To the Graduate Council:

I am submitting herewith a dissertation written by Renato Alberto Salinas entitled "REGULARIZED FUSION: GRAY AND COLOR EDGE DETECTION AND SURFACE RECONSTRUCTION." I have examined the final copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Electrical Engineering.

M. A. Abidi, Major Professor

We have read this dissertation
and recommend its acceptance:

R. C. Gonzalez

Edmund G. Zelnio

J. W. Bowles

Marshall Pace

Accepted for the Council:

Associate Vice Chancellor
and Dean of The Graduate School

REGULARIZED FUSION: GRAY AND COLOR EDGE DETECTION AND SURFACE RECONSTRUCTION

A Dissertation

Presented for the

Doctor of Philosophy

Degree

The University of Tennessee, Knoxville

Renato Alberto Salinas

August 1994

DEDICATION

This dissertation is dedicated to my wife, Roxana, our twins, Carlos and Joanna, my mother Mrs. Fresia Silva, my father Mr. Renato Salinas, and my sisters Chechy and Verónica. Their love and encouragement have been fundamental in my life.

ACKNOWLEDGEMENTS

I would like to thank Professor M. A. Abidi for guiding me through my dissertation, for supporting and advising me, for keeping me focused on the right issues, and for allowing me to work in his research group.

I extend my gratitude to the other members of my committee, Professors D. W. Bouldin, D. Brzakovic, R. C. Gonzalez, E. G. Harris, and M. O. Pace for reading and revising this document. I would like to specially thank Professors Gonzalez and Pace for their help during the admissions process to the Ph.D. program. Additionally, Professor Gonzalez's help has been invaluable for securing extensions from my government.

I would like to thank Professors V. Alexiades and D. Hinton of the Department of Mathematics for their technical advice. I would like to mention the generous help received from Dr. J. Googe, former head of the Electrical and Computer Engineering Department and Dr. C. W. Minkel, , and vice–chancellor and dean of the Graduate School. I would also like to acknowledge Dr. C. Halloy of JICS for his help and for our many discussions.

There are many graduate students who have shared discussions with me and have made my experience as a student more complete. Those include, Dr. M. Abdoulghafour, Dr. A. Perez, Dr. A. Sari-Sarraf, C. Tan, Dr. C. Chen, M. Baccar, L. Gee, A. Dalmia, and many others.

Various organizations also deserve credit, including the University of Santiago

of Chile, the Organization of American States, and the Ministry of Planning and Development of Chile.

I am very grateful to my wife Roxana Moreira and our children, Carlos and Joanna, for leaving home and family in pursuit of my dream.

Last, but not least, I should mention my colleague and friend Ricardo Hinojosa from the University of Santiago and his support throughout these years.

ABSTRACT

The need to extract useful information from multisensor data exists for both biological and artificial systems. Multisensor data fusion is a crucial component for the success of a variety of applications, including intelligent weapon systems, autonomous robots, and advanced medical systems.

The area of data fusion provides tools for solving problems which are characterized by distributed and diverse information sources. In this dissertation, we focus on the problem of extracting given features, such as image discontinuities, from images obtained using different sensing modalities. Since edge detection is an ill–posed problem in the sense of Hadamard, Tikhonov's regularization paradigm has been proposed as a basic tool to solve this inversion problem and to restore well–posedness. The proposed framework includes: (i) a systematic view of one–dimensional as well as two–dimensional regularization, (ii) an evaluation (weighting) of the knowledge sources by considering individual noise characteristics, (iii) extension of the standard Tikhonov regularization method by allowing space–variant regularization parameters, and (iv) further extension of the regularization paradigm by adding, in a natural way, multiple data sources and allowing data fusion.

The theoretical approach has been complemented by developing a series of algorithms and then solving various early vision problems, including regularized edge detection, surface reconstruction, and color edge detection.

TABLE OF CONTENTS

CHAPTER		PAGE
1.	**INTRODUCTION**	**1**
1.1	Problem Statement	5
1.2	Overview of the Dissertation	6
2.	**BACKGROUND**	**7**
2.1	Methods for Inferring Knowledge	7
2.2	MultiSensor Fusion: Applications	11
2.3	Surface Reconstruction Techniques	19
2.4	Color Edge Detection Techniques	31
2.5	Roof Edge Detection Techniques	40
3.	**PROBLEM FORMULATION**	**46**
3.1	Regularized Edge Detection	46
3.2	One–Dimensional Case	48
3.3	Two–Dimensional Case	50
3.4	Boundary Conditions: One–Dimensional Case	52
3.5	Surface Reconstruction	56
3.6	Color Edge Detection	67
3.7	Roof Edge Detection Scheme	70

CHAPTER PAGE

4. EXPERIMENTAL RESULTS 73

4.1 Regularization Results for the One–Dimensional Case for Various Boundary Conditions 74

4.2 Regularization Results for the Different One–Dimensional Functionals 82

4.3 Gray Edge Detection Results 103

4.4 Surface Reconstruction Results 124

4.5 Color Edge Detection Results 131

4.6 Roof Edge Detection Results 142

4.7 Scale–Space for the First– and Second–Order Systems 145

4.8 Effect of Noise on the First– and Second–Order Systems 149

5. CONCLUSION 156

5.1 Summary and Contributions 156

5.2 Recommendations for Future Research 158

BIBLIOGRAPHY 161

VITA 167

LIST OF TABLES

TABLE		PAGE
3.1	Valid corner conditions	61
3.2	Conditions along exterior edges	61

LIST OF FIGURES

FIGURE		PAGE
2.1	*1–D smoothing network.*	24
2.2	*Regularization network.*	27
3.1	*Computational molecule for interior nodes.*	63
3.2	*Computational molecules for corner nodes.*	64
3.3	*Computational molecules for nodes along exterior rows.*	65
3.4	*Computational molecules for nodes along exterior rows one pixel away from the corner.*	65
3.5	*Computational molecules for nodes along rows adjacent to exterior rows at least two pixels away from opposite sides.*	66
3.6	*Computational molecules for "corner" nodes one pixel away from the real corners.*	67
3.7	*Color image irradiance and photometric angles for the isotropic case.*	69
3.8	*Roof edge detection: (a) roof edge (solid line) and its smooth approximation, (b) error signal (solid) and its derivative.*	71
3.9	*Algorithm for roof edge detection by orthogonal scans.*	72
4.1	*1–D regularization for natural boundary conditions (Case 1) and small λ.*	76
4.2	*1–D regularization for natural boundary conditions (Case 1), large λ.*	77

FIGURE **PAGE**

4.3 *1–D regularization for Dirichlet boundary conditions (Case 2), small* λ. 77

4.4 *1–D regularization for Dirichlet boundary conditions (Case 2), large* λ. 78

4.5 *1–D regularization for initial conditions (Case 3), reasonable* λ*'s.* 79

4.6 *1–D regularization for general boundary conditions (Case 4), small* λ. 79

4.7 *1–D regularization for general boundary conditions (Case 4), large* λ. 80

4.8 *1–D regularization for noisy step and general boundary conditions (Case 5), small* λ. 81

4.9 *1–D regularization for noisy step and general boundary conditions (Case 5), large* λ. 82

4.10 *1–D regularization for noisy step and natural boundary conditions (Case 6), small* λ. 83

4.11 *1–D regularization for noisy step and natural boundary conditions, (Case 6), large* λ. 84

4.12 *1–D fusion with first–order stabilizer and small* λ. 84

4.13 *1–D fusion with first–order stabilizer and medium* λ. 85

4.14 *1–D fusion with first–order stabilizer and large* λ. 85

4.15 *1–D fusion with first–order stabilizer, noisy data, and small* λ. . . 86

4.16 *1–D fusion with first–order stabilizer, noisy data, and medium* λ. 86

FIGURE **PAGE**

4.17 *1–D fusion with first–order stabilizer, noisy data, and large* λ. . . 87

4.18 *1–D fusion with second–order stabilizer and small* λ. 87

4.19 *1–D fusion with second–order stabilizer and large* λ. 88

4.20 *1–D fusion with second–order stabilizer, noisy data, and small* λ. 88

4.21 *1–D fusion with second–order stabilizer, noisy data, and big* λ. . 89

4.22 *One–dimensional regularization with varying regularization parameters. The input data is labeled "d", the regularized solution is "f", and the derivative of the solution is "fx".* 89

4.23 *Responses of the first functional (first–order stabilizer with fixed regularization parameter) and noise–free data: (a)* $\lambda = 0.1$, *(b)* $\lambda = 1.0$, *(c)* $\lambda = 10.0$. 90

4.24 *Responses of the first functional and noisy data: (a)* $\lambda = 0.1$, *(b)* $\lambda = 1.0$, *(c)* $\lambda = 10.0$. 91

4.25 *Responses of the second functional (first–order stabilizer and variable regularization parameter* $\lambda(x)$*): (a) noise–free response, (b) response under noisy conditions.* 93

4.26 *Responses of the second–order stabilizer with fixed regularization parameter under noise–free conditions: (a)* $\beta = 0.1$, *(b)* $\beta = 1.0$, *(c)* $\beta = 10.0$. 94

4.27 *Responses of the second–order stabilizer with fixed regularization parameter under noisy conditions: (a)* $\beta = 0.1$, *(b)* $\beta = 1.0$, *(c)* $\beta = 10.0$. 95

FIGURE **PAGE**

4.28 *Typical responses of the third functional (second-order stabilizer with variable regularization parameter): (a) noise-free data, (b) noisy data.* 96

4.29 *Typical responses of the fourth functional (a hybrid system with first- and second-order stabilizers with variable regularization parameters): (a) noise-free data, (b) noisy data.* 98

4.30 *Responses of the hybrid functional (first- and second-order stabilizers with fixed regularization parameters) under noise-free conditions: (a)* $\lambda = \beta = 0.1$, *(b)* $\lambda = 0.1$, $\beta = 1.0$; *(c)* $\lambda = 0.1$, $\beta = 10.0$; *(d)* $\lambda = 1.0$, $\beta = 0.1$; *(d)* $\lambda = \beta = 1.0$, *(f)* $\lambda = 1.0$, $\beta = 10.0$; *(g)* $\lambda = 10.0$, $\beta = 0.1$; *(h)* $\lambda = 10.0$, $\beta = 1.0$; *(i)* $\lambda = \beta = 10.0$. . . . 99

4.31 *Responses of the hybrid functional (first- and second-order stabilizers with fixed regularization parameters) under noisy conditions: (a)* $\lambda = \beta = 0.1$, *(b)* $\lambda = 0.1$, $\beta = 1.0$; *(c)* $\lambda = 0.1$, $\beta = 10.0$; *(d)* $\lambda = 1.0$, $\beta = 0.1$; *(d)* $\lambda = \beta = 1.0$, *(f)* $\lambda = 1.0$, $\beta = 10.0$; *(g)* $\lambda = 10.0$, $\beta = 0.1$; *(h)* $\lambda = 10.0$, $\beta = 1.0$; *(i)* $\lambda = \beta = 10.0$. . . . 100

4.32 *Typical result for the fifth functional (a mixed stabilizer system with two data sources) under noise-free conditions:* $\omega_1 = \omega_2$. 101

4.33 *Typical results for the fifth functional (a mixed stabilizer system with two data sources) under noisy conditions: (a)* $\omega_1 > \omega_2$, *(b)* $\omega_1 = \omega_2$, *(c)* $\omega_1 < \omega_2$. 102

FIGURE **PAGE**

4.34 *Square; fusion with first-order stabilizer: (a) input 1,* $\sigma_n = 5$*, (b) input 2,* $\sigma_n = 10$*, (c) zero-crossing map, (d) gradient edge map, (e) conventional zero-crossing on input 1, (f) conventional zero-crossing on input 2, (g) gradient on input 1, (h) gradient on input 2.* 104

4.35 *Cake; fusion with first-order stabilizer: (a) input 1,* $\sigma_n = 10$*, (b) input 2,* $\sigma_n = 15$*, (c) zero-crossing map, (d) gradient edge map, (e) conventional zero-crossing on input 1, (f) conventional zero-crossing on input 2, (g) gradient on input 1, (h) gradient on input 2.* 106

4.36 *Checkers; fusion with first-order stabilizer: (a) input 1,* $\sigma_n = 10$*, (b) input 2,* $\sigma_n = 20$ *(c) zero-crossing map, (d) gradient edge map, (e) conventional zero-crossing on input 1, (f) conventional zero-crossing on input 2, (g) gradient on input 1, (h) gradient on input 2.* 107

4.37 *Wake; fusion with first-order stabilizer: (a) input 1,* $\sigma_n = 10$*, (b) input 2,* $\sigma_n = 30$*, (c) zero-crossing map, (d) gradient edge map, (e) conventional zero-crossing on input 1, (f) conventional zero-crossing on input 2, (g) gradient on input 1, (h) gradient on input 2.* 108

FIGURE **PAGE**

4.38 *Synthetic objects; fusion with first-order stabilizer: (a) range image, (b) intensity image, (c) zero-crossing map, (d) gradient edge map, (e) conventional zero-crossing on input 1, (f) conventional zero-crossing on input 2, (g) gradient on input 1, (h) gradient on input 2.* 110

4.39 *Real intensity-range data; fusion with first-order stabilizer: (a) range image, (b) intensity image, (c) zero-crossing map, (d) gradient edge map, (e) conventional zero-crossing on input 1, (f) conventional zero-crossing on input 2, (g) gradient on input 1, (h) gradient on input 2.* 111

4.40 *Real infrared-intensity data; fusion with first-order stabilizer: (a) infrared image, (b) intensity image, (c) zero-crossing map, (d) gradient edge map, (e) conventional zero-crossing on input 1, (f) conventional zero-crossing on input 2, (g) gradient on input 1, (h) gradient on input 2.* 113

4.41 *Square; fusion with second-order stabilizer: (a) input 1,* $\sigma_n = 5$, *(b) input 2,* $\sigma_n = 10$, *(c) gradient on ideal image, (d) gradient edge map, (e) conventional zero-crossing on input 1, (f) conventional zero-crossing on input 2, (g) gradient on input 1, (h) gradient on input 2.* 115

4.42 *Cake; fusion with second-order stabilizer: (a) input 1,* $\sigma_n = 10$*, (b) input 2,* $\sigma_n = 15$*, (c) gradient on ideal image, (d) gradient edge map, (e) conventional zero-crossing on input 1, (f) conventional zero-crossing on input 2, (g) gradient on input 1, (h) gradient on input 2.* 116

4.43 *Checkers; fusion with second-order stabilizer: (a) input 1,* $\sigma_n = 10$*, (b) input 2,* $\sigma_n = 20$*, (c) gradient on ideal image, (d) gradient edge map, (e) conventional zero-crossing on input 1, (f) conventional zero-crossing on input 2, (g) gradient on input 1, (h) gradient on input 2.* 117

4.44 *Synthetic objects; fusion with second-order stabilizer: (a) range image, (b) intensity image, (c) fused surface, (d) gradient edge map, (e) conventional zero-crossing on input 1, (f) conventional zero-crossing on input 2, (g) gradient on input 1, (h) gradient on input 2.* 119

4.45 *Real intensity-range data; fusion with second-order stabilizer: (a) range image, (b) intensity image, (c) fused surface, (d) gradient edge map, (e) conventional zero-crossing on input 1, (f) conventional zero-crossing on input 2, (g) gradient on input 1, (h) gradient on input 2.* 120

FIGURE **PAGE**

4.46 *Real infrared–intensity data; fusion with second–order stabilizer: (a) infrared image, (b) intensity image, (c) zero–crossing map, (d) gradient edge map, (e) conventional zero–crossing on input 1, (f) conventional zero–crossing on input 2, (g) gradient on input 1, (h) gradient on input 2.* 121

4.47 *Fusion with first–order stabilizer and weighted inputs: (a) infrared image* $w_1 = 2$, *(b) intensity image,* $w_2 = 1$ *(c) zero–crossing map, (d) gradient edge map, (e) conventional zero–crossing on input 1, (f) conventional zero–crossing on input 2, (g) gradient on input 1, (h) gradient on input 2.* 123

4.48 *Surface reconstruction with membrane stabilizer: (a) input data,* $\sigma_n = 10$, *(b) surface recovered with global regularization parameter, (c) surface recovered with space-variant regularization parameters, (d) sparse input data, (e) surface recovered, (f) random input data with 30% density, (g) recovered surface.* 125

4.49 *Surface reconstruction with thin plate stabilizer: (a) input data,* $\sigma_n = 10$, *(b) recovered surface with global regularization parameter, (c) recovered surface with space-variant regularization parameters, (d) sparse input data, (e) surface recovered, (f) random input data with 30% density, (g) recovered surface.* 127

FIGURE **PAGE**

4.50 *Square; surface reconstruction: (a) input 1,* $\sigma_n = 5$*, (b) input 2,* $\sigma_n = 10$*, (c) ideal surface, (d) recovered surface, first–order stabilizer, (e) recovered surface, second–order stabilizer.* 128

4.51 *Cake; surface reconstruction: (a) input 1,* $\sigma_n = 10$*, (b) input 2,* $\sigma_n = 15$*, (c) ideal surface, (d) recovered surface, first–order stabilizer, (e) recovered surface, second–order stabilizer.* 130

4.52 *Checkers; surface reconstruction: (a) input 1,* $\sigma_n = 10$*, (b) input 2,* $\sigma_n = 20$*, (c) ideal surface, (d) recovered surface, first–order stabilizer, (e) recovered surface, second–order stabilizer.* 132

4.53 *Surface reconstruction of cup: (a) input 1,* $\sigma_n = 10$*, (b) input 2,* $\sigma_n = 20$*, (c) ideal surface, (d) recovered surface, first–order stabilizer, (e) recovered surface, second–order stabilizer.* 133

4.54 *Synthetic image of isoluminant color bands: (a) input color image, (b) color edges detected by our fusion technique.* 134

4.55 *Synthetic arm; color edge detection: (a) input color image (b) red channel, (c) green channel, (d) blue channel, (e) red edge map, (f) green edge map, (g) blue edge map, (h) fused edges.* 136

4.56 *Lena; color edge detection: (a) input color image, (b) red channel, (c) green channel, (d) blue channel, (e) red edge map, (f) green edge map, (g) blue edge map, (h) fused edges.* 138

FIGURE **PAGE**

4.57 *Peppers; color edge detection: (a) input color image, (b) red channel, (c) green channel, (d) blue channel, (e) red edge map, (f) green edge map, (g) blue edge map, (h) fused edges.* 139

4.58 *Parrot; color edge detection: (a) input color image, (b) red channel, (c) green channel, (d) blue channel, (e) red edge map, (f) green edge map, (g) blue edge map, (h) fused edges.* 141

4.59 *Balloon; color edge detection: (a) input color image, (b) red channel, (c) green channel, (d) blue channel, (e) red edge map, (f) green edge map, (g) blue edge map, (h) fused edges.* 143

4.60 *Robotics manipulator; color edge detection: (a) input color image, (b) red channel, (c) green channel, (d) blue channel, (e) red edge map, (f) green edge map, (g) blue edge map, (h) fused edges.* . . . 144

4.61 *Roof edge detection: (a) synthetic range input image, (b) vertical zero–crossings, (c) transposed of input image, (d) vertical zero–crossings of transposed image, (e) horizontal zero–crossings, (f) combined vertical and horizontal edges.* 146

4.62 *First–order responses: (a)* $\lambda = 1$*, (b)* $\lambda = 10$. 147

4.63 *Second–order responses: (a)* $\lambda = 1$*, (b)* $\lambda = 10$. 148

4.64 *First– and second–order responses: (a)* $\lambda = 1$*, (b)* $\lambda = 10$. 150

4.65 *Scale–space representation for the first–order stabilizer.* 151

4.66 *Scale–space representation for the second–order stabilizer.* 152

4.67 *First order stabilizer results under increasing noise levels: (a)* $\sigma_n =$ 10, *(b)* $\sigma_n = 15$, *(c)* $\sigma_n = 20$, *(d)* , $\sigma_n = 25$ *(e)* $\sigma_n = 30$, *(f)* $\sigma_n = 35$. 154

4.68 *Second order stabilizer results under increasing noise levels: (a)* $\sigma_n = 10$, *(b)* $\sigma_n = 15$, *(c)* $\sigma_n = 20$, *(d)* , $\sigma_n = 25$ *(e)* $\sigma_n = 30$, *(f)* $\sigma_n = 35$. 155

CHAPTER 1

INTRODUCTION

There are many systems that require acquisition, processing, and integration of information generated by multiple knowledge sources. There are also well–known biological examples of the synergistic integration of multisensor information [30]. Two such cases are (1) the fusion of infrared and visual information in the optic tectum of the rattlesnake, and (2) the fusion of acoustic and visual information in the barn owl. Another well–studied biological system is the human organism, which can integrate information from acoustic signals, pressure sensors, heat sensors, visual stimuli, and chemical inputs in real time and provides an existence proof of the potential capabilities achievable for technological applications. Thomopoulos [48] provided the following definition of the term "data fusion":

> The process of integrating raw and processed data into some form of meaningful inference that can be used intelligently to improve the performance of a system, measured in any convenient and quantifiable way, beyond the level that any of the components of the system separately or any subset of the system components partially combined could achieve.

