Ottawa: Department of Energy, Mines and Resources, Surveys and Mapping Branch. Web-site: <http://atlas.gc.ca/site/english/facts/coastline.html> .
- Shalowitz, A. L. (1964). *Shore and Sea Boundaries. Volumes 1 and 2*. U.S. Department of Commerce Publication 10-1, Washington, DC. 749 p.

- Shepard, F. P. and Wanless, H. R. (1971). *Our Changing Coastlines*. McGraw-Hill, Inc., New York. 579 p.
- SHOALS (2001). *Scanning Hydrographic Operational Airborne Lidar Survey System* web-site: <http://shoals.sam.usace.army.mil/> .
- Skorton, D. J.; McNary, C. A.; Child, J. S.; Newton, F. C.; Shah, P. M. (1981). Digital Image Processing of Two-Dimensional Echocardiograms: Identification of the Endocardium. *The American Journal of Cardiology*, 48: 479-486.
- Sobel, I. (1970). *Camera Models and Machine Perception*. PhD Dissertation, Stanford University. 89p.
- Steger, C.; Eckstein, W.; Wiedemann, C. (1996). Update of Roads in GIS by Automatic Extraction from Aerial Imagery. *Proceedings of the Second International Airborne Remote Sensing Conference and Exhibition*, San Francisco, California, USA, 24-27 June 1996. pp. III308–III317.
- Strahler, A. N. and Strahler, A. H. (1978). *Modern Physical Geography*. John Wiley & Sons, Inc., New York. 502 p.
- Thurman, H. V. (1994). *Introductory Oceanography*, 7th ed. MacMillan Publishing Company, New York. 550 p.
- TRIM (1992). *British Columbia Specifications and Guidelines for Geomatics Content Series*, Volume 3 – Digital Baseline Mapping at 1:20,000. Ministry of Environment, Lands and Parks – Geographical Data BC – Province of British Columbia. Web-site: <http://home.gdbc.gov.bc.ca/TRIM> .
- UN (1992). *Report of the United Nations Conference on Environment and Development, Rio de Janeiro, 3-14 June*. A/CONF.151/26 (Vol. I) August 12. United Nations, New York. Web-site: <http://www.un.org/documents/ga/conf151/aconf15126-1.htm> .
- Weeks, A. R. (1996). *Fundamentals of Electronic Image Processing*. SPIE Optical Engineering Press, New York. 570 p.
- Zadeh, L. A. (1965). Fuzzy Sets. *Information and Control*, 8: 338-353.
- Zahn, C. T. (1969). A Formal Description for Two-Dimensional Patterns. *Proceedings of the International Joint Conference on Artificial Intelligence, May 1969*. pp. 621-628.
- Zwehl, W.; Levy, R.; Garcia, E.; Haendchen, R. V.; Childs, W.; Corday, S. R.; Meerbaum, S.; Corday, E. (1983). Validation of a Computerized Edge Detection Algorithm for Quantitative Two-Dimensional Echocardiography. *Circulation*, 68 (5): 1127-1135.

APPENDIX A

The following code is contained in a single file. It is part of the Kahindu image processing program created by Douglas A. Lyon (1999). Kahindu was compiled and run on the Java™ 2 Platform, Standard Edition (J2SE™) version 1.3.1_03 obtained from Sun Microsystems, Inc.

MartelliFrame.java performs a heuristic search for an open node with a minimal cost, as explained in Chapter 3.