More specifically, we could cite [2]

> At the core of every sensor fusion scheme is an inference mechanism that transforms the assessment of two (or more) different knowledge sources or sensors often given in the form of probability density functions or fuzzy membership functions into a unique assessment that reflects not only information generated by each sensor, but also knowledge which cannot be inferred by either sensor alone.

Applications that could benefit from more efficient and reliable mechanisms for information assimilation and decision making include autonomous robots employing multiple sensors, automatic target recognition systems (ATR), biomedical decision and support systems, weather prediction systems, air traffic control systems, and military intelligence gathering.

Multiple data needs to be integrated/fused in order to achieve one (or more) of the following goals:

(1) To deal with conflicting, imprecise, or incomplete information needed to infer data that could not be obtained otherwise, *e.g.*, depth from stereo, shape from motion, etc. The application or user of the fused data has to deal with real sensory data that includes not only the event of interest but also noise (both sensor noise and signal noise). Despite all these problems, successful applications are able to extract the necessary information. For example, ATR systems must distinguish typical friends from foes, as well as foes that may not be easily identificable. Another example is provided by medical expert systems, where, given a variety of symptoms, a single diagnostic is expected.

(2) To improve the signal–to–noise ratio by performing multiple frame averaging, or by employing a second kind of sensor when the first one is too noisy. Averaging can be employed for reducing the uncertainty ellipsoid to $\sigma/\sqrt{n}$. More sophisticated (robust) techniques are required, however, if the data contains outliers. On the other hand, a reconfigurable fusion system could discard the data derived from a given sensor if it is deemed too unreliable.

(3) To extend the spatial coverage, because one sensor can look where another cannot. For example, the wide area coverage of an ultrasonic sensor could guide a laser ranging system in a mobile robot. Another example involves distributed sensors, where only a subset of the sensors may be active at a time, but that may be enough to alert the decision center. (Consider the case of an array of radar systems along the border of a country.)

(4) To extend the "effective" dynamic range of operation of a system. For example, using infrared and visual signals would cover a greater portion of the electromagnetic spectrum, thereby increasing the effectiveness of an ATR system,

(5) To extend temporal coverage, because one sensor may be able to capture at a time when other sensors cannot. This may happen because of the nonzero time delay of any real sensor or because some sensors may be temporarily unable to respond due to jamming or other countermeasures.

(6) To compress information with minimal loss, and to facilitate data storage, transmission, and processing. Recent efforts in this area include wavelet–based sensor fusion and compression.

(7) To improve the system's operational reliability. This is made possible due to redundancy. Naturally, there is a trade–off between redundancy and cost.

(8) To increase confidence, where multiple independent measurements are made on the same event. Again, confidence (a relative measure of uncertainty) can be increased if multiple independent measurements are made on the same event or target. Early attempts combined information from multiple sensors into a composite model by averaging. Recent treatments attempt to find a solution that is optimal in some sense. An example is provided by the Gauss–Markov estimation theory.

(9) To reduce vulnerability to countermeasures, weather, and noise, and to improve the detection performance. Different sensing modalities and multiple sensors of the same modality increase survivability (as clearly seen in biological organisms).

In addition to these advantages, according to [31], the information from multiple sensors may be more timely and less costly to acquire when compared to the use of individual sensors.

Data fusion is a very active area of research and results in this field could have general applicability. This dissertation reviews several techniques for data fusion and introduces a new analytical data fusion technique.

1.1 Problem Statement

The objective of this dissertation is to develop an improved technique for feature extraction (edges) by using multisensory images. Data from several knowledge sources is fused (combined) to reduce uncertainty and to enhance detection. It is well known that edge detection is an ill–posed problem in the sense of Hadamard [23]; therefore, we will develop a regularized-based approach to restore well–posedness and to assure stability in the solutions.

We propose a series of functionals of increasing complexity that can be applied to the problems of gray and color edge detection as well as to the problem of surface reconstruction of both smooth and nonsmooth surfaces.

We will develop first the standard regularization approach that involves a single regularization parameter and a single stabilizer. We will generalize the standard regularization method to allow for space variant regularization parameters. In particular, we will employ $\lambda = \lambda(x)$ and $\lambda = \lambda(x, y)$ for the one– and two–dimensional cases, respectively. The need for this generalization arises from the fact that the standard formulation oversmoothes discontinuities.

We will extend the model once more to cover the case of two knowledge sources, and experiments will be performed using this model to fuse both one–dimensional and two–dimensional signals. Finally, we will generalize the model to the multi–source case, and then apply this later model to the problem of color edge detection.

1.2 Overview of the Dissertation

Chapter 2 contains a comprehensive survey of the literature. First, a review of the methods for inferring knowledge situates the present work in the field of data fusion. Second, a review of multisensor fusion applications provides a practical balance and sheds light on current implementations. Third, roof edge detection techniques are reviewed due to their importance to range images and because our database includes image pairs containing television and range images. Fourth, the fundamental papers on surface reconstruction are reviewed. Fifth, basic color concepts and the use of color for edge detection are considered. Chapter 3 provides analytical formulation for the problem of regularized fusion. Both one–dimensional and two–dimensional expressions are included here. Chapter 4 provides results when applying the proposed techniques to one–dimensional and two–dimensional data. Chapter 5 discusses the results and suggests directions for further research and experiments.

CHAPTER 2

BACKGROUND

This chapter provides background material for the major topics to be addressed in this dissertation. Section 2.1 presents the four major categories of methods for inferring knowledge and provides a context in which to situate our work. Section 2.2 reviews the literature of multisensor fusion with emphasis on application. Section 2.4 gives concepts about color and color edge detection. Section 2.3 provides an in–depth review of the surface reconstruction problem. Finally, Section 2.5 considers the problem of roof edge detection and presents recent formulations for detecting this kind of edge.

2.1 Methods for Inferring Knowledge

There are four basic categories of methods for inferring knowledge from two or more data sources [3]. These methods are based on SuperBayesian theory, belief theory, fuzzy sets, and analytic and geometric techniques.

Methods Based on SuperBayesian Theory

In techniques are based on Bayes' theorem, the state of the environment is decided from sensor measurements, knowledge about the types of states expected,

and sensor uncertainty. For example, under the assumption of Gaussian density functions for the sensor data, the best estimate for two sensors can be shown to be [22]

$$\tilde{\theta} = \frac{\sigma_2^2 x_1 + \sigma_1^2 x_2}{\sigma_1^2 + \sigma_2^2},$$

where σ_i represents the standard deviation associated with signal x_i. A great amount of research has been done employing these techniques, including Bayesian estimation of consensus sensors [32], MultiBayesian with ϵ–contaminated Gaussian distributions [15], and Bayes–maximum entropy multisensor data fusion [6].

Methods based on Belief Theory

These techniques use Dempster's rule of evidence combination. The Dempster-Shafer theory is based on probability theory, yet allows probability judgements to capture the imprecise nature of the evidence. Degrees of likelihood are measured by probability intervals, as opposed to point probabilities as in the Bayesian approach [54]. An important advantage of this theory is its ability to express degrees of ignorance. For the particular case that both m_1 and m_2 (probability masses corresponding to two independent sources of evidence) support θ, Dempster's rule becomes

$$m_c(\theta) = 1 - (1 - m_1)(1 - m_2).$$

This theory often produces an overestimation of the final assessment.

Methods Based on Fuzzy Sets

The fuzzy set theory focuses on the issue of representing and managing vague or inexact information. In conventional (or "crisp") set theory, elements either belong or do not belong to a set (hard decision). Fuzzy set theory generalizes the notion of membership from binary to a continuum of grades of membership. The set theoretic operations of *union*, *intersection*, and *complement* are defined by their membership functions. For instance, the membership function $\mu_{A \cup B}$ of the union $A \cup B$ is pointwise defined for all $u \in U$ by $\mu_{A \cup B} = \max\{\mu_A(u), \mu_B(u)\}$. A fuzzy fusion function recently proposed [1] was expressed by

$$\mu = e + [\frac{1}{n} \sum_{i=1}^{n} (\mu_i - e)^{\alpha}]^{1/\alpha},$$

where $u_i(\theta)$ are the truth values committed to θ, e is the identity of the truth region (usually 0.5), and α is a parameter used to control the relative weight of the evidence (e.g., for $\alpha = 1$, the fusion formula coincides with the arithmetic mean). Abdulghafour [1] provided a list of performance criteria and a comparison of the formula above with respect to SuperBayesian theory and Dempster's rule for certain image processing applications.

Methods based on Analytic and Geometric Techniques

These techniques infer knowledge from two assessments by using some constraints on the evidence collected to infer the final judgement. An example of such a technique is presented below [3].

Let μ and ν be two normalized fuzzy membership functions, and let φ be the Taylor series expansion

$$\varphi(\mathbf{x}) = \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} c_{ij} \mu^i \nu^j. \tag{2.1}$$

Here the c_{ij}'s are unknown coefficients that must be determined. Since the $c_{ij}\mu^i\nu^j$ decreases in magnitude as $(i + j)$ increases, higher–order terms can be ignored. The first–order approximation of the series becomes

$$\varphi(\mathbf{x}) = c_{00} + c_{10}\mu + c_{01}\nu. \tag{2.2}$$

Because the constant coefficient does not depend on any of the inputs, it can be ignored. Normalization of the output results in the auxiliary equation $\alpha + \beta = 1$, where $\alpha = c_{10}/(c_{10} + c_{01})$ and $\beta = c_{01}/(c_{10} + c_{01})$ are the normalized coefficients. Two fusion constraints (or principles) were enforced. First, the knowledge source corroboration (KSC) principle, which allows for a feature not detected by one sensor but detected by another to be included in the output. Second, the belief enhancement/withdrawal (BEW) principle, which causes a feature detected by one sensor and not detected by another to be eliminated from the output. These two principles were enforced by minimizing the functional

$$\int \cdots \int (1 - \varphi)^2 d\mathbf{x} + \lambda^2 \int \cdots \int \|\nabla(\alpha\mu - \beta\nu)\|^2 d\mathbf{x}. \tag{2.3}$$

After some algebraic manipulation, Abidi [3] obtained

$$\nabla^2\alpha + \sum_{i=1}^{n} A_i \alpha_{x_i} + B\alpha = C. \tag{2.4}$$

After solving (numerically) for α, the final fusion function is computed as

$$\varphi(\mathbf{x}) = \alpha(\mathbf{x})\mu(\mathbf{x}) + (1 - \alpha(\mathbf{x}))\nu(\mathbf{x}). \tag{2.5}$$

The new fusion technique, which we are to develop and describe in the next chapter, can be classified into this category, as an analytic fusion technique.

2.2 MultiSensor Fusion: Applications

This section presents a review of important papers related to the problem of fusing specific sensor data. It is not mere coincidence that most of the material concerns TV and range data, since these are the two most common sensors available in vision and robotic research. The papers, organized chronologically, were chosen either because they provide details of a complete system ([41], [50]), or because they provide a complementary view of the fusion process, or both.

Gil *et al.* [18] proposed combining intensity and range data sources for segmentation. Since both sources are dissimilar, they proposed using edge maps. This should place both sources of information in the same form. Intensity edges were computed using Kirsh operators. Noisy spurious edges were suppressed by semi-thresholding and nonmaxima suppression. A range edge map was derived by applying nonmaxima suppression on a modified angle of curvature picture. Two procedures were proposed for extracting common edges. In the first method, local AND, each edge point in the intensity edge map located in the neighborhood of the corresponding pixel in the range map, will be marked as an edge point. The

second procedure, global AND, uses an entire intensity edge to guide the search in the range edge map. This approach provides two advantages. First, since only intensity edges of lengths greater than a given threshold were considered, many spurious edges were eliminated. Second, small gaps in range edges may be filled using the corresponding intensity edge.

Tong *et al.* [50] proposed a multisensor data fusion system of laser radar and forward–looking infrared (FLIR) for target segmentation. They made use of several characteristics of range data to aid in segmentation. The first characteristic is range data's relative insensitivity to spectral characteristics in a scene as well as to seasonal or diurnal variations. Second, natural objects, like trees and shrubs, exhibit large range jumps due to the spacing between leaves and branches. Natural backgrounds exhibit a similar phenomena. On the other hand, possible targets, such as vehicles and buildings, present smooth surfaces and slow varying ranges; therefore, small gradients point to possible targets. To the gradient magnitude, a logarithmic transformation was applied to improve detectability. (They assumed that the small number of target pixels exhibit small gradients and the large number of background pixels exhibit large gradients). A logarithmic transformation was also applied to the magnitude of gradient of the FLIR image. The combination of the (enhanced) LADAR and FLIR data through multiple masks presented information which was not visible to either of the sensors by itself.

Chou and Brown [11] developed a technique for incorporating intensity discontinuity observations in order to reconstruct and segment a sparse depth map.

They proposed a coupled Markov random field (MRF) model that incorporates depth and intensity observations. The following energy functional was minimized

$$V(f,l|g,O) = \sum_c V_c(f,l) + (\sum_s (f_s - g_s)^2/\sigma_s - \sum_d \log \lambda_d(l_d)),$$

where V_c is the potential function of clique c, λ_d is the likelihood ratio of site d given the intensity observation O_d, f represents the depth sites, l represents the discontinuity sites, and g represents the depth measurements. The authors employed a deterministic gradient descent technique called "highest confidence first" (HCF) to minimize the functional. The algorithm was tested on noisy synthetic depth and intensity images.

Krotkov and Kories [28] explored how two particular visual ranging processes could work together to compute the 3–D position of objects in an unknown environment. They estimated the distance and uncertainty for both focus ranging and stereo ranging. The stereo ranging procedure matched extracted line segments and calculated disparities, and then computed the range to the midpoint of matched line segments. Next, a verification procedure was used to try to confirm a match by focusing on the matched feature. A set of verification rules was implemented that either rejected or accepted the match. If accepted, the image point location and the ranges were stored. A similar cross–checking procedure was implemented for focus ranging (verified with stereo). Reliability was defined by how well the cooperative processes rejected inconsistent measurements. Suggested improvements include increasing both the quantity and density of the range points.

Shaw *et al.* [42] proposed a technique to integrate the information available from TV images and low–resolution radar scattering cross sections (RCS) to reconstruct individual objects' 3–D shapes. In their system, a partial surface model acquired from the image data was used to guide the conversion of RCS data into a more complete surface model. These two sensors complement each other in several aspects; for example, matte surfaces are better suited for optical imaging; smooth metallic surfaces, on the other hand, create specular points and present difficulties for optical processing. The same surfaces, however, are the easiest to model in RCS. From the optical image they extracted occluding contours and a partial surface shape description for the scattering object. A decision module attempted to match the shape in the optical image with some simple geometrical shapes. The proposed technique was applied in order to recover the edge of a flat plate in a simulated environment. Because electromagnetic modeling techniques are only feasible and reliable for simple scattering geometries, reconstruction of complex 3–D shapes is not currently practical.

Poggio *et al.* [37] proposed an MRF–based approach for integrating information between different visual cues. Several early vision processes, such as edge detection (brightness edges), stereopsis, motion, texture, and color, give separate cues about object surfaces. The first step is to combine the brightness edges with discontinuities in each of the modules separately. The central point in their integration scheme assumed that changes of brightness guide the computation of discontinuities in the physical properties of the surface. They conjectured that

changes in surface properties usually produce large brightness gradients in the image. Following Geman and Geman [17], they employed line and continuous processes with the prior energy term:

$$U_i(f,l) = \sum_j (f_i - f_j)^2(1 - l_i^j) + \beta V_c(l_i^j),$$

where l_i^j is a binary line element between sites i and j, and V_c reflects the likelihood of a given configuration of line processes.

Wright [53], following Geman and Geman [17], expressed the data term and prior cost function for a single channel as

$$U(I) = \sum_i \frac{(I_i - D_i)^2}{2\sigma^2} + \sum_i \sum_j V_2(i,j,l_{ij}) + \sum_k V_k(l). \tag{2.6}$$

He stated that "for multi-sensed data there will be no correlation between the absolute pixel values. This correlation, therefore, takes place over some hidden representation of the images." The proposed correlation will be the position of a region boundary. Wright introduced then an additional measure to the prior term which enhanced the coincidence of an edge value occurring at the same edge site in the different images. The coupling function implemented was

$$V_3(lrs) = \begin{cases} -C & , \text{ if } \sum_{k=1}^{N} l_{rsk} = 0 \text{ or } 3 \\ +C & , \text{ otherwise,} \end{cases} \tag{2.7}$$

where C is a coupling constant, l_{rsk} are ternary cliques, and $N = 3$ is the number of images. Wright presented results for two experiments. In the first experiment, an artificial RGB color image degraded with Gaussian noise was reconstructed with and without the coupling term ($C = 1$, and $C = 0$ respectively). In a second

experiment, he considered N images of the same type, each with different levels of noise added. Since the minimization procedure was based on simulated annealing, he warned that it would not be practical for images 256×256 or bigger.

Nasrabadi *et al.* [35] proposed a cooperative method in which image intensity and optical flow information were integrated into a single stereo technique by modeling the input data as coupled MRFs. Their goal was to develop a motion stereo–matching algorithm that would be minimally affected by occlusion. They assumed an independent module computed dense optical–flow estimates. The optical–flow information was used to indicate object boundaries (depth discontinuities) and occluded boundaries in order to improve the disparity solution in occluded regions. They proposed two algorithms: optical flow into intensity–based stereo matching and edge–based stereo matching. They proposed two binary fields to help in the reconstruction process. The first binary field indicated a potentially occluded region; the second one was employed to prevent smoothing over discontinuities in the disparity solution. Actual data and smoothing terms were

$$D_{ij} = (1 - \phi_{ij})c_A[I^l(i,j) - I^r(i + d_{ij}, j)]^2 + c_B[\rho^l(i,j) - \rho^r(i + d_{ij}, j)]^2 + c_\phi \phi_{ij}, \tag{2.8}$$

where ϕ_{ij} is a binary field indicating occlusion, I is the intensity image, d_{ij} is the disparity to be recovered, ρ is the magnitude of the optical flow, and c_A, c_B, and c_ϕ are constants.

$$S_{ij} = \sum_{k,l \, \epsilon \, N_{ij}} [d(i,j) - d(k,l)]^2 (1 - \psi_{k,l}) + c_\rho \psi_{k,l}, \tag{2.9}$$

where d is the disparity solution, ψ is a binary field indicating depth discontinuities, and c_ρ is a constant. Nasrabadi *et al.* presented results for synthetic data only. Stochastic annealing was employed for energy minimization.

Bilbro and Snyder [7] addressed the problem of fusing registered range and luminance data. Given noisy luminance and range images l_o and z_o, they obtained restored images l and z such that l resembles the effect of light reflecting off z. They proposed the following objective function:

$$H = \sum_i (l_i - l_{oi})^2/\sigma_l^2 + \sum_i (z_i - z_{oi})^2/\sigma_z^2 + \sum_i (l_i - \rho(n_i.z_i))^2/\tau, \qquad (2.10)$$

where σ_l^2 and σ_z^2 are the noise variances of the luminance and range data, τ is a user selected parameter equivalent to temperature, ρ is the albedo (measures the fraction of the incident light reemitted by the surface), n_i is the surface normal, and z_i is the vector $(0, 0, 1)$. Mean field annealing was employed for minimizing the objective function. A term that penalizes variations from locally planar surfaces was also suggested. The proposed algorithm was tested with synthetic data.

Roggemann's [41] work concerned the problem of automatic detection of man–made military vehicles in natural backgrounds. The working hypothesis was that the performance of automatic target detection systems could be improved through the use of multiple sensors. Processing proceeded in two major stages: sensor–dependent processing and multiple sensor processing. FLIR image segmentation was based on two observations: targets in the database appeared brighter than the background, and targets occupied a minor fraction of the total pixels. Hence, a threshold was selected based on the premise that the target brightness levels

were contained in the "rough" region of the histogram. Range image segmentation was based on the observation that the surfaces of tactical targets are reasonably well modeled as collections of small, approximate planar patches of varying orientations. Planes were fit to 3×3 regions in the smoothed range image. A range–dependent error threshold was then applied to the error image. Heuristics were applied to the threshold image to reject more non–target pixels and to recover some of the target pixels. Two single sensor detection algorithms (FLIR–only and range–only) and three multiple sensor detection algorithms (FLIR assisted by range, range assisted by FLIR, and a single decision algorithm) were implemented. The multiple sensor approaches were found to provide improved performance in all performance measures.

Mathur *et al.* [33] developed a CMOS chip to perform pixel level data fusion in real time. To overcome the problem of oversmoothing, they proposed minimizing the 1–D functional:

$$E(u,l) = \sum_i g_i(u_i - d_i)^2 + \gamma \sum_i (1 - l_i)(u_{i+1} - u_i)^2 + \alpha \sum_i l_i,$$

where u is the unknown function, d is the measured data, and l is a binary line process. An example was provided in which the two sensors provide complementary information. For example, laser radar (LADAR) provided range data and IR focal plane array (IR FPA) provided temperature data. The algorithm consisted of four steps: (1) boundaries in the laser data were identified, (2) these boundaries were used as line processes for IR FPA data, (3) data smoothing was performed on the FPA data for noise reduction and spreading of information, and (4) reading

out fused data. In their chip the line processes were implemented by an array of switches. The depth discontinuities detected in LADAR data controlled the state of the switches while a grid of resistors implemented the data smoothing operation. The chip was tested with (noisy) synthetic data.

Beckerman [6] employed a Bayes-maximum entropy formalism to fuse ultrasound and visual sensor data acquired by a mobile robot. The main objective was to treat systematic errors (as opposed to the removal of noise or other random errors). The positions of the vertical edges in the visual image were extracted using a Canny–like edge detector. The locations in the ultrasound viewing plane of the angular positions of the depth discontinuities were also extracted. He used a box tall enough to intercept the ultrasound viewing plane when placed on the floor. Maximum entropy maps were used to represent the occupied and empty regions of space. Bayes' theorem was employed to test the hypothesis that a pixel under consideration was consistent with the location of an ultrasound boundary and a corresponding visual depth discontinuity. A dual visual–ultrasound representation was devised, which included visual vertical edges as well as horizontal ultrasound strings (after mapping both into the same space).

2.3 Surface Reconstruction Techniques

This section begins with a review of the seminal paper by Grimson, and it follows with a review of various papers in chronological order. The emphasis has

been put on variational formulations.

Grimson [21] developed a computational theory of visual surface interpolation and proposed an algorithmic implementation. He wanted to account for the ability of the human visual system to perceive complete surfaces when the input (3–D information) is given only at particular points of the image: the zero–crossings. He postulated the *surface consistency constraint* (or "no news is good news") that basically stated that between known depth values the surface cannot change in a radical manner because such changes would give rise to additional zero–crossings. From the fact that any appropriate functional should measure the amount of variation in the local surface orientation (i.e., it should measure some factor of the second order derivatives of the surface), he obtained various possible functionals. Under small deflection assumptions, he proved that the square Laplacian and the quadratic variation guarantee a solution (up to an element of the null space). The surface reconstruction problem states that given a set of scattered depth constraints corresponding to points along the zero–crossings contours of the primal sketch, the surface that is most consistent with the image irradiances is that which minimizes the potential energy functional:

$$E(f) = \int\int (f_{xx}^2 + 2f_{xy}^2 + f_{yy}^2)\, dx\, dy + \frac{\beta}{2}\sum_i [f(x_i, y_i) - c(x_i, y_i)]^2. \qquad (2.11)$$

Grimson proposed two algorithms for minimizing the above functional:

- the gradient projection interpolation algorithm (when $f(x_i, y_i) = c(x_i, y_i)$)
- the conjugate gradient approximation algorithm

He tested his theory on both synthetic and real data (disparity values obtained using the Marr-Poggio stereo theory). Results seem to confirm his prediction that the quadratic variation functional produces surfaces that better agree with a human observer. The algorithms assume that the surface to be interpolated is twice continuously differentiable. This will not be a valid assumption along occluding contours between two surfaces, separated in depth in the scene. While it is straightforward to adjust the algorithms to interpolate only within certain boundaries, the question of how best to detect the discontinuities is still open.

Koch *et al.* [27] addressed the use of analog networks to implement and solve problems in early vision, such as reconstructing and smoothing images from sparsely sampled data. Surface reconstruction was formulated in terms of minimizing a quadratic energy functional

$$E(f) = \|Bf - d\| + \alpha\|Sf\|, \tag{2.12}$$

where f is to be computed, B is a diagonal matrix, and the stabilizer S corresponds to the operator associated with a membrane or thin plate (α is a regularization parameter). E can be reformulated as

$$L(V) = \frac{1}{2}\sum_{ij} T_{ij}V_iV_j + \sum_i V_iI_i, \tag{2.13}$$

where V_i corresponds to the voltage at the processing element i, and I_i corresponds to a current injected into neuron i and T_{ij} the conductance between neuron i and neuron j. Quadratic variational problems can then be solved by an appropriate electrical network. The authors generalized the Hopfield–Tank network to solve

nonconvex energy functionals. (It is known that the line processes, which were introduced to control the oversmoothing across discontinuities, make the problem nonquadratic). Following Hopfield they mapped the binary line processes h_{ij} and v_{ij} into continuous variables bounded by 0 and 1. They proposed a cost function of four terms: $E = E_I + E_D + E_L + E_G$, where E_I implements a membrane type of surface interpolation; E_D implements the data term; E_L implements the line potential network; and E_G implements the gain (transfer function) term. They simulated an analog network for a 32×32 image in a digital computer. Applied on synthetic data, the piecewise smooth–surface reconstruction technique performs better than the smooth–surface reconstruction method. The single most important advantage of analog networks is their speed (convergence times are on the order of several time constants). The principal drawback of the method is that there is no guarantee that the network will converge to the state of lowest energy.

Blake and Zisserman [8] define *visual reconstruction* as the process of reducing visual data to stable descriptions. They suggest that continuity is a necessary simplifying assumption about the world. Global continuity can be unreasonable when there are sudden changes across the scene. In that case, they suggest employing *weak continuity constraints* — constraints that can be broken occasionally if that makes for a simple overall description. The mechanism for actually imposing the constraints is the graduated non-convexity (GNC) algorithm. To illustrate their method, let's consider the problem of detecting step discontinuities in 1–D. The

goal is to construct a piecewise smooth function $u(x)$ that is a good fit to the data $d(x)$. This can be achieved by modeling $u(x)$ as a weak elastic string (WES). The weak elastic string is specified by its associated energy E, so finding $u(x)$ is then the problem of minimizing the expression $E = D+S+P$, where $D = \int (u-d)^2 dx$: a measure of faithfulness to the data; $S = \lambda^2 \int u'^2 dx$: a measure of smoothness; P: sum of penalties (α) imposed for each break in the string. In discrete form, the problem is then

$$\min_{\{u_i,l_i\}} E = \sum (u_i - d_i)^2 + \lambda^2 \sum (u_i - u_{i-1})^2 (1 - l_i) + \alpha \sum l_i, \tag{2.14}$$

where l_i (line process) are boolean variables (1 indicates the WES is broken between nodes $i-1$ and i, therefore, disabling the smoothness constraint). By minimizing over the variables $\{l_i\}$ first, the problem is converted to

$$\min_{\{u_i\}} F = D + \sum g(u_i - u_{i-1}), \tag{2.15}$$

where g, the neighbor interaction function, has a U-shaped form with $g_{max} = \alpha$. Since F is nonconvex, there may be many local minima. The GNC algorithm optimizes a whole sequence of cost functions (starting with a convex approximation to F) using the result of one optimization step as the starting point for the next. The authors have implemented their algorithm in both sequential and parallel machines. Tests indicate that it compares favorably to simulated annealing minimization approaches. Potential analog VLSI implementation is suggested, using a mixture of resistors and nonlinear elements.

According to Harris [24], surface reconstruction seeks to find a plausible surface

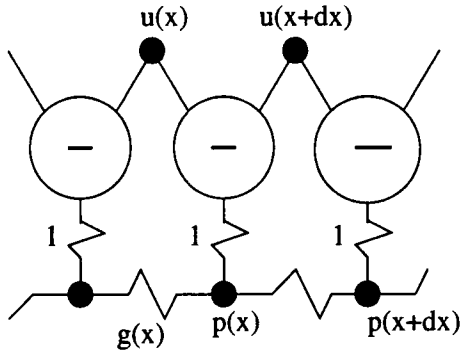


Figure 2.1: *1–D smoothing network.*

given sparse, noisy, and possibly conflicting measurements of an original surface. To accomplish this goal, Harris proposed a network consisting of depth nodes connected to slope nodes by ideal subtract constraint boxes. (See Fig. 2.1).

The total power dissipated by this network is given by

$$E = \int (\frac{du}{dx} - p)^2 + g(\frac{dp}{dx})^2 dx.$$

Applying Euler–Lagrange equations to find the stationary points of the energy functional and rearranging terms, it is shown that the solution u satisfies the 1–D form of the biharmonic equation, which means that the network smoothes the depth values to biharmonic smoothness. The generalized coupled depth/slope network is capable of implementing the controlled-continuity stabilizers of Terzopoulos (depth and orientation discontinuities are achieved by "breaking" the resistors at appropriate places). Two–dimensional extensions require two resistor meshes to smooth the p and q slope values. Harris indicated that the proposed algorithm was appropriated on a computer such as the Connection Machine. The

algorithm was tested with synthetic data. The computational model has a number of advantages: if only depths are given, the model reconstructs a surface with the same biharmonic smoothness used by Terzopoulos and Grimson, and discontinuities (depth and orientation) can be used to compute the reconstructed surface.

Terzopoulos [47] developed a computational theory of visible–surface representations. The visible surface reconstruction process proposed unifies four computational goals: (1) integration (of depth constraints and orientation constraints), (2) interpolation (propagation of shape information into regions lacking shape constraints), (3) detection of discontinuities (to limit the interpolation process), and (4) efficiency (via multiresolution relaxation). Terzopoulos recognized that surface reconstruction is an ill-posed inverse problem and proposed using the regularization method for its solution. He proposed a variational approach (or principle): VP1: find u that minimizes $E(v) = S(v) + P(v)$, where

$$S(v) = \frac{1}{2}\int\int \rho(x,y)\{\tau(x,y)(v_{xx}^2+2v_{xy}^2+v_{yy}^2)+[1-\tau(x,y)](v_x^2+v_y^2)\}\,dx\,dy \quad (2.16)$$

is a controlled–continuity stabilizer and $P(v)$ is a penalty functional that measures discrepancy to depth and orientation constraints. Since a closed–form solution is not feasible (due to the irregular occurrence of constraints and discontinuities), he discretized the continuous problem using finite elements (square and triangular). He then developed a set of computational molecules to represent the different terms of the nodal equations. Terzopoulos argued that multilevel relaxation methods provide the necessary speed–up required for solving various early

vision problems. The proposed surface reconstruction algorithm was tested on a variety of data sets including synthetic data, laser rangefinder data, photometric data, and digital terrain model data. Further research directions include tuning the smoothness of the reconstructed surface according to the noise present in the data; analog computing by electrical networks, and applying concepts from differential geometry to generate stable higher-level shape descriptions which are appropriate for object recognition.

Poggio and Girosi [38] considered learning an input/output mapping from a set of examples as a problem of hypersurface reconstruction. Since the problem of learning a smooth mapping from examples is ill–posed (the information in the data is not sufficient to reconstruct the mapping uniquely in regions where data are not available), they developed a theoretical framework based on regularization techniques. The inverse problem of finding a (hyper)surface $f(x)$, given its value y_i on a finite set of points x_i, is solved in regularization theory by minimizing a cost functional consisting of two terms:

$$H[f] = \sum_{i=1}^{N}(y_i - f(x_i))^2 + \lambda\|Pf\|^2. \tag{2.17}$$

They proved that the solution to the regularization problem is given by

$$f(x) = \sum_{i=1}^{N} c_i G(x; x_i), \tag{2.18}$$

where $G(x; x_i)$ is the Green's function centered at the point x_i, and the c_i are coefficients to be found by solving a set of linear equations. In particular, if the operator P is rotationally and translationally invariant, G will be a radial function:

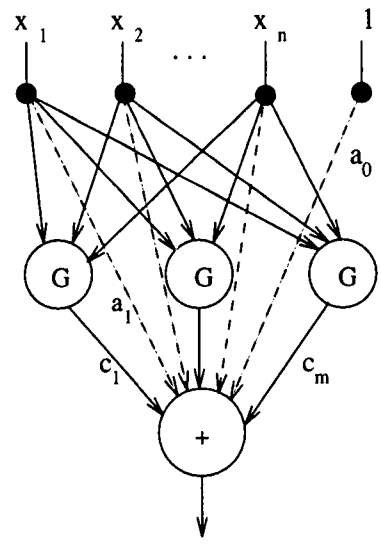


Figure 2.2: *Regularization network.*

$G = G(\|x-y\|)$. The proposed technique can be implemented by a simple network with one hidden layer as shown in Fig. 2.2. There is one hidden unit for each data point x_i. Depending on the stabilizer, constant (bias), linear, and higher order polynomials may be needed (shown here as direct connections from the input to the output). The most general scheme was noted as the hyper basis functions (HyperBF) scheme because it is related to the radial basis functions. A gradient–descent approach to learn center's positions and norm weights was also included. Three applications of this scheme were sketched: (1) recognizing a 3–D object from its perspective views, (2) learning dynamical systems, and (3) learning perceptual and motor tasks.

In their report, Stevenson and Delp [44] addressed two problems: the development of an algorithm which is essentially invariant to the choice of coordinate

system, and the development of a computationally efficient algorithm which will reconstruct curves and surfaces with discontinuities. They cast the problem as an inverse one that must be stabilized. In abstract form, the process of constraint formation was modeled by $c_i(u) = l_i[r(u)] + e_i(u)$, where $r(u)$ is the parametric representation of the curve or surface, l_i is the measurement functional, and c_i is the noise–corrupted shape estimate. The object reconstruction problem is to form an estimate $\hat{r}(u)$ of the object $r(u)$ for all $u \in U$, given a finite collection of noisy and sparse shape constraints. Using Tikhonov regularization, they proposed minimizing the functional (for curve reconstruction):

$$M^{\lambda}[r^*(u), S] = \int_U \kappa^2(u)\nu(u)du + \lambda \sum_{c_i} \|r^*(u_i) - c_i\|^2, \tag{2.19}$$

where $\nu(u) = \|r'(u)\|$ is the velocity of the curve, $\kappa(u) = \frac{\|t'(u)\|}{\nu(u)}$ is the curvature of the curve, and $t(u)$ is the tangent vector to the curve. Providing for coordinate system invariance caused nonconvexity, however. To form a convex approximation, a two–step algorithm was proposed. The authors listed several open research problems, including the robustness of the algorithms when there are outliers in the data, and computation of the parameters λ and T from the data.

Suter [45] derived his constraint networks from augmented Lagrangian approaches to constrained minimization:

$$L_A = J(\psi) + \lambda G(\psi) + \rho(G(\psi))^2,$$

where $G(\psi) = 0$ is a single constraint, $\rho \geq 0$ is fixed, and λ is a Lagrange multiplier constant. This approach generalizes the standard multiplier approach

and the standard penalty approach. Suter's method generalizes Harris' coupled depth/slope formulation in two ways. First, the augmented Lagrangian approach specializes to Harris' penalty–based approach, or specializes to an alternative Lagrangian formulation. Second, his technique permits variations in the manner in which introduced functions (derivative approximations) are expressed and related by constraints. In this paper, Suter focused on analog network solutions to the objective functionals. He presented two examples, one– and two–dimensional reconstruction, and proposed using the constraint network for data fusion. Suter's work also benefitted from the analog solutions proposed by Platt. Future research includes the analog VLSI realization of first–order nets and regularization and spline fitting by analog networks.

Chen [10] proposed a new surface reconstruction method based upon a robust back propagation (RBP) algorithm, which extended the BP algorithm to handle outliers in the training data. Instead of minimizing the sum of squared errors (least squares — LS principle) the new algorithm minimizes a new objective function whose form changes with time. The objective function $\phi_t(r)$ resembles Hampel's tanh estimator (inverted bell shaped). The contributions of data having large noise contamination are minimized by $\phi_t(r)$. A new learning algorithm is derived by minimizing this new objective function using gradient descent. The proposed algorithm requires selecting five parameters. A critical one is q, an upper bound on the percentage of outliers presented in the training data. Experimental results, for both curves and surfaces, indicate that the robust BP (RBP) method performs

satisfactorily even in the presence of outliers in the training data. In which case, conventional BP produced erroneous results. There are two known situations where the RBP algorithm may fail. The first is when the percentage of outliers exceeds the value of q. The second is when the initial estimator (of the mapping) fails to be near the underlying function.

Szeliski [46] presents a Bayesian model which captures uncertainty associated with low–level vision. The model, applicable to 2-D dense fields, such as optical flow, consists of three components:

- The prior model – captures any *a priori* information about the structure of the field.
- The sensor model – describes the behavior and noise characteristics of the measurement system.
- The posterior model – combines information from the prior and sensor models using Bayes' rule.

The specific Bayesian models developed are based on Markov random fields (MRF), which describe complex probability distributions over dense fields in terms of local interactions. His method computes best estimates and the level of uncertainty of those estimates using second–order statistics (variance and covariance). The author observed that the estimates obtained with regularization and other energy–based algorithms were equivalent to posterior estimates obtained with MRF modeling. The proposed method has been tested on incremental image–based

depth from motion, motion estimation without correspondence, and generation of fractal surfaces.

Muraki *et al.* [34] proposed an algorithm for reconstructing a topographic shape from a topographic map using contour lines and a sparse set of altitude data. They added a new constraint to the traditional regularization methods. This constraint enforces that points on the same contour line have the same height. The functional they minimize is

$$E = E_{smooth} + \lambda E_{data} + \mu E_{contour}. \tag{2.20}$$

Experimental results indicate improvements over Grimson's method when applied to color images. They reported processing times of 38 minutes on a Unix workstation. Future work may include dealing with additional constraints as well as massive parallelism.

2.4 Color Edge Detection Techniques

We shall illustrate the generalization capabilities of our methodology by applying it to the problem of color edge detection. This section begins with a review of the basic concepts about color. A literature review of color edge detection techniques follows.

Color Fundamentals

Light is the electromagnetic radiation that stimulates our visual response. The eye contains two different photoreceptors called *rods* and *cones*. The rods provide *scotopic* (night) vision, while the cones provide both color and *photopic* (day) vision. The *luminance* or intensity of a spatially distributed object with light distribution $I(x, y, \lambda)$ is defined as

$$f(x, y) = \int_0^\infty I(x, y, \lambda) V(\lambda)\, d\lambda,$$

where $V(\lambda)$ is called the relative luminous efficiency function of the visual system (a bell-shaped curve). The *brightness* of an object is its perceived luminance, which is dependent on the luminance of its surroundings.

The perceptual attributes of color are *brightness*, *hue*, and *saturation*. The hue of a color refers to its dominant spectral component (e.g., greenness). Saturation refers to the amount of white light added to a monochromatic light (e.g., intense red or pale red). Typically, the brightness, hue, and saturation color space is represented as a cone; where brightness varies along the vertical axis, the hue varies along the circumference, and saturation varies along the radial distance. Color representation is based on the classical theory of Young (1802), who stated that any color can be reproduced by mixing an appropriate set of three primary colors. It has been found that the human retina has three types of cones whose responses peak in the blue, green, and yellow–green regions of the visible electromagnetic spectrum. Based on this three–color theory, the color sensation that a colored

light will produce can be described by spectral responses given by

$$\alpha_i(C) = \int_{\lambda_{min}}^{\lambda_{max}} S_i(\lambda)C(\lambda)\,d\lambda, \quad i = 1,2,3,$$

where $S_i(\lambda)$ represents the absorption spectra of the receptors, and $C(\lambda)$ is the spectral energy distribution of the "colored" light.

There are a variety of color coordinate systems. One the most popular is the Commission Internationale de l'Eclairage (CIE) spectral primary system RGB established in 1931. This standard was first used in color television and eventually in computers. Other systems are the CMY (subtractive) system used in color printers, the NTSC YIQ transmission system, the CIE XYZ, the $U^*V^*W^*$ system, and the HSV (HSB) system. There are a couple of reasons for this variety of systems. First, no practical set of three primaries has been found that can reproduce all colors. Second, different color spaces possess different color metrics.

Color Edge Detection

Robinson [40] considered five different color coordinate systems for edge extraction. These included: RGB, NTSC YIQ, CIE $L^*a^*b^*$, principal components $P_1P_2P_3$, and $G_1^*G_2^*G_3^*$. The NTSC YIQ color transmission coordinates partially remove the correlation known to exist in the R, G, and B components of an image. The use of the Karhunen-Loeve transformation decorrelates the color components and allows edge detection to be performed on the individual principal components of the particular image. The CIE color coordinate system was suggested because

it possesses metric difference sensitivity to a close approximation. The components G_1^*, G_2^* and G_3^* were obtained using a model of the human visual system developed by Frei. Robinson proposed using the edge activity index, EAI, as an objective measure for comparing the color coordinate systems for suitability for edge extraction. She applied the set of Kirsh masks on all the components and concluded that if only one component were to be used for edge extraction, then the component that ought to be chosen should be G, P_1, Y, L^*, and G_1^* in the five color systems considered.

Nevatia [36] proposed a color step edge detector by extending the (achromatic) Hueckel edge operator to color edges. He chose the YT_1T_2 color coordinate system, defined as follows: $Y = c_1R+c_2G+c_3B$, $T_1 = R/(R+G+B)$, $T_2 = G/(R+G+B)$, where Y gives the luminance of the image pixel, and T_1 and T_2 jointly describe the chromatic information associated with a pixel. Furthermore, by considering the color triangle in the $T_1 - T_2$ space, the distance r from the white point (W) and any other point (P) represented a measure of *saturation*, whereas the angle θ between the line joining P to W and the horizontal (T_1-axis) measured the *hue*. Nevatia assumed the edges in the three color components were independent, except for the constraint of having the same orientation. Following Hueckel, he then minimized the expression $N^2 = N_1^2 + N_2^2 + N_3^2$, where $N_k^2 = \sum_{i=0}^{8}(a_{ik} - s_{ik})^2$, a_{ik} and s_{ik} being the ith terms of the expansion for the signal and the step for the kth component of the color image. Nevatia tested the method on two real images and compared the results for chromatic versus luminance edges. He also computed hue and

saturation edges. By analyzing the experimental results, the author concluded that (i) separating the edges into hue and saturation components did not improve the results, (ii) while a larger percentage of edges in the chromatic components corresponded to the desired edges, these edges tended to have more gaps, and (iii) a luminance edge was also present for a very large percentage of the cases where a chromatic edge was present. Finally, the author suggested that although the process generalized to an arbitrary number of multispectral components, the correct choice of coordinates for these spectral spaces needed to be determined because there existed no analogies in the human visual system.