Code name: MartelliFrame.java

Created by Douglas A. Lyon – September 1998

Modified by Eduardo A. Loos – October 2000

```
package gui;
import java.util.*;
import java.awt.*;
import java.awt.event.*;
import java.math.*;

public class MartelliFrame extends PaintFrame {
    private Vector nodes = new Vector();
    // a list of all edges
    private Vector expanded = new Vector();

    private int MaxI=0;
    //Point Begin;
    //Point End;

    // The following number is image specific.
    // Martelli only finds one countour at a time.
```

```

// on multiple runs, you will
// alter the following.
public static final int maximumNumberOfEdges = 15000;
public static final int edgeLength = 20;

/**I have set the search time in seconds here,√ - EAL
public static final int runTimeInSeconds = 10;

private Timer t = new Timer();

Menu heuristicMenu = new Menu("Heuristic Edge Detection");
MenuItem processUserPoints_mi =
    addItem(heuristicMenu, "[E-M]process user points");
MenuItem negativeRobertsOnGreen_mi =
    addItem(heuristicMenu, "use NegativeRobertsOnGreen");
MenuItem averageWithChild_mi =
    addItem(fileMenu, "averageWithChild");

MenuItem printPath_mi = addItem(heuristicMenu, "printPath");

// Inputs the green plane and outputs edges found on the
// red plane. The red plane is cleared upon initialization.

public MartelliFrame(String title) {
    super(title);
    boundaryMenu.add(heuristicMenu);
}

public static void main(String args[]) {
    MartelliFrame xf =new MartelliFrame("MartelliFrame");
}

//evaluates whether Martelli should stop searching or not
private boolean terminateSearch(Edgel el){
    //if (edges.size() > maximumNumberOfEdges ) return true;
    if (t.getElapsedTime() > runTimeInSeconds)
        return true;
    if (distance(el,endPoint) < 2) return true;
    return false;
}

// Finds out whether the current edge coordinates
// have been visited

```

```

Edgel getMarked(Edgel e) {
    for(int i=0; i < nodes.size(); i++) {
        Edgel el = (Edgel) nodes.elementAt(i);
        if ((el.p1.x==e.p1.x) &&
            (el.p1.y==e.p1.y) &&
            (el.p2.x==e.p2.x) &&
            (el.p2.y==e.p2.y)) {
                return el;
            }
    }
    return null;
}

//The cost of an edge element
private int C2(Edgel e) {
    int pc = 0;
    Edgel p = e.getParent();
    int d = distance(e,endPoint);
    if (p != null) ;/pc = p.getCost()/e.getPly();
    return
        clip(pc+
            d+MaxI-(
                g[e.p1.x][e.p1.y]-
                g[e.p2.x][e.p2.y]
            )
        );
}
// assume that local cost is stored in the
// green plane.
private int C(Edgel e) {
    int pc = 0;
    Edgel p = e.getParent();
    int d = distance(e,endPoint);
    //if (p != null) pc = p.getCost()-e.getPly();
    return
        clip(pc+
            d+child.g[e.p1.x][e.p1.y]);
}

public void averageWithChild() {
    for (int x=0; x < width; x++)
        for (int y=0; y < height; y++) {
            r[x][y] = (short)((r[x][y] + child.r[x][y]) / 2);
            g[x][y] = (short)((r[x][y] + child.g[x][y]) / 2);
            b[x][y] = (short)((r[x][y] + child.b[x][y]) / 2);
        }
    short2Image();
}

```

```

}

private void negativeRobertsOnGreen() {
    int p[] = new int[4];
    float delta_u = 0;
    float delta_v = 0;
    for (int x=0; x < width-1; x++)
        for (int y=0; y < height-1; y++) {
            p[0] = r[x][y];
            p[1] = r[x+1][y];
            p[2] = r[x][y+1];
            p[3] = r[x+1][y+1];
            delta_u = p[0] - p[3];
            delta_v = p[1] - p[2];
            g[x][y] =
                (short) (255- Math.sqrt(delta_u*delta_u + delta_v*delta_v));
        }
}

```

```

private int clip(int x) {
    if ( x < 0 ) return 0;
    return x;
}

```

```

//Euclidean Distance
private int distance(Edgel e, Point p) {
    int dx = e.p2.x - p.x;
    int dy = e.p2.y - p.y;
    return (int)Math.sqrt(dx*dx + dy*dy);
}