Hunterberger and Descalzi [25] addressed the problem of color edge extraction through an image model representation based on fuzzy set theory. Individual pixels were represented by their fuzzy membership values to clusters in a color vector space. These fuzzy membership values were generated by an iterative segmentation scheme based on the fuzzy c–means algorithm. A color edge was defined as the zero–crossings of differences between membership values at each pixel. The proposed operator was applied to the natural (color) scene of a house. Color spaces chosen for the experiments were the RGB space, a color space derived from the results of the Karhunen–Loeve transformation, and the CIE XYZ color space. They also used a Riemannian metric for the RGB color space in the iterative segmentation phase. Results indicated that the Riemannian metric gave the cleanest edges and retained details better.

Shiozaki [43] proposed a new operator called an entropy operator for extracting

edges in monochrome as well as color pictures. In the later case, if H_R, H_G, and H_B are the entropies of the brightness in the regions of the R, G, and B components, and if a_R, a_G, a_B are the brightness levels of the respective components, then the entropy H in the region of the color picture was defined as:

$$H = q_R H_R + q_G H_G + q_B H_B, \tag{2.21}$$

where $q_R = a_R/(a_R + a_G + a_B)$, and so forth. The entropy is small when the change of brightness in the region is large, and it is small when the change of hue in the region is large. The proposed entropy operator was applied to real color images and the resultant edges were compared against the edges obtained using the color distance technique. The pictures showed improved definition and less noisy edges.

Delcroix and Abidi [14] presented two analytical approaches for the detection of edges in color images. In both approaches the problem of fusing correlated arrays of data was expressed as a minimization of a fusion functional. By assuming analyticity and expanding in a converging Taylor series, a first–order approximation was sought. Pairwise fusion, easier to implement and less sensitive to noise was the chosen alternative (as opposed to simultaneous three–way fusion). Delcroix and Abidi tested the pairwise fusion method on real (RGB) color images. The resultant fused edges were compared against the arithmetic average of the inputs and against the luminance map derived from original monochromatic pictures. Their results showed that the fused edge maps provided edges visually superior to any of those obtained from a single color channel.

Lee [29] proposed a vector gradient approach to detect boundaries in multidimensional data with multiple attributes (vector fields). Letting $\mathbf{f} : S \subset R^n \rightarrow R^m$ be a vector field, if $\mathbf{f}$ is a color image, then f_1, f_2, and f_3 might represent the red, green, and blue components of the image. The first–order Taylor expansion takes the form $\mathbf{f}(\mathbf{x}+\mathbf{a}) = \mathbf{f}(\mathbf{x}) + \mathbf{f}'(\mathbf{x})\mathbf{a}$, where $\mathbf{f}'(\mathbf{x})$ is an $m \times n$ matrix $\mathbf{D}(\mathbf{x})$:

$$\mathbf{f}'(\mathbf{x}) = \mathbf{D}(\mathbf{x}) = [d_{ij}] = [D_j f_i], \tag{2.22}$$

where $D_j f_i$ is the first partial derivative of the ith component of $\mathbf{f}$ with respect to the jth component of $\mathbf{x}$. A unit displacement $\mathbf{u}$ in the spatial domain will correspond to a displacement $\mathbf{d} = \sqrt{\mathbf{u^T D^T D u}}$ in the color (attribute) domain. Since the eigenvector of $\mathbf{D^T D}$ that corresponds to the largest eigenvalue is the vector that maximizes $\mathbf{d}$, then the square root of the largest eigenvalue and its corresponding eigenvector are the equivalents of the gradient magnitude and the gradient direction of a scalar field. Lee also analyzed the statistics of the noise, assuming stationary, white, and Gaussian noise, and assuming that all attributes had the same amount of noise. He compared the results (jump edges) detected by the scalar gradient and by the vector gradient for two color images degraded by additive Gaussian noise. Fewer edges were detected by the vector gradient than by the scalar gradient. Also, the edges that differed were mostly very short and visually insignificant. Hence, Lee concluded that the use of the vector gradient allowed avoiding some of the insignificant edges which were caused by noise.

Cumani *et al.* [12] proposed an extension of zero–crossing methods applied to gray–scale images to multispectral images. Contours were detected as the

zero–crossings of a suitable second–order differential operator were applied to a Gaussian smoothed color image. Given an m–band image represented by m scalar luminance functions $f_i(x,y)$, and given a unit vector $\vec{u} = (u_1, u_2)$ in the image plane, the *multispectral contrast function* in the direction $\vec{u}$ was defined as

$$S(x,y,u) = Eu_1^2 + 2Fu_1u_2 + Gu_2^2, \tag{2.23}$$

where

$$E = \sum_{i=1}^{m}(\frac{\partial f_i}{\partial x})^2, \quad F = \sum_{i=1}^{m}(\frac{\partial f_i}{\partial x})(\frac{\partial f_i}{\partial y}), \quad G = \sum_{i=1}^{m}(\frac{\partial f_i}{\partial y})^2. \tag{2.24}$$

The maximal value of S is $\lambda(x,y) = E + G + \sqrt{(E-G)^2 + 4F^2}/2$, and this maximum occurs when $\vec{u}$ is the eigenvector $\vec{n}$ corresponding to the maximum eigenvalue of the form

$$\vec{n} = (\cos\theta, \sin\theta), \quad \theta = \arctan(\frac{2F}{E-G}). \tag{2.25}$$

Cumani *et al.* then proposed considering the zero–crossings of the derivative of $\lambda(x,y)$ in the direction $\vec{n}$ as candidates for edge detection. The method was tested on natural color images, taking the R, G, and B channels as input data and assuming a pure Euclidean metric in RGB space.

Trehanias and Venetsanopoulos [51] proposed a class of color edge detection operators based on vector order statistics. Since the ordering of multivariate data is not uniquely defined, they chose the reduced or aggregate ordering (R–ordering) because it gives a natural definition of the vector median as the first sample of the sorted vectors. Three edge detectors of increasing complexity were proposed: vector range (VR) edge detector, vector dispersion edge detector (VDED), and

maximum vector dispersion edge detector (MVD). For quantitative evaluation, an artificial color image was created, and Pratt's figure of merit was used as a performance measure. A number of edge detection experiments were conducted using Gaussian, double–exponential, and impulsive noise for various noise levels. Subjective evaluations were also reported for real images.

Cebrián *et al.* [9] studied four methods for edge detection in multispectral remote sensing images. These methods were (1) the components averaged method, (2) the multispectral contrast method, (3) the maximal gradient method, and (4) the method which combines the edge images of the image bands. A brief description of each method follows. (1) In this method, a one–band image is computed as the average of the different bands which belong to the multispectral image. The edge image will be the result of applying the one–band second directional derivative (SDD) method to this average image. (2) This is the method proposed by Cumani *et al.* [12] but with a new test for local maximality of the multispectral contrast. (3) This is a new method that combines the gradients of the different spectra and assigns to each point P of the image plane the gradient at P of the band with maximal gradient norm at P. (4) This is a new method in which the SDD edge detection technique was applied to each image component. The resultant edge images were superposed and then thinned. Cebrián *et al.* tested the various methods on real three-band remote sensing images. They concluded that although the fourth method was the slowest, it was the only method that preserved all the edges of the image components.

2.5 Roof Edge Detection Techniques

Until very recently, most of the references to edge detection appearing in the literature considered only one kind of edge — step edges. The availability of range data has forced researchers to consider another kind of edge — roof edges. These edges occur whenever there is a significant first–order discontinuity. Since our database includes TV and range image pairs, we felt it necessary to include a review of roof edge detection techniques. The surveyed papers are ordered chronologically.

Godin and Levine [19] presented an approach for extracting a structured edge map from range images acquired using an active triangulation laser rangefinder. The method consisted of five main modules: edge detection, contour following, corner detection, limb and occluding edge detection, and junction building. (The first two steps shall be described here). The approach was useful for objects with planar and curved faces. For the detection of jump edges (discontinuities in depth), they used local maxima of depth difference and shadow analysis. Creases (discontinuities in surface orientation) were detected using estimates of the tangent plane at a point by performing a least–squares fitting of a small plane over a 3×3 patch. Crease edge detection was attempted only at locations sufficiently distant from a previously detected jump edge. A crease was detected if the angular

difference between a pair of planes in at least one of four directions (row, column, diagonals) was larger than a threshold and if the two planes intersected properly at the middle point to form a dihedral partition. Next, edge elements were organized into segments that were estimated to be continuous in depth by using a contour tracking procedure. The edge segments were labeled *convex*, *concave*, or *jump*.

Since the surfaces of objects are not usually entirely smooth but piecewise smooth, Yokoya and Levine [55] proposed to characterize local surface properties using the signs of Gaussian (K) and mean (H) curvatures (defined for smooth surfaces) and to combine them with jump and roof–edge maps for reliable segmentation. The signs of K and H yield a set of eight surface primitives. Since they are local characteristics of a surface, they can be used to characterize a surface when occlusion is a problem. By fitting a biquadratic form to a window, a set of coefficients is obtained. These coefficients are combined to obtain a set of partial derivative estimators. From these, the Gaussian and mean curvatures are computed. To accurately estimate the surface curvatures, the surface fit window must not overlap a discontinuity in the range data. So, for each point a window is chosen that provides a minimum fitting error among the covering windows. The computational flow consists of three major steps:

1. Local surface characterization. First and second partial derivatives are estimated at each surface point. From them the surface normal, and surface curvatures are calculated.

2. Initial segmentation. Three maps are computed: a region–based curvature

map (K–H sign map), a jump–edge map, and a roof–edge map.

3. Final segmentation. The initial segmentation maps are combined to produce the final segmentation in which each region must satisfy the conditions for a surface region.

At each point in the range image, the roof edge magnitude, M_r, was computed as the maximum angular difference between adjacent unit surface normals ($\vec{n}(x,y)$). Analytically this is,

$$M_r(x,y) = \max\left\{\arccos\left(\frac{\vec{n}(x,y)\cdot\vec{n}(x+i,y+j)}{|\vec{n}(x,y)||\vec{n}(x+i,y+j)|}\right)\right\}.$$

Al–Hujazi and Sood [4] proposed using the absolute value of the residual (AR) between the original (range) image and a Gaussian smoothed image for detecting step and roof edges. The following multi–step process illustrates the method.

1. Detect step edges using a small scale filter and threshold T_s.

2. Detect roof edges by computing the AR in four directions using a large Gaussian mask. Step edges detected in Pass 1 are used to restrict the neighborhood size of the filter. If the AR is a local maximum and higher than a chosen threshold (T_{ra}), in any of the four directions, it is marked as a roof edge point.

3. Increase accuracy in detecting roof edges by recalculating the AR, but this time both step edges and roof edges are used to restrict the size of the filter. Again, they used a large window, but this time the threshold was (T_{rb}). As

before, if $AR > T_{rb}$ for any of the four chosen directions, then the pixel is marked as a roof edge point.

4. Close small gaps and thinning.

Cung *et al.* [13] presented a scheme for multi–scale edge detection and classification in range images. Their method involved detecting significant jump and roof edges at an initial coarse resolution, segmenting coarse edges into smaller segments having smooth magnitude and direction, focusing of coarse resolution edges, and classifying of focused edge segments. Given a noisy range image, an initial smoothing stage (using a Gaussian smoother) preceded both roof and jump edge detection. In particular, finding the roof edges involved computing

$$mr_1(x, y, \sigma) = \sqrt{s_{xx} + s_{xy}}, \tag{2.26}$$

$$dr_1(x, y, \sigma) = \arctan(s_{xy}/s_{xx}), \tag{2.27}$$

$$mr_2(x, y, \sigma) = \sqrt{s_{yx} + s_{yy}}, \tag{2.28}$$

$$dr_2(x, y, \sigma) = \arctan(s_{yy}/s_{yx}). \tag{2.29}$$

The set of roof edge points $R(\sigma)$ was computed as $R(\sigma) = R_1(\sigma) \cup R_2(\sigma)$, where $R_i(\sigma)$ is the set of points that are local maxima of mr_i, $i = 1, 2$. Experimental results were provided for both synthetic and real 256×256 range images.

Fan [16] proposed a method for roof edge detection consisting of five steps: (1) adaptive smoothing through Saint Marc operators, (2) curvature computation, (3) maxima detection, (4) hysteresis, and (5) thinning. The output of the algorithm

were three data sets (*images*) corresponding to positive extrema, zero–crossings and negative extrema. From this information, edges may be classified further as convex or concave creases. Another procedure (Canny's) was employed for detection of depth discontinuities. Curves corresponding to depth or surface discontinuities were then extended as necessary to become the boundaries of closed patches. A set of features was then extracted and used in a model–based object recognition system.

Gonzalez *et al.* [20] proposed, as a part of their range image segmentation system, detecting roof edges by finding discontinuities in the direction of the surface normals. If r_x and r_y represent the partials of the function $r(x, y)$, then an approximation to the surface normal was computed using $\gamma(x, y) = \arctan(r_y/r_x)$. Then γ was scaled and clipped (for increased robustness). A 5×5 median filtering step was applied to the thresholded image to obtain the surface normal image. After a gradient operation, they obtained the image g_r. Another gradient image, g_s, was obtained after a 3×3 median filtering of the input range image. The final edge map, containing both jump and roof edges, was obtained after implementing a fusion step that combined g_r and g_s and a subsequent morphological stage.

Baccar and Abidi [5] proposed a dual–track approach for detecting step and roof edges. To compute roof edges, they proposed to compute curvature magnitude as $CM = \sqrt{H^2 - K}$, where H is the mean curvature and K is the Gaussian curvature. They smoothed the noisy range image with three Gaussian masks (sizes 5, 7, and 9) to obtain three CM images. Using these CM images, three binary

images were generated through a non–maxima suppression algorithm. A pixel was considered a roof–edge pixel if $CM > 0.125$. The resultant binary images were then merged through a binary AND operator to generate the final roof edge map.

CHAPTER 3

PROBLEM FORMULATION

In this chapter we will develop the analytical formulation of the proposed fusion technique and apply it to the problems of edge detection and surface reconstruction. The proposed formulation will be applied to both gray and color step edge detection. The surface reconstruction methodology will be applied to both dense and sparse depth maps. In the last section of this chapter, we will show a roof edge detection scheme based on the concepts developed earlier in this chapter.

3.1 Regularized Edge Detection

Edge detection is an ill–posed problem [39]. Mathematically, ill–posed problems are those whose solutions either do not exist, are not unique, or do not depend continuously on the data. Edge detection (numerical differentiation), the detection of physical changes such as object boundaries, is an ill–posed problem because its solution does not depend continuously on the data. To regularize an ill–posed problem and make it well posed, we have to introduce constraints. The basic idea of regularization, as formulated by Tikhonov [49], is to restrict the space of acceptable solutions by choosing the function that minimizes the appropriate

functional. The problem can be stated as that of finding the regularized function z from the data y such that the following functional is minimized:

$$\|\mathbf{A}z - y\|^2 + \lambda \|\mathbf{P}z\|^2, \tag{3.1}$$

where $\mathbf{A}$ is an operator that samples the function z on a lattice, λ is the regularization parameter that controls the degree of regularization of the solution and its closeness to the data, and $\mathbf{P}$ is a stabilizing functional. A natural choice for $\mathbf{P}$ is Tikhonov's stabilizing functional

$$\|\mathbf{P}z\|^2 = \int \sum_{r=0}^{p} c_r(\zeta)(\frac{d^r z}{d\zeta^r})^2 \, d\zeta, \tag{3.2}$$

where $c_r(\zeta)$ are nonnegative weighting functions. In one dimension, the usual choice is $\mathbf{P} = d^k/dx^k$, with $k = 1$ or 2. The choice $k = 2$ corresponds to a constraint of smoothness on the intensity profile z. The physical justification is that the noise–free image has to be smooth in the sense that its derivatives must be bounded because the image is band–limited by the optics. The smoothness constraint eliminates the noise caused by the sampling and transduction processes and makes the differentiation process stable. The standard regularization method for the one–dimensional edge detection problem can then be stated as follows: Find the function f that minimizes

$$\sum_i (y_i - f(x_i))^2 + \lambda \int (f''(x))^2 \, dx. \tag{3.3}$$

3.2 One–Dimensional Case

Based upon the standard regularization theory, as described above, we shall introduce, in order of increasing complexity, a series of fusion functionals.

First Functional

$$\int (f_1(x) - f(x))^2 \, dx + \lambda^2 \int (f'(x))^2 \, dx \tag{3.4}$$

In this functional, $f_1(x)$ corresponds to the given data and λ is a positive regularization parameter. We employ a first–order stabilizer and have replaced the summation by an integral.

Second Functional

$$\int (f_1(x) - f(x))^2 \, dx + \int \lambda(x)(f'(x))^2 \, dx \tag{3.5}$$

In this functional, we allow for the regularization parameter to be space–variant. This extends our method beyond standard regularization.

Third Functional

$$\int (f_1(x) - f(x))^2 \, dx + \int [\beta(x)(f''(x))^2] \, dx \tag{3.6}$$

In this functional, we have a second order smoothing term to allow for better detection of different kinds of edges. If $\beta(x)$ were a constant, then this functional would correspond to the standard regularized functional.

Fourth Functional

$$\int [(f_1(x) - f(x))^2]\, dx + \int [\lambda(x)(f'(x))^2 + \beta(x)(f''(x))^2]\, dx \tag{3.7}$$

In this functional, we have added both smoothing terms and allowed for space–variant regularization parameters.

Fifth Functional

$$\int [\omega_1(f_1(x) - f(x))^2 + \omega_2(f_2(x) - f(x))^2]\, dx + \int [\lambda(x)(f'(x))^2 + \beta(x)(f''(x))^2]\, dx \tag{3.8}$$

In this expression, $f_1(x)$ and $f_2(x)$ represent two independent knowledge sources. We also allow for different weights (ω_1, ω_2) to be associated to each knowledge source. More precisely, we assign different weights to the constraints.

Sixth Functional

$$\sum_{i=1}^{m} \int \omega_i(f_i(x) - f(x))^2\, dx + \int [\lambda(x)(f'(x))^2 + \beta(x)(f''(x))^2]\, dx \tag{3.9}$$

We have generalized the fifth functional to m knowledge sources.

Seventh Functional

$$\sum_{i=1}^{m} \int \omega_i(x)(f_i(x) - f(x))^2\, dx + \int [\lambda(x)(f'(x))^2 + \beta(x)(f''(x))^2]\, dx \tag{3.10}$$

In this functional, the weights $\omega_i(x)$ are now space–variant and should be related to the noise characteristics of each data source. Naturally, we should give more credit to the less noisy sources and less credit to the noisier ones.

3.3 Two–Dimensional Case

We can obtain the two–dimensional formulation by generalizing the one–dimensional case as follows:

$$\sum_{i=1}^{m} \int\int \omega_i(\mathbf{x})(f_i(\mathbf{x}) - f(\mathbf{x}))^2 \, d\mathbf{x} + \int\int [\lambda(\mathbf{x})(\|\nabla f(\mathbf{x})\|)^2 + \beta(\mathbf{x})(\Delta f(\mathbf{x}))^2] \, d\mathbf{x}. \tag{3.11}$$

In this functional, the weights $\omega_i(\mathbf{x})$ are space–variant and can be related to the noise characteristics of each data source. Alternatively, these weights can be user–specified based on some general properties of the various sensory channels. The $f_i(\mathbf{x})$ represent the data sources, $\lambda(\mathbf{x})$ and $\beta(\mathbf{x})$ are space–variant regularization parameters, ∇ and Δ represent first– and second–order stabilizers, respectively.

Certainly, this functional could be generalized one step further by allowing the domain to be R^n and by employing higher order stabilizers.

Let us show now an example of the procedures employed in solving a particular case from this family of functionals. Let $m = 1$, $\beta(\mathbf{x}) = 0$, and $\omega_1 = 1$. Let E be the resulting energy functional:

$$E = \int\int (f_1(\mathbf{x}) - f(\mathbf{x}))^2 \, d\mathbf{x} + \int\int [\lambda(\mathbf{x})(\|\nabla f(\mathbf{x})\|)^2] \, d\mathbf{x}. \tag{3.12}$$

There are several cases of interest, depending on the handling of the regularization parameter λ. These cases include:

$$E = \int\int (f_1(\mathbf{x}) - f(\mathbf{x}))^2 \, d\mathbf{x} + \lambda \int\int [(\|\nabla f(\mathbf{x})\|)^2] \, d\mathbf{x}, \tag{3.13}$$

$$E = \int\int (f_1(\mathbf{x}) - f(\mathbf{x}))^2 \, d\mathbf{x} + \int\int \lambda(\mathbf{x})[(\|\nabla f(\mathbf{x})\|)^2] \, d\mathbf{x}, \tag{3.14}$$

and

$$E = \int\int (f_1(\mathbf{x}) - f(\mathbf{x}))^2 \, d\mathbf{x} + \int\int [\lambda_h(\mathbf{x}) f_x^2(\mathbf{x}) + \lambda_v(\mathbf{x}) f_y^2(\mathbf{x})] \, d\mathbf{x}. \tag{3.15}$$

We shall illustrate a minimization procedure based on form 3.15 of the functional. For the two–dimensional case, this functional can be expressed as

$$\begin{aligned} E &= \int\int (f_1(x,y) - f(x,y))^2 \, dx \, dy + \\ &\quad \int\int [\lambda_h(x,y) f_x^2(x,y) + \lambda_v(x,y) f_y^2(x,y)] \, dx \, dy. \end{aligned} \tag{3.16}$$

When λ_h and λ_v are fixed, E is quadratic and, therefore, convex. This guarantees the uniqueness of the optimal solution f_{opt}. Letting

$$G(x,y,f,f_x,f_y) = (f_1(x,y) - f(x,y))^2 + \lambda_h(x,y) f_x^2(x,y) + \lambda_v(x,y) f_y^2(x,y), \tag{3.17}$$

the Euler–Lagrange equation for E is given by

$$\frac{\partial G}{\partial f} - \frac{\partial}{\partial x}\left[\frac{\partial G}{\partial f_x}\right] - \frac{\partial}{\partial y}\left[\frac{\partial G}{\partial f_y}\right] = 0, \tag{3.18}$$

or, after some manipulation,

$$f - \lambda_h f_{xx} - \lambda_v f_{yy} = f_1. \tag{3.19}$$

We may attempt to solve 3.19 directly, or we may employ an iterative procedure. An example of the latter assumes that f varies in time and employs

$$\frac{\partial f}{\partial t} = -\tau \frac{dE}{dt} = -\tau[2(f_1 - f) - 2\lambda_h f_{xx} - 2\lambda_v f_{yy}], \tag{3.20}$$

where τ is a time increment. In discrete form, the update expression for f has the form $f^{(t+1)}(i,j) = f^{(t)}(i,j) - \tau \frac{dE_d}{dt}$, where E_d represents the discrete form of the energy expression above. The iterations can be stopped whenever the difference between two consecutive solutions is below a chosen threshold.

3.4 Boundary Conditions: One–Dimensional Case

In order to have a better understanding of the behavior of the solutions under various conditions, we implemented the standard regularized edge detection technique for the case of a first–order stabilizing functional, i.e.,

$$E(\lambda, f) = \int (f_1(x) - f(x))^2 \, dx + \lambda^2 \int (f'(x))^2 \, dx. \tag{3.21}$$

Clearly, since this functional represents the starting point in our development, and since it allows for analytical solutions (for certain input functions), we implemented it first. Let us then consider the functional

$$E(\lambda, f) = \int (f_1(x) - f(x))^2 \, dx + \lambda^2 \int (f'(x))^2 \, dx. \tag{3.22}$$

We wish to minimize this functional for different values of the parameter λ. Let

$$G = (f_1(x) - f(x))^2 + \lambda^2 (f'(x))^2. \tag{3.23}$$

From the Calculus of Variations, this problem is equivalent to solving the following partial differential equation:

$$\frac{\partial G}{\partial f} - \frac{d}{dx}\left[\frac{\partial G}{\partial f'}\right] = 0, \tag{3.24}$$

where

$$\frac{\partial G}{\partial f} = -2(f_1(x) - f(x)), \tag{3.25}$$

and

$$\frac{d}{dx}\left[\frac{\partial G}{\partial f'}\right] = 2\lambda^2 f''(x). \tag{3.26}$$

After replacing, the Euler–Lagrange equation becomes

$$-\lambda^2 f''(x) + f(x) = f_1(x). \tag{3.27}$$

A comparison with a general second–order ordinary differential equation (ODE) reveals that $a(x) = -\lambda^2$, $b(x) = 0$, $c(x) = 1$, and $g(x) = f_1(x)$. From the homogeneous equation, we have

$$ap^2 + 1 = 0, \tag{3.28}$$

that is,

$$p_{1,2} = \pm 1/\lambda, \tag{3.29}$$

with the two solutions given by

$$\phi_1(x) = e^{x/\lambda} \tag{3.30}$$

and

$$\phi_2(x) = e^{-x/\lambda}. \tag{3.31}$$

Since the Wronskian $W(\phi_1, \phi_2) = \phi_1\phi_2' - \phi_1'\phi_2 = -2/\lambda$ is nonzero, we conclude that the solutions $\phi_1(x)$ and $\phi_2(x)$ are linearly independent, thus enabling us to employ Variation of Parameters. Let

$$\begin{aligned} y_p &= u_1(x)\phi_1(x) + u_2(x)\phi_2(x) \\ &= u_1(x)e^{x/\lambda} + u_2(x)e^{-x/\lambda}. \end{aligned} \tag{3.32}$$

There are two conditions to be satisfied:

$$u_1'\phi_1 + u_2'\phi_2 = 0 \tag{3.33}$$

and

$$au_1'\phi_1' + au_2'\phi_2' = g. \tag{3.34}$$

Solving these two equations in terms of u_1' and u_2, we get

$$u_1'(x) = \frac{\lambda g(x)e^{-x/\lambda}}{2a} = -\frac{\phi_2 g}{Wa} \tag{3.35}$$

and

$$u_2'(x) = -\frac{\lambda g(x)e^{x/\lambda}}{2a} = \frac{\phi_1 g}{Wa}. \tag{3.36}$$

After simplifying,

$$u_1'(x) = -\frac{ge^{-x/\lambda}}{2\lambda} \tag{3.37}$$

and

$$u_2'(x) = \frac{ge^{x/\lambda}}{2\lambda}, \tag{3.38}$$

from where

$$u_1(x) = -\frac{1}{2\lambda}\int_0^x ge^{-s/\lambda}\,ds \tag{3.39}$$

and

$$u_2(x) = \frac{1}{2\lambda}\int_0^x ge^{s/\lambda}\,ds. \tag{3.40}$$

Finally, the complete solution can be written as

$$f(x) = c_1e^{x/\lambda} + c_2e^{-x/\lambda} - e^{x/\lambda}\frac{1}{2\lambda}\int_0^x ge^{-s/\lambda}\,ds + e^{-x/\lambda}\frac{1}{2\lambda}\int_0^x ge^{s/\lambda}\,ds. \tag{3.41}$$

The latter expression represents a family of solutions. For a unique solution, we need to impose some boundary conditions. Naturally, two conditions are required for a second-order differential equation. As mentioned earlier, there are various sets of reasonable conditions. However, only one set arises naturally from

the minimization of the functional E. We show this now. Let us consider the functional

$$E = \int G(x, f, f')\, dx, \tag{3.42}$$

where the function G depends not only on x but also on the function $f(x)$ and its first derivative $df(x)/dx$. Let

$$\Phi(\epsilon) = \int_{x_0}^{x_1} G(x, f + \epsilon\eta, f' + \epsilon\eta')\, dx. \tag{3.43}$$

The derivative of Φ with respect to ϵ is given by

$$\frac{d\Phi(\epsilon)}{d\epsilon} = \int_{x_0}^{x_1} \left\{ \frac{\partial G}{\partial x}\frac{dx}{d\epsilon} + \frac{\partial G}{\partial u}\frac{du}{d\epsilon} + \frac{\partial G}{\partial v}\frac{dv}{d\epsilon} \right\} dx, \tag{3.44}$$

where $u = f + \epsilon\eta$, $v = f' + \epsilon\eta'$. The first term in the integral vanishes because x doesn't depend on ϵ. Letting ϵ go to zero, we obtain

$$\Phi'(\epsilon) = \Phi'(0) = \int_{x_0}^{x_1} \left\{ \frac{\partial G}{\partial f}\eta + \frac{\partial G}{\partial f'}\eta' \right\} dx. \tag{3.45}$$

Integrating the second term by parts, we get

$$\int_{x_0}^{x_1} \frac{\partial G}{\partial f'}\eta'\, dx = \frac{\partial G}{\partial f'}\eta \Big]_{x_0}^{x_1} - \int_{x_0}^{x_1} \eta \frac{d}{dx}\left[\frac{\partial G}{\partial f'}\right] dx. \tag{3.46}$$

Combining this result with our previous equation, we have

$$\Phi'(0) = \int_{x_0}^{x_1} \eta(x) \left\{ \frac{\partial G}{\partial f} - \frac{d}{dx}\left[\frac{\partial G}{\partial f'}\right] \right\} dx = 0. \tag{3.47}$$

Since this result must be valid for all $\eta(x)$, then

$$\frac{\partial G}{\partial f} - \frac{d}{dx}\left[\frac{\partial G}{\partial f'}\right] = 0 \tag{3.48}$$

is the Euler–Lagrange equation for this problem, and the natural boundary conditions of this problem (at $x = x_0$ and $x = x_1$) are

$$\frac{\partial G}{\partial f'} = 0. \tag{3.49}$$

In particular, for our functional $E(\lambda, f)$ this condition reduces to

$$\begin{aligned} \frac{\partial G}{\partial f'} &= \frac{\partial[(f_1(x) - f(x))^2 + \lambda^2 (f'(x))^2]}{\partial f'} \\ &= 2\lambda^2 f'(x) \\ &= 0, \end{aligned} \tag{3.50}$$

or since λ is greater than zero, $f'(x_0) = 0$, and $f'(x_1) = 0$.

We have obtained, therefore, analytical expressions for the solution $f(x)$ for the first–order stabilizer. A similar procedure could be employed for finding solutions for the second–order stabilizer, but this time we would have to deal with a fourth–order differential equation.

3.5 Surface Reconstruction

Reconstruction of visible surfaces from sparse and noisy data is an important problem in computer vision. The list of applications include:

- Reconstruction of terrestrial topography from altimetry data,
- Construction of CAD models from range data of physical objects,
- Reconstruction of visible surfaces from sparse 3–D stereo data, and

- Reconstruction of surfaces (and volumes) from tomographic data.

The problem of surface (or more generally, object) reconstruction from image data is, however, ill–posed. The main reason for this difficulty is that much of the information is lost during the projection from the 3–D world into the 2–D images.

The development of laser ranging sensors has alleviated this problem somewhat, but there are still various factors that contaminate the sensed depth maps. Worth mentioning are the loss of information due to specular reflections and/or absorption, the ambiguities arising from the phase–shift detection schemes, and variations in the amplitude of the returned beam due to the material properties of the surface.

To obtain a unique and stable reconstruction of the visible surfaces in the scene, supplementary information must be added so the problem becomes well posed.

Before addressing our formulation of the surface reconstruction problem, we need to provide some basic definitions.

Definition 3.1 *The* **explicit** *form of surface reconstruction describes the surface by the equation* $z = f(x, y)$*, where* z *is the distance from the viewer to the object points and* (x, y) *are the image plane coordinates.*

Definition 3.2 *The* **implicit** *form of a surface is a function of three variables* f *given by* $f : \Re^3 \mapsto \Re | f(x, y, z) = constant$*, where* (x, y, z) *are the Cartesian coordinates of the surface points.*

Definition 3.3 *The* **parametric** *description of a surface S is a map from a region of the plane $\Re^2$ (usually a rectangle) with Cartesian coordinates (u, v) to the surface, for instance $M : \Re^2 \mapsto S$.*

Parametric representations are just one kind of full 3–D representations. Other 3–D representations are surface meshes, particle systems, and volumetric models. The implicit and parametric forms are employed when we wish to estimate the parameters of a model (e.g. quadrics or superquadrics.) The explicit form is employed in variational formulations where we are minimizing some kind of energy functional. In our work we shall be employing the explicit form of surface representation in image– or viewer–centered reconstructions.

The variational formulation of the surface reconstruction problem allows physical modelling of the surface. The surface can now be seen as a set of mobile particles subject to known forces and constraints. The change from one state to another can be regulated by known evolution equations.

In regularized surface reconstruction, a visible surface is computed from a set of (possibly sparse) constraints $d(x, y)$ by minimizing the function $u(x, y)$ which minimizes the weighted sum of two energies:

$$E(u) = E_d(u, d) + \lambda E_s(u, d).$$

The data constraint functional E_d measures the distance between the solution and the data. The stabilizing (smoothness) functional E_s measures the smoothness of the solution. The regularization parameter λ controls the balance between these

two energy terms. Lower values of λ cause the solution to follow the data closely (clearly undesirable if the data is noisy). On the other hand, higher values of λ cause the solution to be very smooth (clearly undesirable if the data contains discontinuities).

We will now present the two stabilizers chosen to help solve the surface reconstruction problem. The first stabilizer measures the potential energy of a membrane (rubber sheet) and is given by

$$E_1 = \int\int (u_x^2 + u_y^2) dx dy.$$

This is a first–order stabilizer with C^0 smoothness of the surface. The associated Euler–Lagrange equation is the Laplace equation.

The second stabilizer measures the strain energy of a thin plate with the Poisson constant $\sigma = 0$, and is given by

$$E_2 = \int\int (u_{xx}^2 + 2u_{xy}^2 + u_{yy}^2) dx dy.$$

This is a second–order stabilizer with C^1 smoothness of the surface. The associated Euler–Lagrange equation is the biharmonic equation.

First–Order Stabilizer

We now show implementation details concerning the first–order stabilizer. Recall the functional 3.15

$$E(f(\mathbf{x})) = \int\int (f_1(\mathbf{x}) - f(\mathbf{x}))^2 \, d\mathbf{x} + \int\int [\lambda_h(\mathbf{x}) f_x^2(\mathbf{x}) + \lambda_v(\mathbf{x}) f_y^2(\mathbf{x})] \, d\mathbf{x} \quad (3.51)$$

The corresponding discrete form is given by

$$E(f_{ij}) = \sum_i \sum_j (f_{ij} - d_{ij})^2 + \sum_i \sum_j [\lambda^h_{ij}(f_{ij} - f_{i-1,j})^2 + \lambda^v_{ij}(f_{ij} - f_{i,j-1})^2], \quad (3.52)$$

where d_{ij} represents the (single) data source, and λ^h_{ij} and λ^v_{ij} represent horizontal and vertical regularization parameters, respectively. Since we are looking for the unknown f_{ij} that minimizes the energy functional, we will minimize it with respect to f_{ij}, resulting in

$$\begin{aligned} \frac{\partial E}{\partial f_{ij}} &= (f_{ij} - d_{ij}) + \lambda^h_{ij}(f_{ij} - f_{i-1,j}) + \lambda^h_{i+1,j}(f_{ij} - f_{i+1,j}) + \\ &\quad \lambda^v_{ij}(f_{ij} - f_{i,j-1}) + \lambda^v_{i,j+1}(f_{ij} - f_{i,j+1}), \end{aligned} \quad (3.53)$$

or after rearranging,

$$\begin{aligned} \frac{\partial E}{\partial f_{ij}} &= (1 + \lambda^h_{ij} + \lambda^h_{i+1,j} + \lambda^v_{ij} + \lambda^v_{i,j+1}) f_{ij} - d_{ij} - \\ &\quad (\lambda^h_{ij} f_{i-1,j} + \lambda^h_{i+1,j} f_{i+1,j} + \lambda^v_{ij} f_{i,j-1} + \lambda^v_{i,j+1} f_{i,j+1}) \end{aligned} \quad (3.54)$$

or $\frac{\partial E}{\partial f_{ij}} = a_{ij} f_{ij} - b_{ij} d_{ij} - \Delta ij$. A reasonable choice for finding the minimal solution is the iterative scheme Successive Overrelaxation (SOR). The general form of SOR is

$$f^{n+1}_{ij} = f^n_{ij} - \frac{\omega}{T} \frac{\partial E}{\partial f_{ij}}, \quad (3.55)$$

where $0 < \omega < 2$ is the relaxation parameter, and T is a factor that has to be chosen to insure convergence. For interior nodes, choose $T = (1 + \lambda^h_{ij} + \lambda^h_{i+1,j} + \lambda^v_{ij} + \lambda^v_{i,j+1})$. The iterative procedure has to be adjusted for exterior nodes.

Assuming the correspondence $i \sim x$, $j \sim y$, and assuming a right-hand coordinate system with y pointing downwards, Table 3.1 shows the proper corner conditions.

Table 3.1: Valid corner conditions

Corner	a_{ij}	Δij
NW	$1+\lambda^h_{i+1,j}+\lambda^v_{i,j+1}$	$\lambda^h_{i+1,j}f_{i+1,j}+\lambda^v_{i,j+1}f_{i,j+1}$
NE	$1+\lambda^h_{ij}+\lambda^v_{i,j+1}$	$\lambda^h_{ij}f_{i-1,j}+\lambda^v_{i,j+1}f_{i,j+1}$
SW	$1+\lambda^h_{i+1,j}+\lambda^v_{ij}$	$\lambda^h_{i+1,j}f_{i+1,j}+\lambda^v_{ij}f_{i,j-1}$
SE	$1+\lambda^h_{ij}+\lambda^v_{ij}$	$\lambda^h_{ij}f_{i-1,j}+\lambda^v_{ij}f_{i,j-1}$

Table 3.2 shows the corrections needed in the iterative procedure for pixels lying along the exterior edges, excluding those in the corners.

Table 3.2: Conditions along exterior edges

Edge	a_{ij}	Δij
N	$1+\lambda^h_{i,j}+\lambda^h_{i+1,j}+\lambda^v_{i,j+1}$	$\lambda^h_{i,j}f_{i-1,j}+\lambda^h_{i+1,j}f_{i+1,j}+\lambda^v_{i,j+1}f_{i,j+1}$
E	$1+\lambda^h_{ij}+\lambda^v_{i,j}+\lambda^v_{i,j+1}$	$\lambda^h_{ij}f_{i-1,j}+\lambda^v_{i,j}f_{i,j-1}+\lambda^v_{i,j+1}f_{i,j+1}$
S	$1+\lambda^h_{i,j}+\lambda^h_{i+1,j}+\lambda^v_{i,j}$	$\lambda^h_{i+1,j}f_{i+1,j}+\lambda^h_{i+1,j}f_{i+1,j}+\lambda^v_{ij}f_{i,j-1}$
W	$1+\lambda^h_{ij}+\lambda^v_{ij}+\lambda^v_{i,j+1}$	$\lambda^h_{i+1,j}f_{i+1,j}+\lambda^v_{ij}f_{i,j-1}+\lambda^v_{ij}f_{i,j+1}$

Second–Order Stabilizer

We now show the details of the procedures regarding the computer implementation of the second–order stabilizer. Let us begin with the continuous formulation:

$$E=\int\int[f(\mathbf{x})-f_1(\mathbf{x})]^2\,d\mathbf{x}+\lambda\int\int[f_{xx}^2+2f_{xy}^2+f_{yy}^2]\,d\mathbf{x} \tag{3.56}$$

The discrete form of this functional is

$$\begin{aligned} E(f_{ij})= \; & \textstyle\sum_i\sum_j(f_{ij}-d_{ij})^2+\lambda\{\sum_i\sum_j(f_{i-1,j}-2f_{i,j}+f_{i+1,j})^2+ \\ & 2\textstyle\sum_i\sum_j(f_{i,j-1}-f_{i+1,j}-f_{i,j+1}+f_{i+1,j+1})^2+ \\ & \textstyle\sum_i\sum_j(f_{i,j-1}-2f_{i,j}+f_{i,j+1})^2\}, \end{aligned} \tag{3.57}$$

where d_{ij} is now the discrete form of $f_1(\mathbf{x})$. As in the case of the first–order stabilizer, we shall use the SOR iterative method to find the unknown f_{ij}. This

requires finding $\frac{\partial E}{\partial f_{ij}}$ and adapting the update rule for conditions around the boundaries. This time, however, it is more practical to show the proper computational molecules (a graphical device to visualize the equations) than to give a table with expressions. Let us begin with the derivation of the updating rule for interior nodes. From the three terms inside the brackets (the discrete approximations to the partial derivatives of f), we obtain:

$$\begin{aligned} S_1 &= (f_{i+1,j} - 2f_{i,j} + f_{i-1,j})^2 && \text{(3.58-a)} \\ &\quad + (f_{i,j} - 2f_{i-1,j} + f_{i-2,j})^2 && \text{(3.58-b)} \\ &\quad + (f_{i+2,j} - 2f_{i+1,j} + f_{i,j})^2, && \text{(3.58-c)} \end{aligned}$$

$$\begin{aligned} S_2 &= (f_{i,j-1} - 2f_{i,j} + f_{i,j+1})^2 && \text{(3.59-a)} \\ &\quad + (f_{i,j} - 2f_{i,j-1} + f_{i,j-2})^2 && \text{(3.59-b)} \\ &\quad + (f_{i,j+2} - 2f_{i,j+1} + f_{i,j})^2, && \text{(3.59-c)} \end{aligned}$$

and

$$\begin{aligned} S_3 &= (f_{i,j} - f_{i+1,j} - f_{i,j+1} + f_{i+1,j+1})^2 && \text{(3.60-a)} \\ &\quad + (f_{i-1,j} - f_{i,j} - f_{i-1,j+1} + f_{i,j+1})^2 && \text{(3.60-b)} \\ &\quad + (f_{i-1,j-1} - f_{i,j-1} - f_{i-1,j} + f_{i,j})^2 && \text{(3.60-c)} \\ &\quad + (f_{i,j-1} - f_{i+1,j-1} - f_{i,j} + f_{i+1,j})^2. && \text{(3.60-d)} \end{aligned}$$

Taking the partial derivative with respect to $f_{i,j}$ and adding, we get:

$$\sum_{k=1}^{3} \frac{\partial S_k}{\partial f_{ij}} = 2f_{i-2,j} + 4f_{i-1,j-1} - 16f_{i-1,j} + 4f_{i-1,j+1} + 2f_{i,j-2} - 16f_{i,j-1} + 40f_{i,j} - 16f_{i,j+1} + 2f_{i,j+2} + 4f_{i+1,j-1} - 16f_{i+1,j} + 4f_{i+1,j+1} + 2f_{i+2,j} \quad (3.61)$$

Figure 3.1 shows the corresponding computational molecule. For the other nodes,

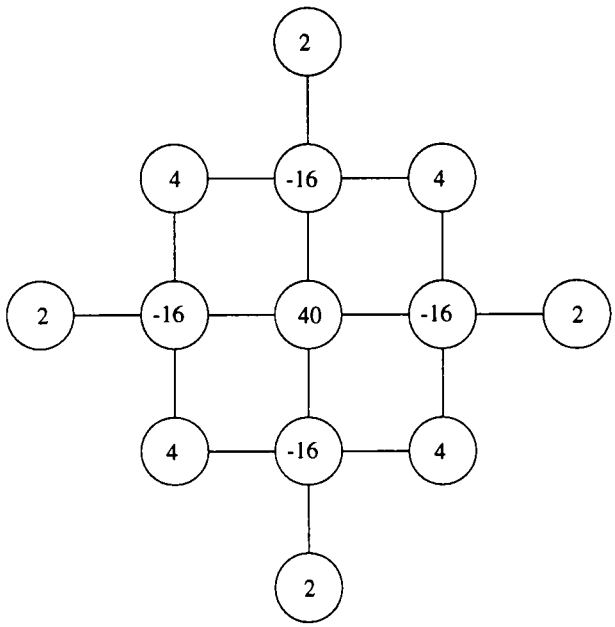


Figure 3.1: *Computational molecule for interior nodes.*

the correct expressions are found by realizing that only a few terms (in the general equations for S_1, S_2, and S_3) apply. In what follows, we shall consider a typical case for the various border conditions. The other cases that can be derived from rotations and reflections of the given model follow the same methodology and will be given in their final form only.