```

```

//adds an edge element to the 'expanded' list,
//if it is within the boundaries of the image
private void expAdd(int x1, int y1, int x2, int y2, Edgel parent ) {
    if (
        (x1>=0)&&(x1<width)&&
        (x2>=0)&&(x2<width)&&
        (y1>=0)&&(y1<height)&&
        (y2>=0)&&(y2<height)) {
        Edgel e = new Edgel();
        e.setCoordinates(new Point(x1,y1), new Point(x2,y2));
        e.setParent(parent);
        e.setCost(C(e));
        expanded.addElement(e);
    }
}

```

```

}

//expands the current edge element in
// three directions, depending on its
//orientation
private void expand(Edgel el){
    int choice=0;
    int x1, y1, x2, y2;
    expanded = new Vector();
    x1=el.p1.x;
    y1=el.p1.y;
    x2=el.p2.x;
    y2=el.p2.y;
    int dx = x2 - x1;
    int dy = y2 - y1;

    if (dx>0) choice=1;
    if (dx<0) choice=2;
    if (dy>0) choice=3;
    if (dy<0) choice=4;
    if (choice == 0) {
        System.out.println("This should never happen!! Choice=0!!!");
    }
    //E = current edge
    switch( choice){
        case 1:
            //      E
            //    --- ---
            //      |
            expAdd( x1, y1, x1,y1+1,el);
            expAdd( x1,y1+1,x1+1,y1+1,el);
            expAdd(x1+1,y1+1,x1+1, y1,el);
            break;
        case 2:
            //      |
            //    --- ---
            //      E
            expAdd( x1, y1, x1,y1-1,el);
            expAdd( x1,y1-1,x1-1,y1-1,el); //dd
            expAdd(x1-1,y1-1,x1-1, y1,el);
            break;
        case 3:
            //      |
            //    --- E
            //      |
            expAdd( x1, y1,x1-1, y1,el); //dd

```

```

        expAdd(x1-1, y1,x1-1,y1+1,el);
        expAdd(x1-1,y1+1, x1,y1+1,el);
        break;
    case 4:
        //      |
        //     E ---
        //      |
        expAdd( x1, y1,x1+1, y1,el);
        expAdd(x1+1, y1,x1+1,y1-1,el);
        expAdd(x1+1,y1-1, x1,y1-1,el); //dd
        break;
    }
}

//returns the edge with the lower cost that has been marked as OPEN
public Edgel minOpenNode(){
    if (nodes.size() == 0) return null;
    int minCost = 1000000;
    Edgel lowCostNode = new Edgel();
    lowCostNode.setType(Edgel.OPEN);
    lowCostNode.setCost(minCost);
    for (int i=0; i< nodes.size(); i++){
        Edgel el = (Edgel) nodes.elementAt(i);
        if (el.getType() != Edgel.OPEN) continue;
        if (el.getCost() > lowCostNode.getCost()) continue;
        lowCostNode=el;
    }
    if (lowCostNode.getCost() == minCost) return null;
    return lowCostNode;
}

public int countPath(Edgel pos){
    int i=0;
    while (pos != null){
        i++;
        pos=pos.getParent();
    }
    return i;
}

public Polygon getPath(Edgel pos){
    Polygon p = new Polygon();
    while (pos != null){
        p.addPoint(pos.p2.x,pos.p2.y);
        pos=pos.getParent();
    }
}

```

```

    }
    return p;
}

public void drawPath(Edgel pos) {
    Polygon p = getPath(pos);
    //printPoly(p);
    drawPoly(p);
}

//clears the red plain so that the edge can be draw with it
//public void clearRedPlane2() {
//    r = new short[width][height];
//}

//returns the maximum difference between two adjacent points in the image
// --key number for the cost function
private int computeMaxI() {
    int maxi = -10;
    for(int i=0;i<width-1;i++)
        for (int j=0;j<height-1;j++){
            int tmpdiff=Math.abs(g[i][j]-g[i+1][j]);
            if (tmpdiff>maxi) maxi=tmpdiff;
        }
    return maxi;
}

private void initialize() {
    nodes = new Vector();
    expanded = new Vector();
    MaxI=0;
}

private void processUserPoints(Vector v){
    if (v.size() < 2) {
        System.out.println("Select start and end point(s)");
        return;
    }
    if (child == null) {
        System.out.println("You need a child frame for a cost matrix"
            + "\ncreating one for you!");
        copyToChildFrame();
        child.roberts2();
        child.negate();
    }
    setPolyList( new Vector());
}