For the NW corner ($i = j = 0$), only Eqs. 3.58-c, 3.59-c, and 3.60-a apply. The

corresponding sum of partial derivatives is, therefore,

$$\sum_{k=1}^{3} \frac{\partial S_k}{\partial f_{ij}} = 8f_{i,j} - 8f_{i+1,j} - 8f_{i,j+1} + 4f_{i+1,j+1} + 2f_{i+2,j} + 2f_{i,j+2}.$$

Figure 3.2 shows the four molecules for the corner pixels. For elements along

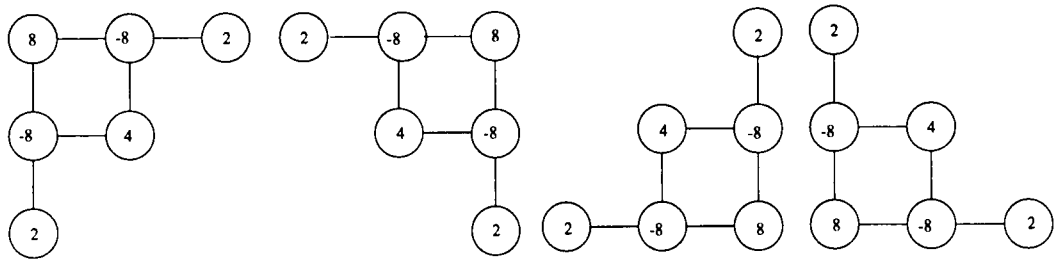


Figure 3.2: *Computational molecules for corner nodes.*

the west column (and all other exterior rows) at least two pixels away from the corners, we find that equations 3.58-a, 3.58-b, 3.58-c, 3.59-c, 3.60-a, 3.60-b apply. Hence,

$$\sum_{k=1}^{3} \frac{\partial S_k}{\partial f_{ij}} = 22f_{i,j} - 12f_{i+1,j} - 12f_{i-1,j} + 2f_{i-2,j} + 2f_{i+2,j} - 12f_{i,j+1} + 4f_{i+1,j+1} + 4f_{i-1,j+1},$$

and the corresponding molecules are shown in Fig. 3.3. For elements along an exterior row, one pixel away from the corner, the following expressions apply (west row): Eqs. 3.58-a, 3.58-c, 3.59-c, and 3.60-a. The corresponding sum of partial derivatives is, therefore,

$$\sum_{k=1}^{3} \frac{\partial S_k}{\partial f_{ij}} = 20f_{i,j} - 12f_{i+1,j} - 8f_{i-1,j} + 2f_{i+2,j} + 2f_{i,j+2} - 12f_{i,j+1} + 4f_{i+1,j+1} + 4f_{i-1,j+1}.$$

Figure 3.4 shows the computational molecules for this case. For elements along rows adjacent to exterior row, at least two pixels away from the borders, the

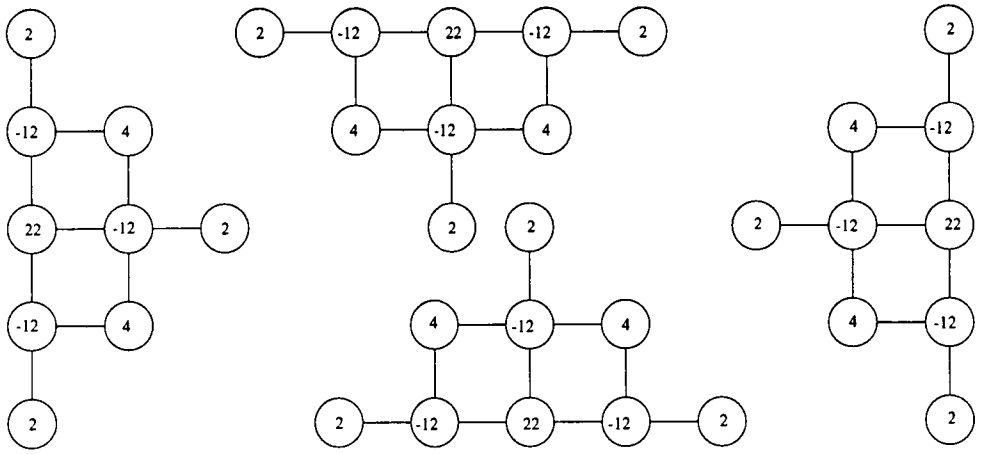


Figure 3.3: *Computational molecules for nodes along exterior rows.*

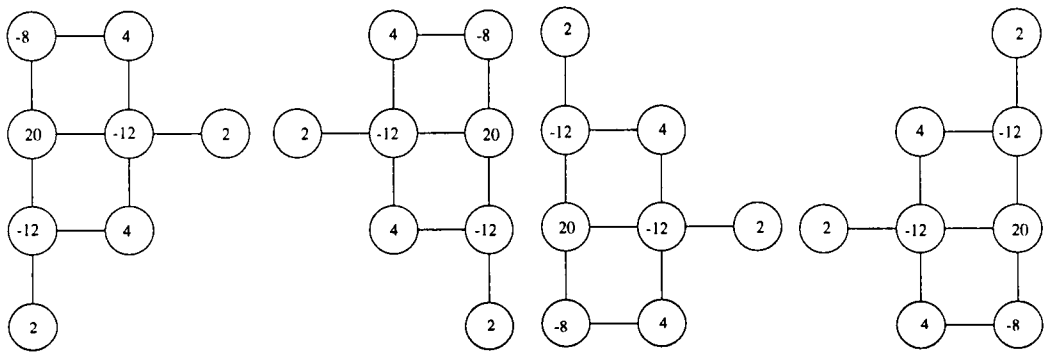


Figure 3.4: *Computational molecules for nodes along exterior rows one pixel away from the corner.*

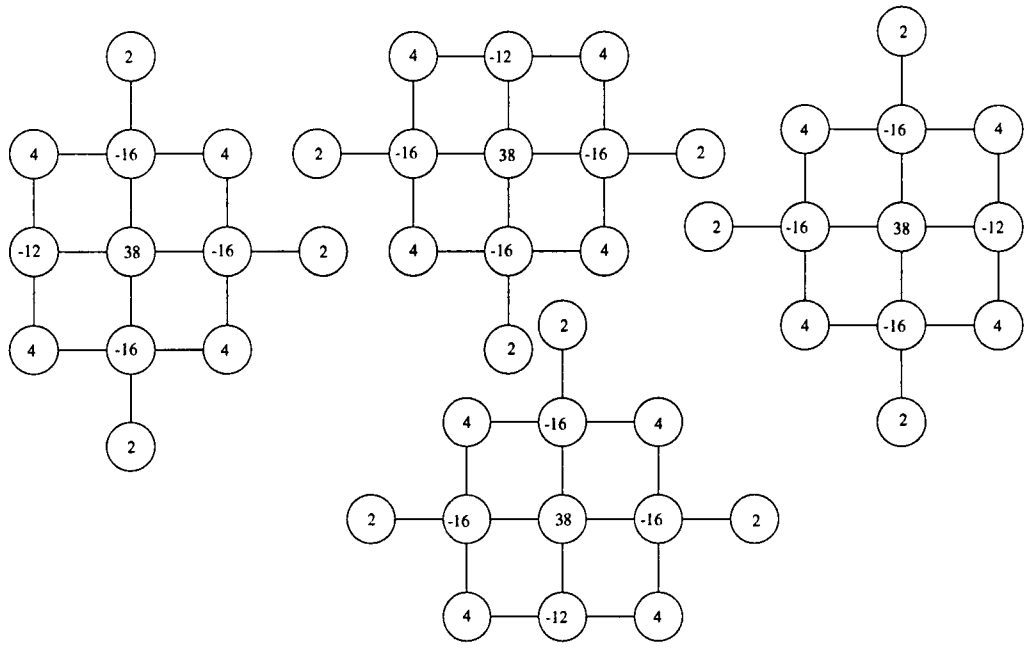


Figure 3.5: *Computational molecules for nodes along rows adjacent to exterior rows at least two pixels away from opposite sides.*

following expressions apply (west side): Eqs. 3.58-a, 3.58-b, 3.58-c, 3.59-a, 3.59-c, 3.60-a, 3.60-b, 3.60-c, and 3.60-d. The sum of partial derivatives becomes

$$\sum_{k=1}^{3} \frac{\partial S_k}{\partial f_{ij}} = -16f_{i+1,j} + 38f_{i,j} - 16f_{i-1,j} + 2f_{i-2,j} + 2f_{i+2,j} - 12f_{i,j-1} - 16f_{i,j+1} + 2f_{i,j+2} + 4f_{i+1,j+1} + 4f_{i-1,j+1} + 4f_{i-1,j-1} + 4f_{i+1,j-1}. \quad (3.62)$$

Figure 3.5 illustrates the corresponding computational molecules. The final set of molecules is obtained by considering the cases of pixels located at the "corners" adjacent to the exterior corners. In particular, for the NW location, the following are the valid expressions: Eqs. 3.58-a, 3.58-c, 3.59-a, 3.59-c, 3.60-a, 3.60-b,

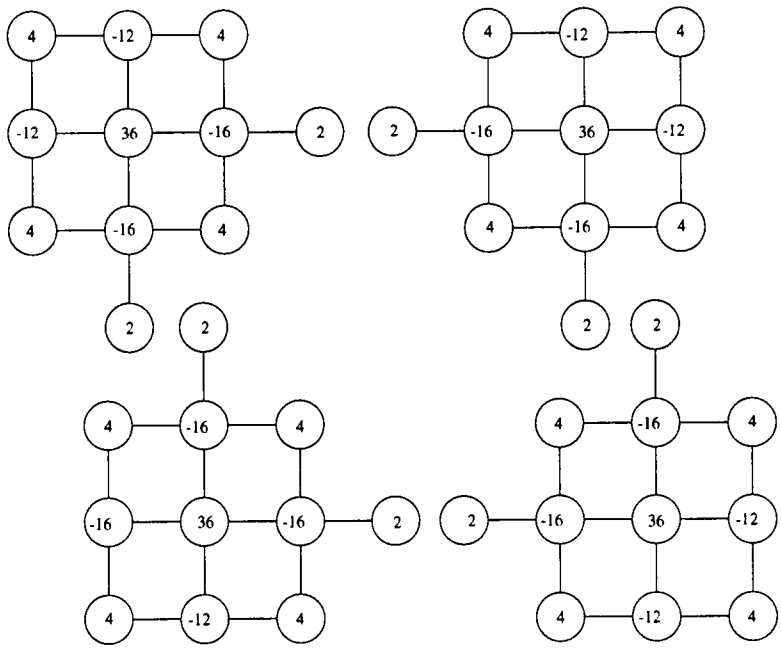


Figure 3.6: *Computational molecules for "corner" nodes one pixel away from the real corners.*

3.60-c, and 3.60-d. The sum of the corresponding partials is given by

$$\sum_{k=1}^{3} \frac{\partial S_k}{\partial f_{ij}} = -16f_{i+1,j} + 36f_{i,j} - 12f_{i-1,j} + 2f_{i+2,j} - 12f_{i,j-1} - 16f_{i,j+1} + 2f_{i,j+2} + 4f_{i+1,j+1} + 4f_{i-1,j+1} + 4f_{i-1,j-1} + 4f_{i+1,j-1}. \quad (3.63)$$

Figure 3.6 illustrates the corresponding computational molecules.

3.6 Color Edge Detection

The problem with measuring color, as opposed to measuring length (e.g., range) or weight, is that there is no single (scalar) physical scale for measuring color. In addition, three elements, light, object, and observer, are necessary

for the perception of color. Color perception is, therefore, subjective and depends on the responses of the color receptors in the retina and higher-level processes in the brain, which are subject to variations among individuals.

Various simplifying assumptions are required to handle the problem of color edge detection. First of all, we shall assume that the surfaces of the objects are *Lambertian* and, therefore, all the light that the surface reflects is independent of the angle of observation and depends only on the illumination E and the body reflection (or albedo) ρ. Real objects usually possess a mixture of diffuse and specular reflectivity.

The relationship between illumination and object reflectance is described by the color image irradiance equation:

$$I(i, e, g, \lambda, \vec{r}) = \rho(\lambda, \vec{r}) F(i, e, g) E(\lambda, \vec{r}),$$

where λ is the wavelength of the light; $\vec{r}$ is the position of the observed point; $\rho(\lambda, \vec{r})$, or the albedo, depends on the material properties of the surface at the point of reflection; $F(i, e, g)$ is a geometric factor considering the directions of illumination and observation; $E(\lambda, \vec{r})$ is the spectral distribution of the illuminant in space; and $I(i, e, g, \lambda, \vec{r})$, or the image irradiance, is the amount of light energy that falls onto the image plane at each wavelength.

Figure 3.7 provides a geometric explanation of the color image irradiance equation. In this figure, $\vec{s}$ is a unit vector in the illuminant direction, $\vec{n}$ is the unit normal vector, $\vec{v}$ is the unit vector in the viewer's direction, i is the angle of incidence, e is the view angle, and g is the phase angle.

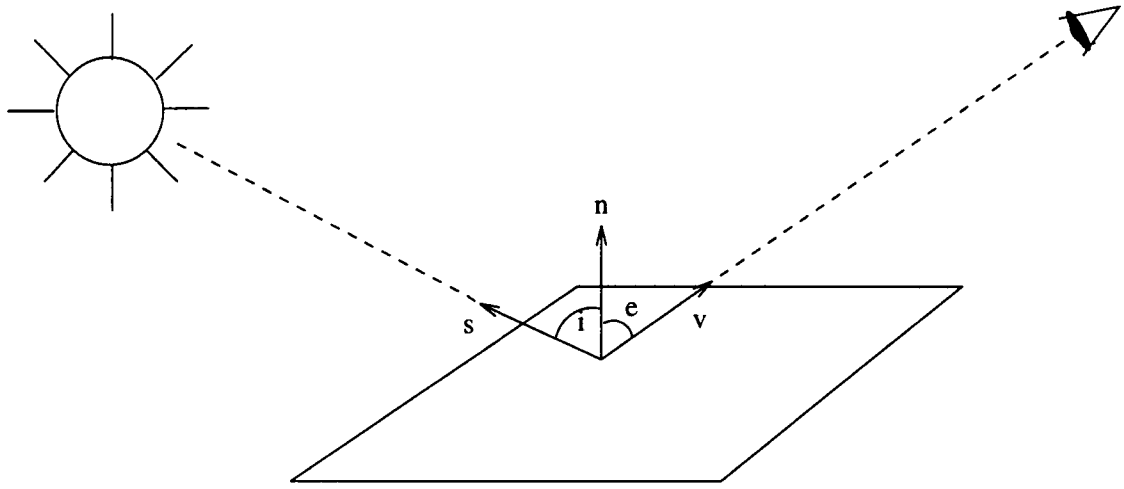


Figure 3.7: *Color image irradiance and photometric angles for the isotropic case.*

The recovery of the true color of an object from the sensed color image is an ill–posed problem, and, therefore, we should employ regularization in the inversion process.

The second assumption in our analysis is that all color signals can be represented by three variables. This is a reasonable constraint based on known facts about the color receptors (cones) in the retina and psychobiological experimentation. In particular, we shall assume that the red, green, and blue (RGB) components of the color image correspond to that of the standard observer. That is to say, the responses of the camera sensors are close to those of the standard observer.

The third assumption is that the illuminant color has a constant spectral distribution across the visible spectrum.

Our last assumption is that there are no interobject reflections. All four assumptions would make the problem of recovering ρ from I well posed. However,

because we shall be dealing with real images and because we are trying to find edges from noisy data (a well–known ill–posed problem), we shall employ the regularization technique in our color edge detection scheme.

3.7 Roof Edge Detection Scheme

In this section we show implementation details of a new roof edge detection scheme. It is known that the error signal between a step discontinuity and its smooth approximation shows a zero–crossing at the location of the discontinuity. For a roof edge, the error signal attains its maximum at the location of the discontinuity. If we are interested in concave and convex creases, then we have to consider the absolute value of the error signal. In turn, the first derivative of this residual has a zero–crossing at the location of the discontinuity. So, a zero–crossing detector could indicate either a step discontinuity or a roof discontinuity, provided we feed it the proper information.

Figure 3.8 graphically presents the components of the roof edge detection scheme. Figure 3.8(a) shows an ideal roof edge (solid line) and its smooth approximation (dotted line). Figure 3.8(b), aligned vertically with Fig. 3.8(a), shows the behavior of the error signal (solid line) and the behavior of the derivative of the error signal (dotted line). We can clearly see that the zero–crossings of the derivative occur at the location of the roof discontinuities. Figure 3.9 shows the roof edge detection algorithm by orthogonal scans. The input to the algorithm is

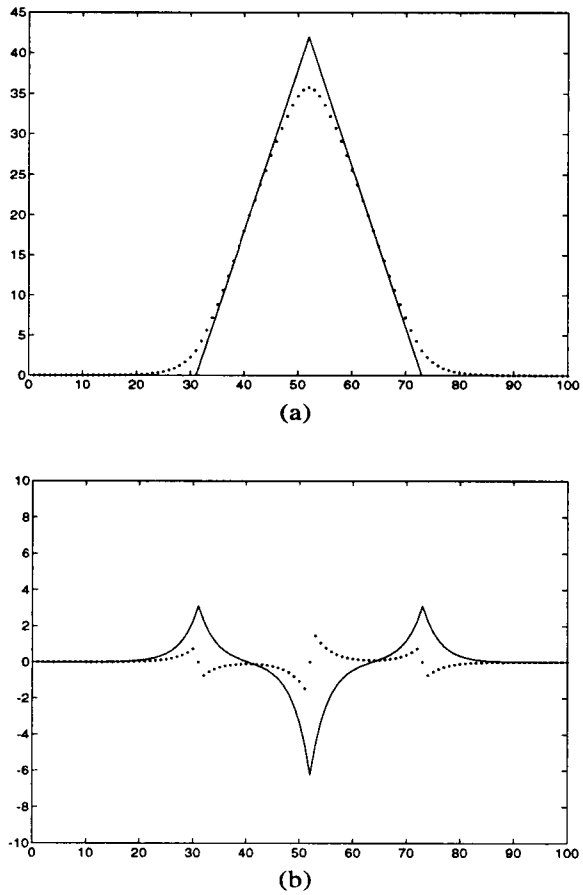


Figure 3.8: *Roof edge detection: (a) roof edge (solid line) and its smooth approximation, (b) error signal (solid) and its derivative.*

a gray–level range image, $r(x, y)$, and the output is a binary image, $B(x, y)$.