```

```

    Point r = (Point) v.elementAt(0);
    Vector polyList = getPolyList();
    for (int i = 1; i < v.size(); i++) {
        Point r2 = (Point) v.elementAt(i);
        endPoint = new Point(r2.x,r2.y);
        Polygon p1 = getPath(
            searchFromPoint(new Point(r.x,r.y)));
        r = r2;
        polyList.addElement(p1);
    }
}

private Point endPoint = null;

// the A* algorithm
public Edgel searchFromPoint(Point startPoint){
    initialize();
    Edgel lowestCostNode;

    //get Maxi, the highest difference between two adjacent pixels
    MaxI = computeMaxI();
    //A* starts here
    Edgel s = new Edgel();
    s.setCoordinates(startPoint,
        new Point(
            startPoint.x+1,
            startPoint.y));
    s.setType(Edgel.OPEN);
    nodes.addElement(s);
    while ((lowestCostNode = minOpenNode()) != null) {
        if (terminateSearch(lowestCostNode)) break;
        lowestCostNode.setType(Edgel.CLOSED);
        expand(lowestCostNode);
        if (expanded.size() == 0) continue;
        processExpandedNodes();
        drawTracks(lowestCostNode);
    }
    lowestCostNode = minOpenNode();
    //System.out.println("ply for lcn="+lowestCostNode.getPly());
    //System.out.println("cost for lcn="+lowestCostNode.getCost());
    return lowestCostNode;
} // end martelli

// Let the user see the search run...
// great fun!

```

```

private void drawTracks(Edgel lowestCostNode) {
    if (child == null) System.out.println(
        "You need a child frame for a cost matrix");
    Graphics g = child.getGraphics();
    g.setColor(Color.red);
    g.setXORMode(Color.red);
    g.drawOval(lowestCostNode.p1.x-1,lowestCostNode.p1.y-1,1,1);
}

```

```

private void processExpandedNodes() {
    for (int i=0; i < expanded.size();i++){
        Edgel e = (Edgel) expanded.elementAt(i);
        Edgel MarkedNode=getMarked(e);
        if (MarkedNode==null) {
            nodes.addElement(e);
            continue;
        }
        if (e.getCost()<MarkedNode.getCost())
            MarkedNode = e;
    }
}

```

```

private Point getMaxPoint() {
    int max = -10;
    Point p=null;
    for (int x=0; x < width-1; x++)
        for (int y=0; y < height; y++) {
            if (Math.abs(g[x][y]-g[x+1][y]) > max) {
                p = new Point(x,y);
                max = Math.abs(g[x][y]-g[x+1][y]);
            }
        }
    return p;
}

```

```

private Edgel pathStart = null;

```

```

public void actionPerformed(ActionEvent e) {


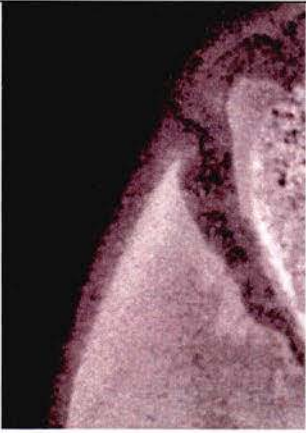




    if (match(e, averageWithChild_mi)) {
        averageWithChild();
        return ;
    }
    if (match(e, negativeRobertsOnGreen_mi)) {

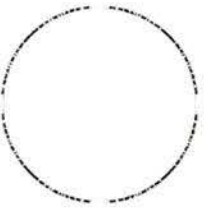



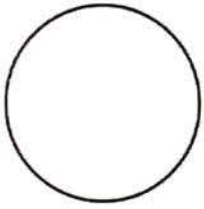