Algorithm: *Roof edge detection*
Input: Matrix $R = r(x, y)$ representing range data (gray level). Threshold θ to reject spurious edges.
Output: Binary matrix B representing the edges of the image.

{

Step 1: For each row in R
- find $s(x)$, the 1-D regularized solution to the data,
- find (vertical) zero-crossings for $s(x)$,
- store zero-crossings in buffer B_1.

Step 2: For each row in R^T
- find $s(x)$, the 1-D regularized solution to the data,
- find (vertical) zero-crossings for $s(x)$,
- store zero-crossings in buffer B_2.

Step 3: Perform transpose on output of Step 2.
- let $B_2 = B_2^T$.

Step 4: (Combine results of Steps 1 and 3)
- $B = B_1 \cup B_2$.

}

Figure 3.9: *Algorithm for roof edge detection by orthogonal scans.*

CHAPTER 4

EXPERIMENTAL RESULTS

In this chapter we present experimental results obtained by applying the regularized fusion functionals described in Chapter 3 to a variety of early vision problems. We begin in Section 4.1 with results for the one-dimensional case and different boundary conditions. Next, in Section 4.2 we examine the responses of the various one-dimensional functionals under the same boundary conditions. Next, in Section 4.3 we consider the case of gray–level edge detection, with results provided for both first–order and second–order stabilizers. Next, in Section 4.4 we give the surface reconstruction results obtained employing first– and second–order stabilizers. In Section 4.5, color edge detection results are provided. Finally, in Section 4.6 we show the roof edge detection capabilities of a new roof edge detection algorithm. For computer implementation, we have employed finite differences for solving both first– and second–order stabilizer problems. All our programs have been implemented in the C–language, running under Unix and X–windows on a Sun Sparcstation 10.

4.1 Regularization Results for the One–Dimensional Case for Various Boundary Conditions

In this section, we will consider the solution $f(x)$ when a step discontinuity is present in the input data. (Both noise–free and noisy data sources are being considered). In particular, for the standard regularization functional with a first–order stabilizer, we have implemented the solution for four reasonable choices of boundary conditions (B.C.):

- B.C. 1: $y'(A) = y'(B) = 0$, the so called "natural" boundary conditions,
- B.C. 2: $y(A) = y(B) = 0$, Dirichlet's boundary conditions,
- B.C. 3: $y(A) = y'(A) = 0$, just initial conditions,
- B.C. 4: $y(A) = f1(A),\ y(B) = f1(B)$.

We obtained results for six different cases:

1. noise–free step and B.C. 1,
2. noise–free step and B.C. 2,
3. noise–free step and B.C. 3,
4. noise–free step and B.C. 4,
5. noisy step (zero–mean Gaussian noise) step and B.C. 4,
6. noisy step (zero–mean Gaussian noise) and natural boundary conditions.

For the standard regularized edge detection problem, Figs. 4.1 and 4.2 show the solution $f(x)$ when natural conditions—denoted B.C. 1—were imposed. For values of the regularization parameter λ at and less than 0.00023, the response becomes rapidly oscillatory (see Fig. 4.1(c)). As shown in Fig. 4.2(c), for $\lambda = 10$ the response becomes almost flat, and at the limit it would be a sloped line. Figures 4.3 through 4.4 show the solution $f(x)$ when B.C. 2 were imposed. As seen in Fig. 4.3(c) the oscillatory behavior becomes apparent when λ is 0.000219, and again the response is almost flat for $\lambda = 10$. Figure 4.5 shows the solution $f(x)$ for reasonable λ values when B.C. 3 were imposed. In this case the solution is clamped at the left–hand side, and for λ greater than 10 the response will become increasingly negative. In all these figures, a step discontinuity was present at $x = 50$. Figure 4.6 shows Case 4 for small λ's, while Fig. 4.7 shows Case 4 for bigger values of the regularization parameter. Since this case represents a generalization of Case 2, the results are expected to be better, and this was verified experimentally. The following two cases (Cases 5 and 6) illustrate the behavior of the standard regularization algorithm for noisy input data. We should mention here that solving the equations involves numerical integration. We employed the extended trapezoidal rule for 1–D numerical integration. Figure 4.8 shows Case 5 for small values of λ, and Fig. 4.9 shows the same case for larger values of λ. For small λ's, the response approaches the underlying noise–free step very well, but for large λ's the output becomes flat due to too much smoothing. Figure 4.10 shows Case 6 (noisy step and natural B.C.) for small values of λ, while Fig. 4.11 shows

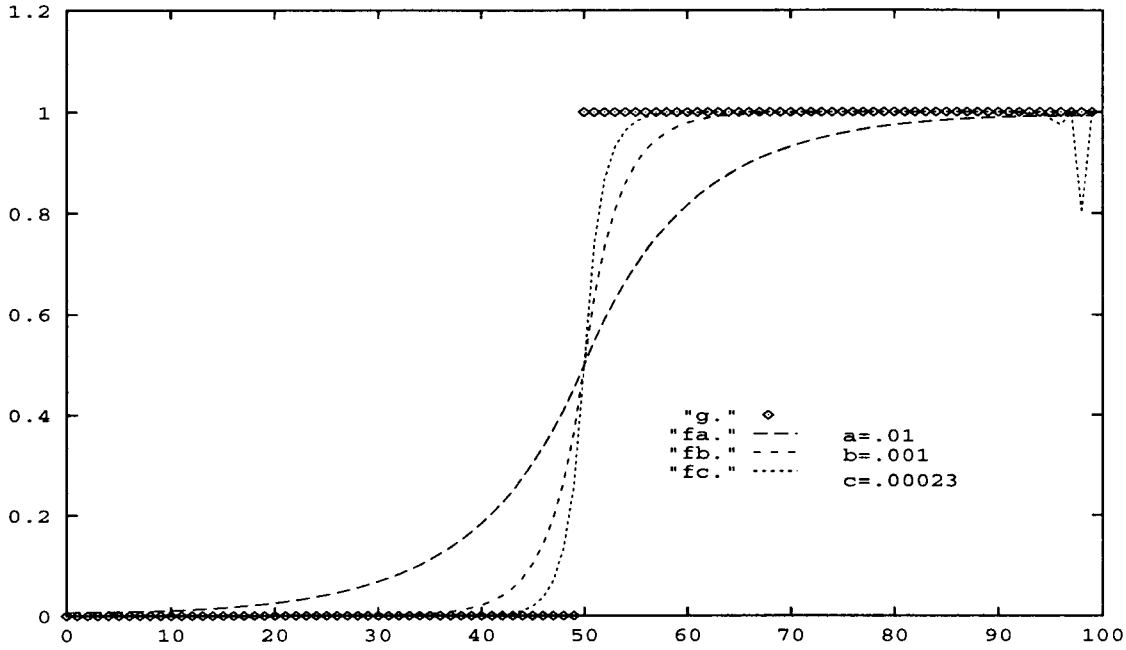


Figure 4.1: *1–D regularization for natural boundary conditions (Case 1) and small* λ.

Case 6 for larger λ values. As we can see from Fig. 4.10(a), the over–smoothing effects start even for small values of λ. Next, we shall illustrate the results of fusing 1–D data employing the first–order stabilizer (FOS) and the second–order stabilizer (SOS). For all these fusion functionals, and in order to keep the number of figures within reason, only two cases were considered: (a) the case in which both $f_1(x)$ and $f_2(x)$ are noise-free, where $f_1(x)$ and $f_2(x)$ represent the input data sources; (b) the case in which both $f_1(x)$ and $f_2(x)$ are contaminated with Gaussian noise. Added clarity can be achieved by dropping $f_1(x)$ and $f_2(x)$ altogether from the plots. In detail, Fig. 4.12 through 4.14 show the noise–free case for the first–order functional. Figures 4.15 through 4.17 show the noisy inputs

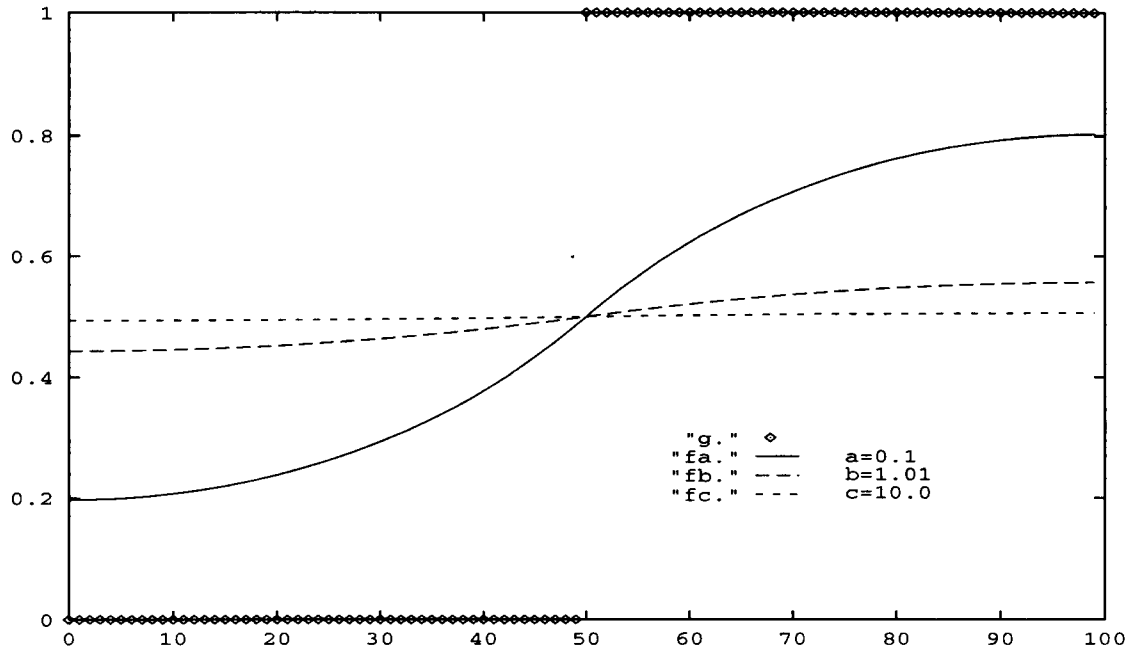


Figure 4.2: *1–D regularization for natural boundary conditions (Case 1), large* λ.

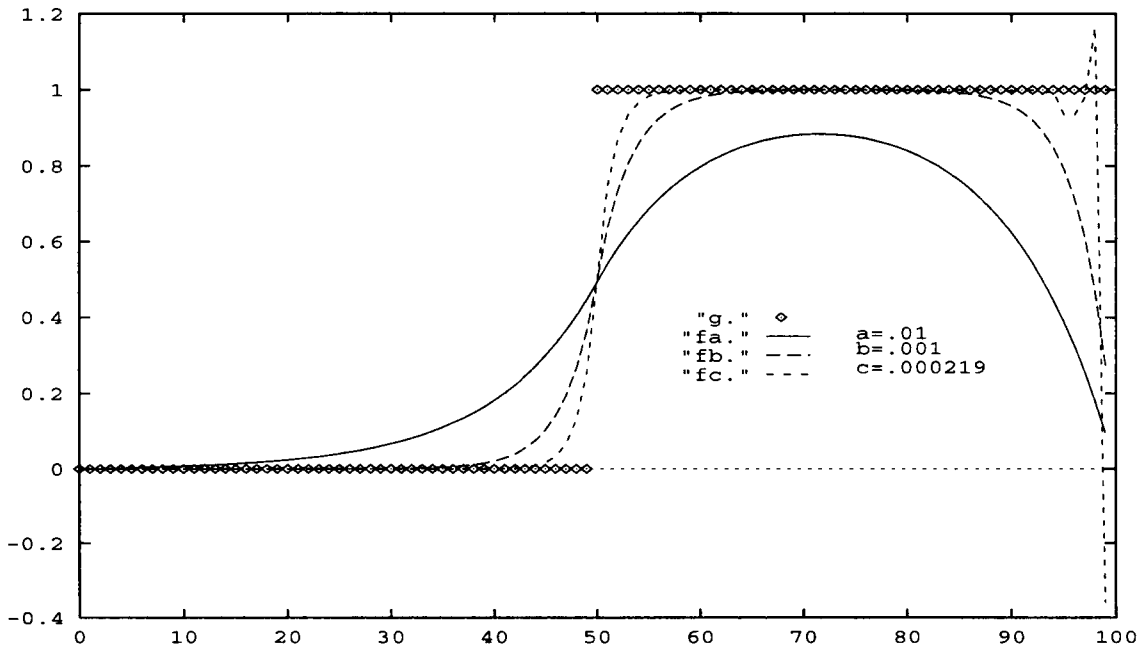


Figure 4.3: *1–D regularization for Dirichlet boundary conditions (Case 2), small* λ.

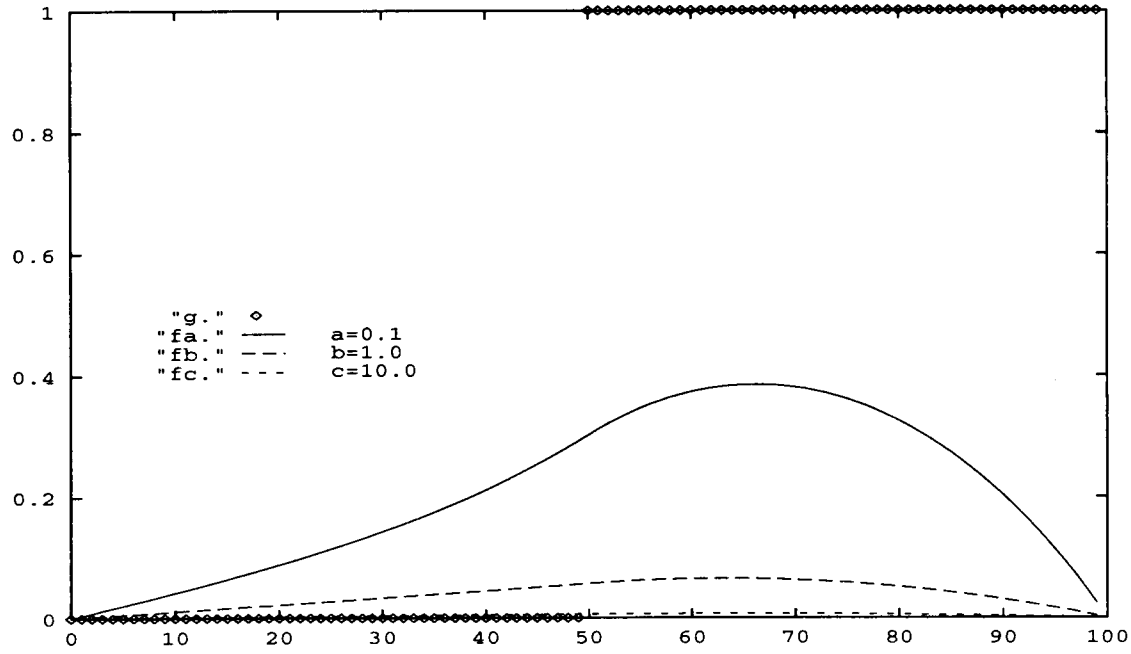


Figure 4.4: *1–D regularization for Dirichlet boundary conditions (Case 2), large* λ.

case for the same functional. Figures 4.18 and 4.19 show the noise–free case for the second–order functional, and Figs. 4.20 and 4.21 illustrate the noisy case for the same (second–order) functional. Figure 4.22 depicts the 1–D technique employing variable regularization parameters. It is evident that good edge detection was attained. In previous figures, we ran the regularization algorithm for a specific value of the regularization parameter; then we modified this parameter and ran the algorithm again, and so forth. Then we collected the various plots in a single figure. By examining the results, an optimal value for λ may be estimated. For Fig. 4.22, we chose instead a maximal value and a minimal value for λ that ensured stability in the response. The algorithm then starts modifying the values of λ from a highly smoothed solution toward a minimally smoothed one,

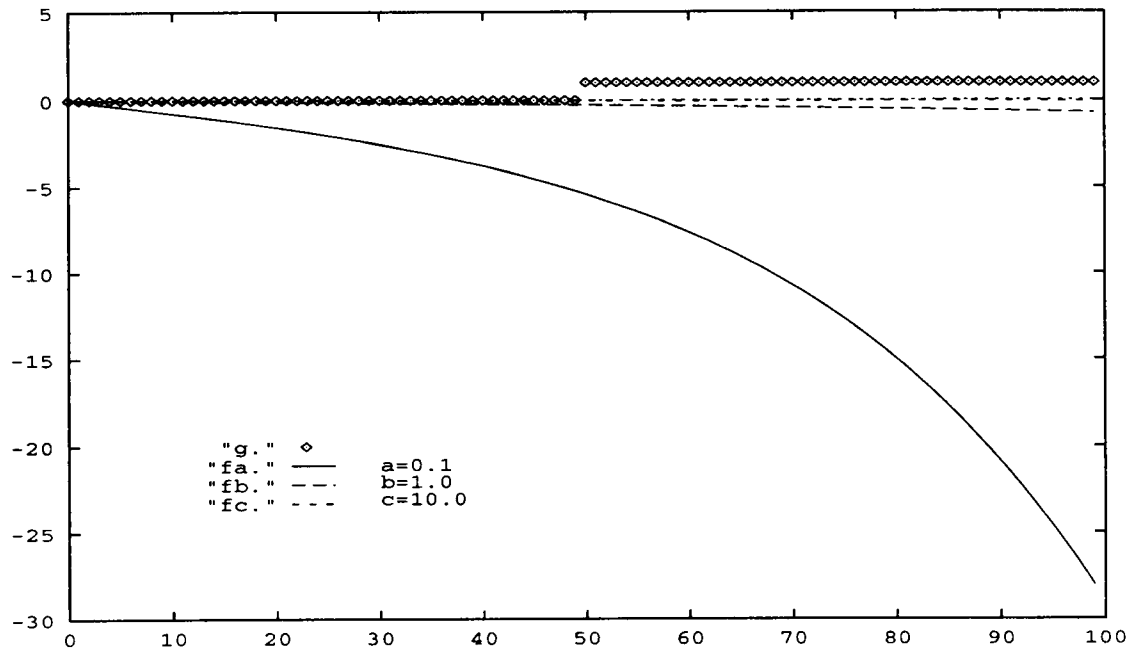


Figure 4.5: *1-D regularization for initial conditions (Case 3), reasonable λ's.*

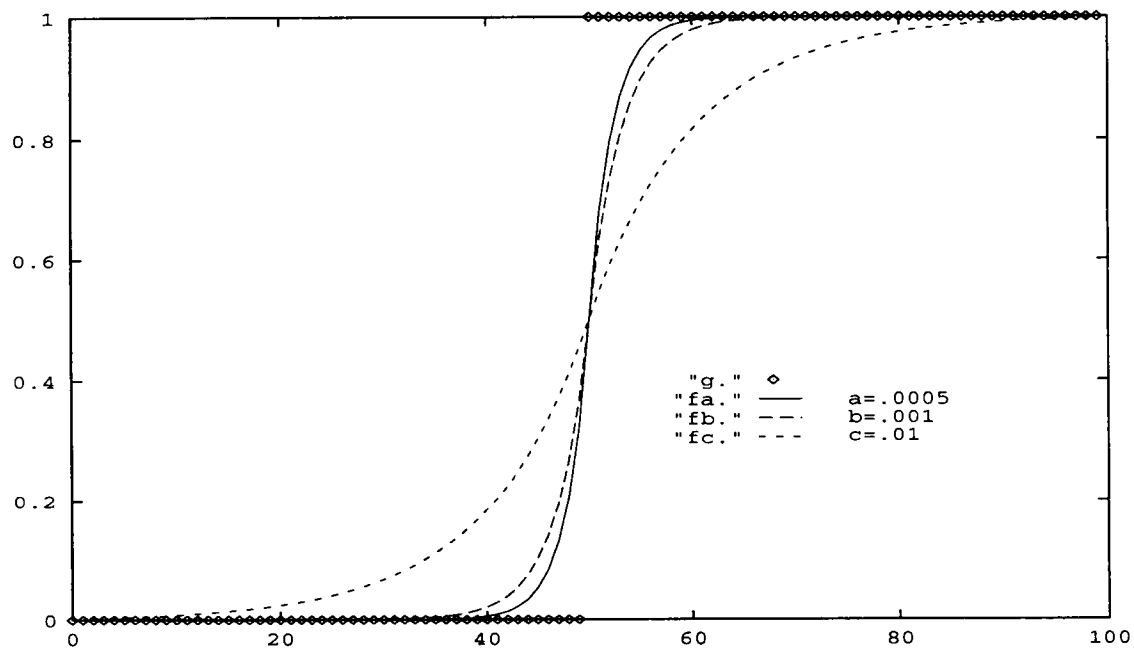


Figure 4.6: *1-D regularization for general boundary conditions (Case 4), small λ.*

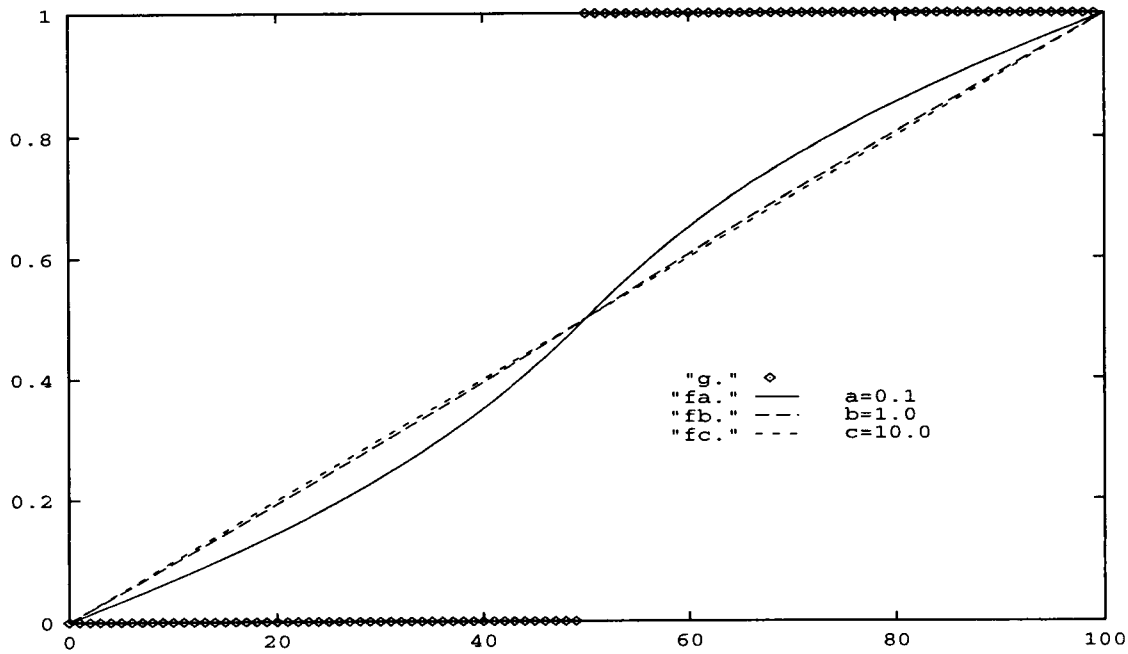


Figure 4.7: *1-D regularization for general boundary conditions (Case 4), large* λ.

according to the slopes in the smoothed response. (Actually, we want to minimize the smoothing near the discontinuities.)