```

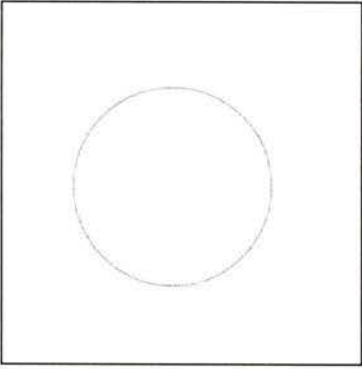

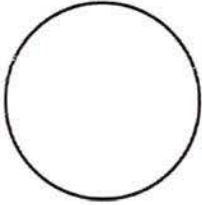

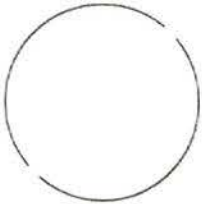

```
        negativeRobertsOnGreen();
        return ;
    }
    if (match(e, printPath_mi)) {
        drawPath(pathStart);
        return ;
    }
    if (match(e, processUserPoints_mi)) {
        System.out.println("shapes.size()="+shapes.size());
        t.start();
        processUserPoints(shapes);
        t.print("Martelli done");
        drawPolys();
        return ;
    }
    super.actionPerformed(e);
}
}
```

APPENDIX B

Effects of Spatial Filters on Images, as explained in Chapter 2.

Spatial Filter	Control Image	Aerial Orthophoto (1 m)
Original		
Average		
Gaussian 7x7		

Spatial Filter	Control Image	Aerial Orthophoto (1 m)
Roberts		
Sobel		
Prewitt		

Spatial Filter	Control Image	Aerial Orthophoto (1 m)
Canny		
Frei-Chen		
Pixel Difference		

APPENDIX C

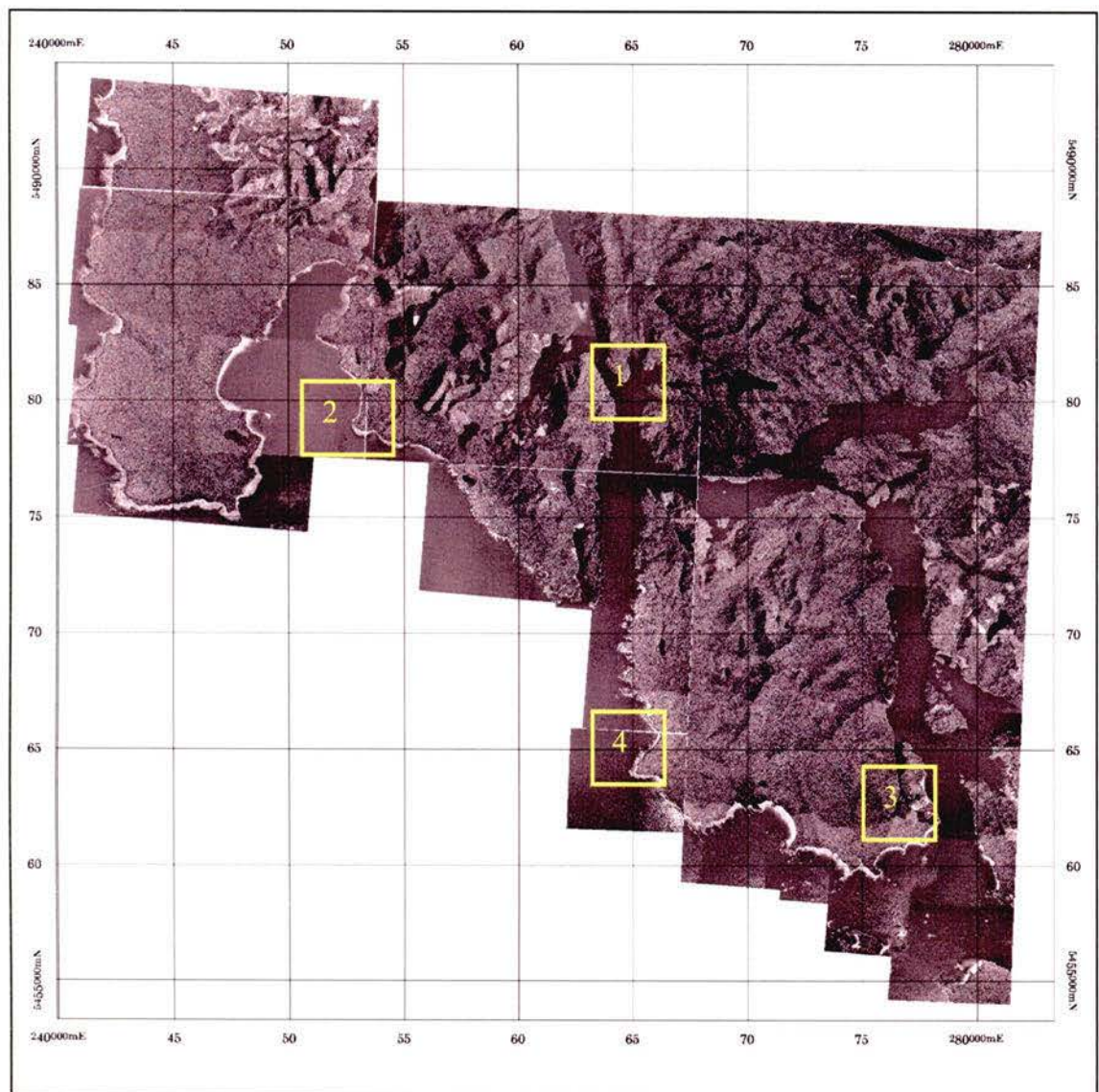
Area of Study: Clayoquot Sound – Vancouver Island - BC

Mosaic of 1-m aerial orthophotos

Projection: UTM

Datum: North American Datum 1983 (Canada)

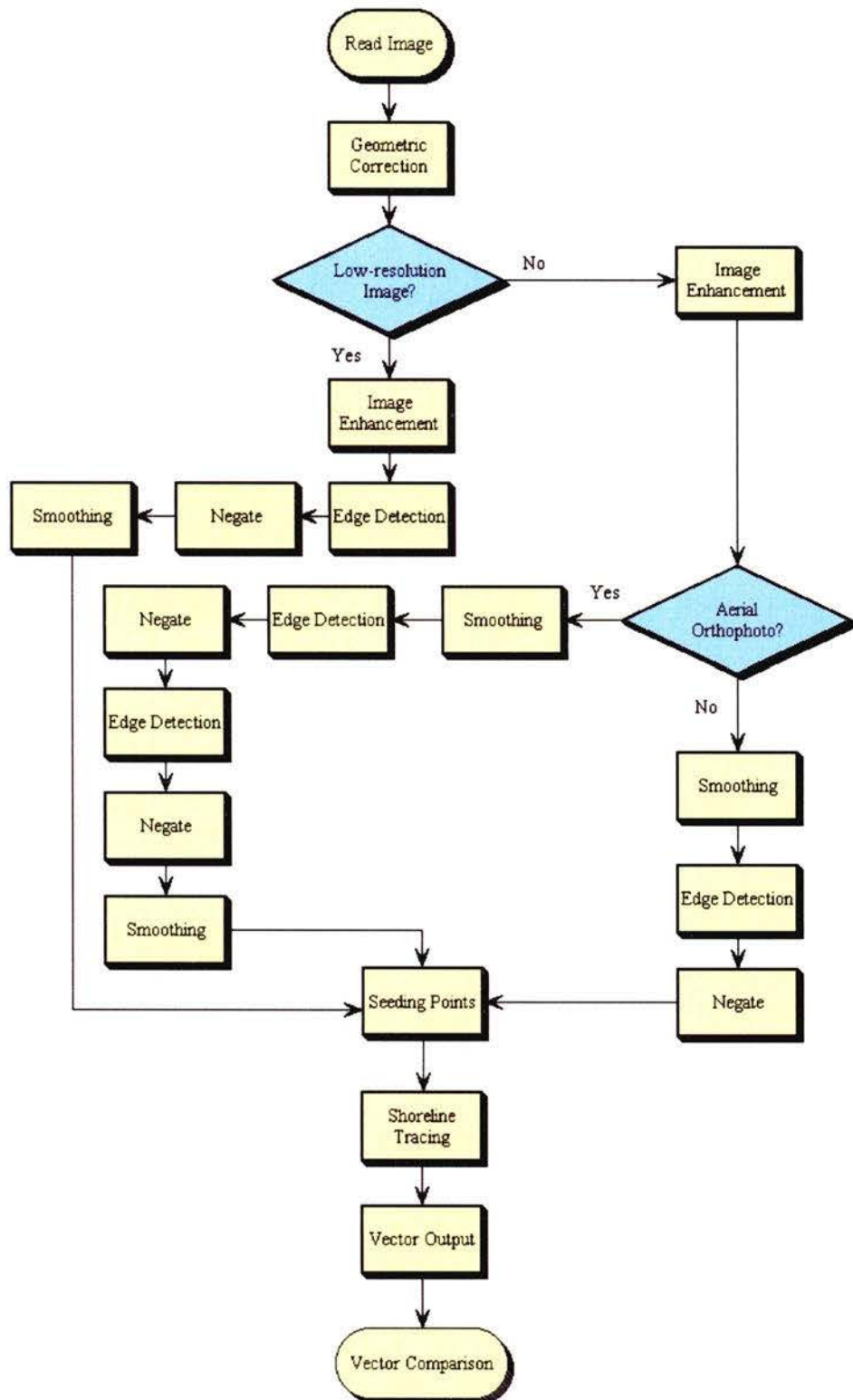
Yellow rectangles indicate areas mentioned in Chapter 4: 1) Coast with vegetation line; 2) Sandy beaches; 3) Mudflats; and 4) Rocky coast.



Mosaicked by Eduardo A. Loos using GCPWorks (PCI Geomatics Inc., Richmond Hill, Ontario).

APPENDIX D

Image Processing and Tracing Algorithm.



VITA

Surname: Araujo Loos

Given Name: Eduardo

Place of Birth: Rio de Janeiro, Rio de Janeiro, Brazil

Educational Institutions Attended:

University of Victoria (UVic)	1999 to 2002
Universidade do Estado do Rio de Janeiro (UERJ)	1993 to 1997
Swiss School – Rio de Janeiro	1979 to 1992

Degrees Awarded:

B.Sc. (Honours) in Oceanography Universidade do Estado do Rio de Janeiro	1997
---	------

Honours and Awards:

UVic Graduate Research and Teaching Fellowship	1999 to 2002
PCI Geomatics Award of GeoIntelligence	2001
UERJ Research Assistantship	1995 to 1996

Publications:

"The Effect of the Ocean Circulation on Shark (Chondrichthyes, Elasmobranchii) Distribution Patterns along the Brazilian Coast". B.Sc. Honours Thesis, Universidade do Estado do Rio de Janeiro. 1997

UNIVERSITY OF VICTORIA PARTIAL COPYRIGHT LICENSE

I hereby grant the right to lend my thesis to users of the University of Victoria Library, and to make single copies only for such users or in response to a request from the Library of any other university, or similar institution, on its behalf or for one of its users. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by me or a member of the University designated by me. It is understood that copying or publication of this thesis for financial gain by the University of Victoria shall not be allowed without my written permission.

Title of Thesis:

Shoreline Feature Extraction from Remotely-Sensed Imagery

Author



Eduardo Araujo Loos

August 8th, 2002