After reviewing the figures, we can evaluate the results as follows:

Case of Standard Regularization:

- The *natural boundary* conditions generated the best results.
- With all other factors the same, certain boundary conditions caused rapid instability of the response.
- At low values of the regularization parameter, we can see the oscillatory response of the system (for B.C. 1).

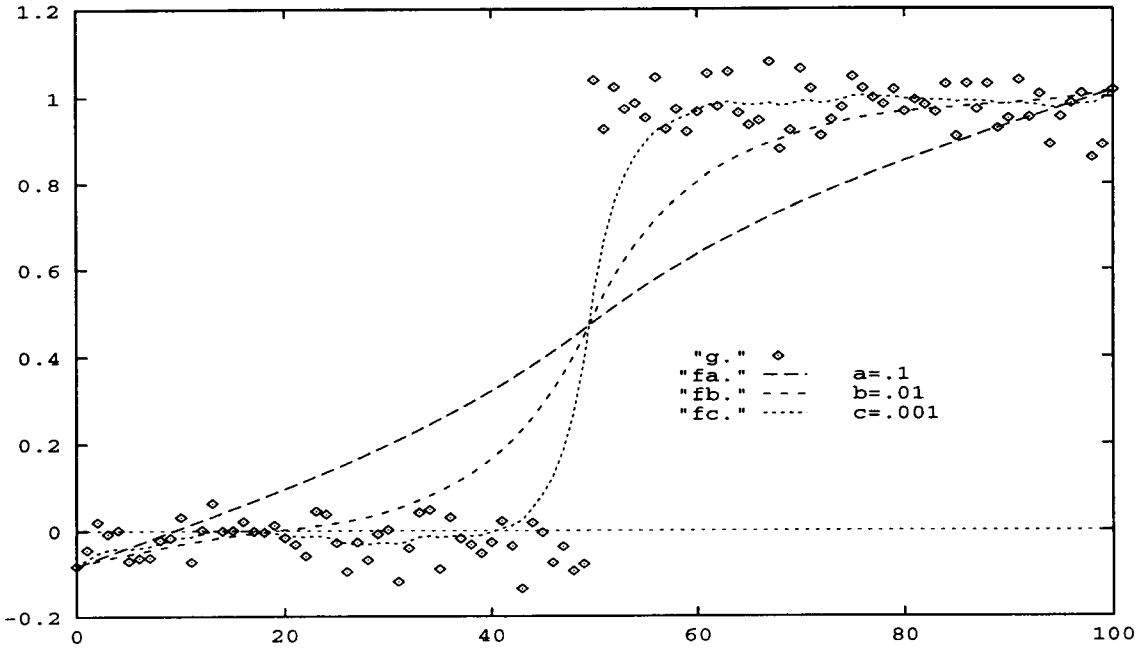


Figure 4.8: *1-D regularization for noisy step and general boundary conditions (Case 5), small λ.*

- At high values of the regularization parameter, the response of the system is maximally flat (for B.C. 1).

Case of First Order Regularized Fusion:

- For low values of λ, undersmoothing may occur for the noisy case.
- For high values of λ, oversmoothing may be a problem.

Case of Second–Order Regularized Fusion:

- We have a smaller interval of values of λ before instabilities occur. This is due to the fourth–order nature of the corresponding differential equation.

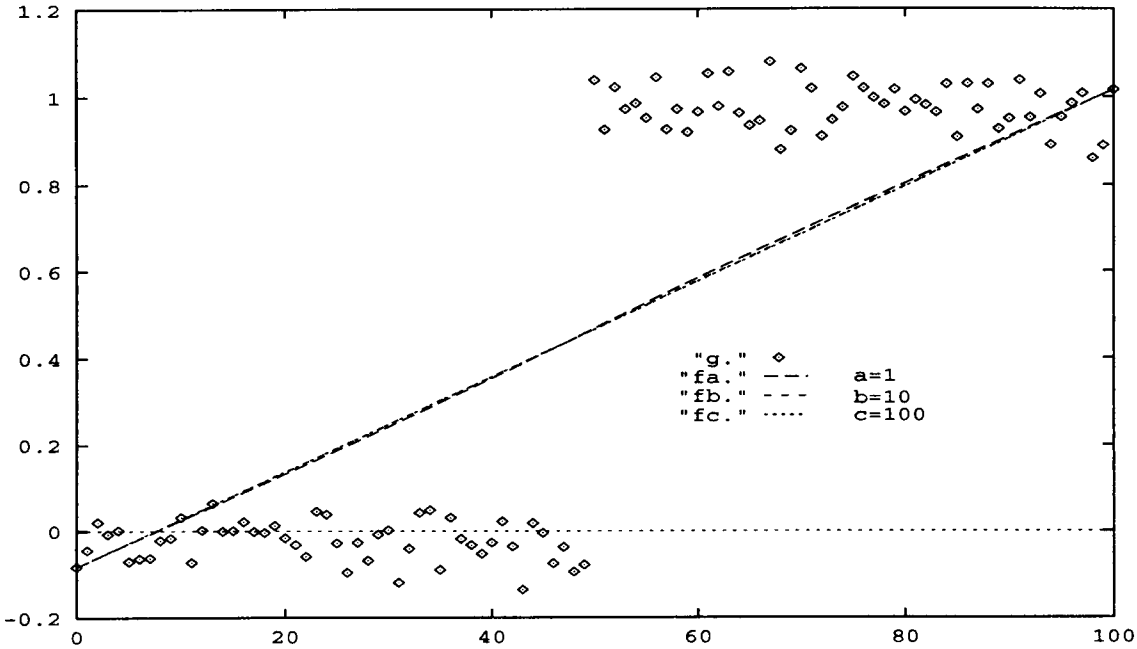


Figure 4.9: *1-D regularization for noisy step and general boundary conditions (Case 5), large λ.*

4.2 Regularization Results for the Different One–Dimensional Functionals

We shall now evaluate the responses of the various functionals proposed in the previous chapter (see Section 3.2). For completeness and symmetry we have added two cases that can be derived from the functionals presented in Section 3.2. These two cases include the second–order stabilizer with fixed regularization parameter and the hybrid (or mixed) functional with fixed regularization parameters.

For all the experiments in this section, we have first considered the case in which the input data is noise–free and then the case in which the input data has been contaminated with zero–mean Gaussian noise of standard deviation $\sigma_n = 2$.

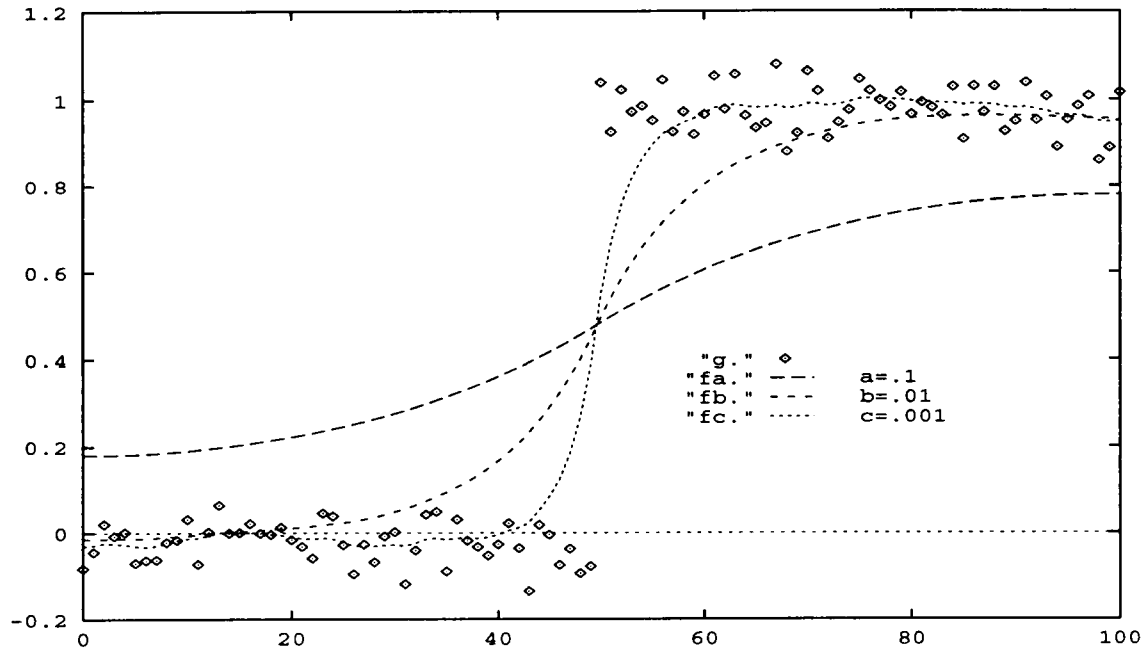


Figure 4.10: *1-D regularization for noisy step and natural boundary conditions (Case 6), small λ.*

For each particular functional, a number of experiments, or subcases, were performed. The results include typical results and/or results for low, medium, and high values of the regularization parameters.

Case 1: First Functional

$$\int (f_1(x) - f(x))^2 \, dx + \lambda^2 \int (f'(x))^2 \, dx$$

We studied the behavior of this functional for three subcases: low, medium, and high values of the regularization parameter. Figure 4.23 shows the responses for the noise-free case, while Fig. 4.24 shows the responses under noisy conditions.

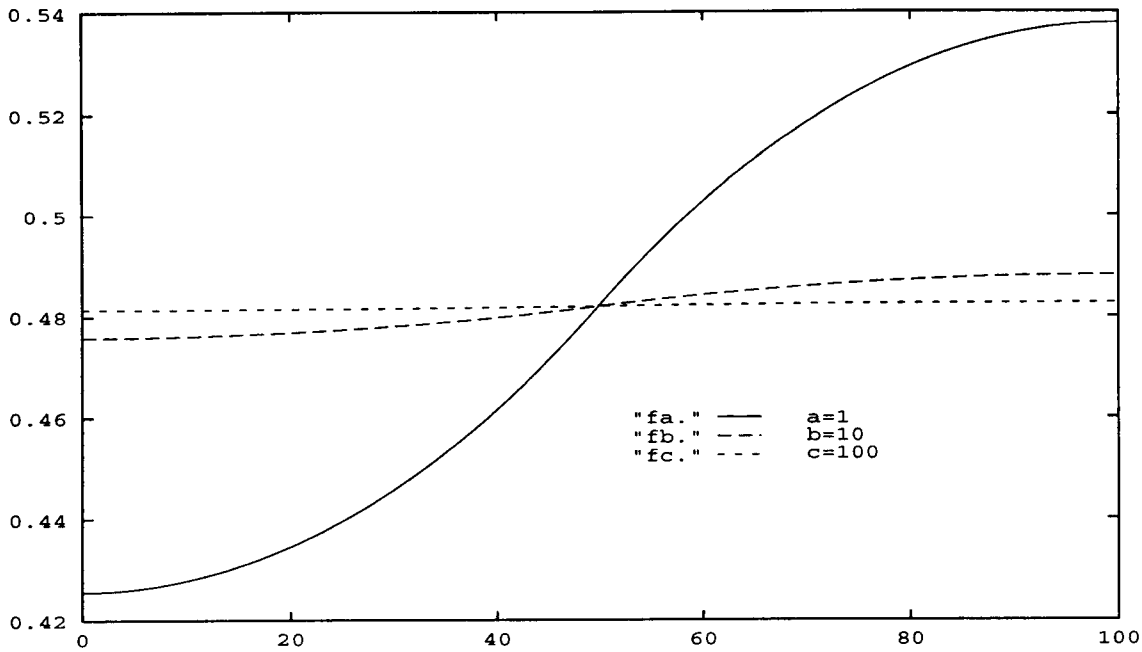


Figure 4.11: *1–D regularization for noisy step and natural boundary conditions, (Case 6), large λ.*

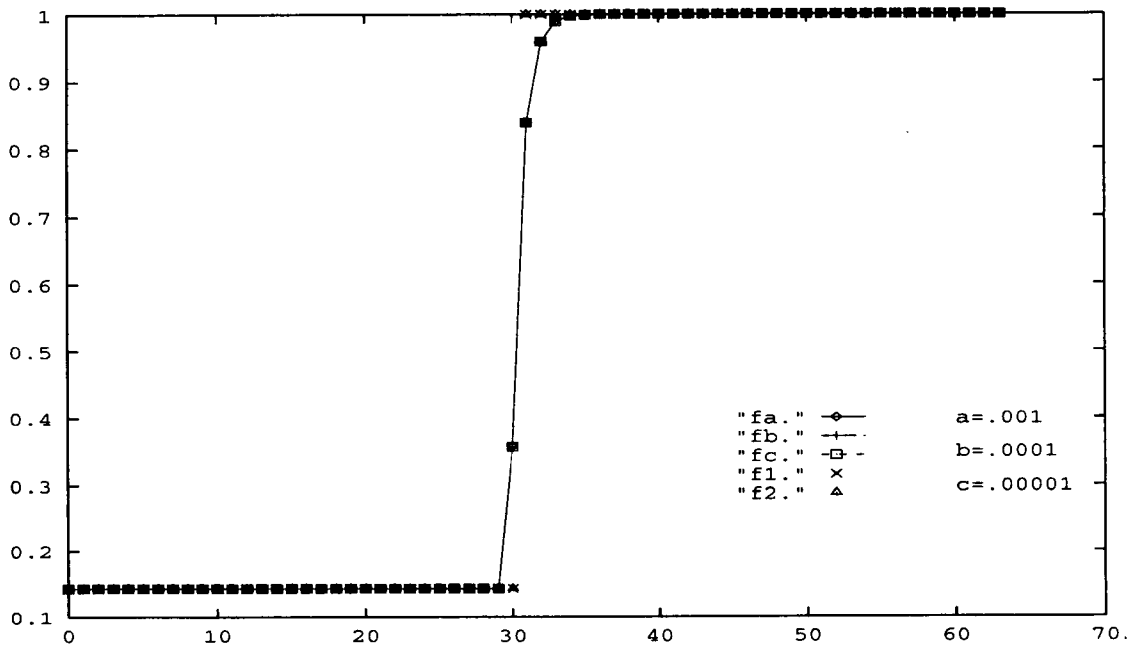


Figure 4.12: *1–D fusion with first–order stabilizer and small λ.*

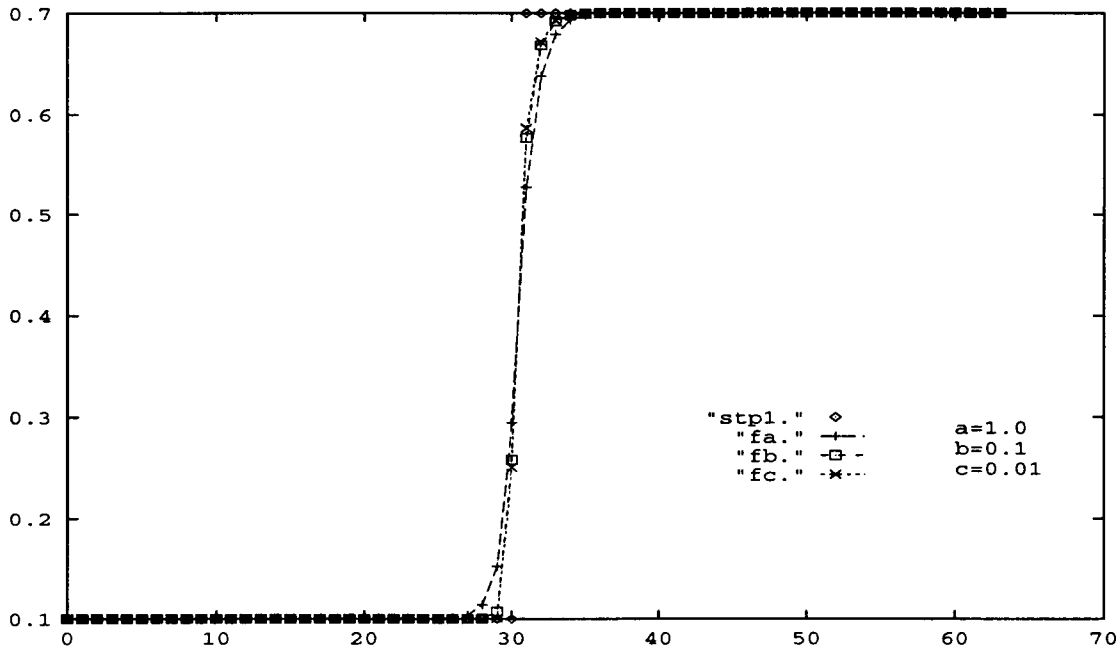


Figure 4.13: *1-D fusion with first-order stabilizer and medium λ.*

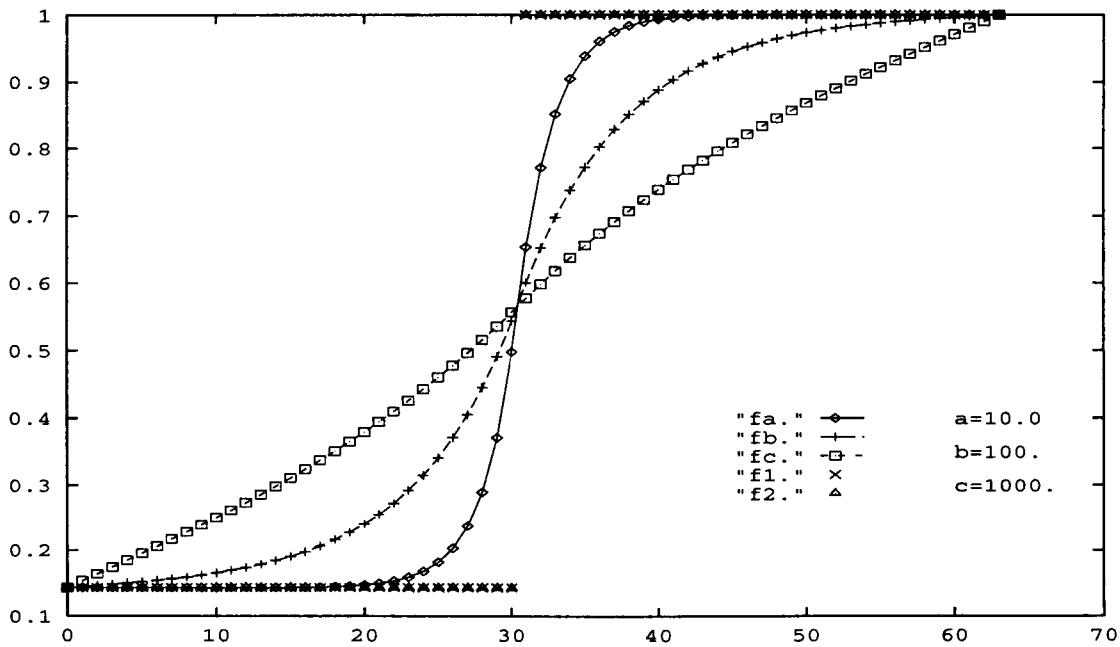


Figure 4.14: *1-D fusion with first-order stabilizer and large λ.*

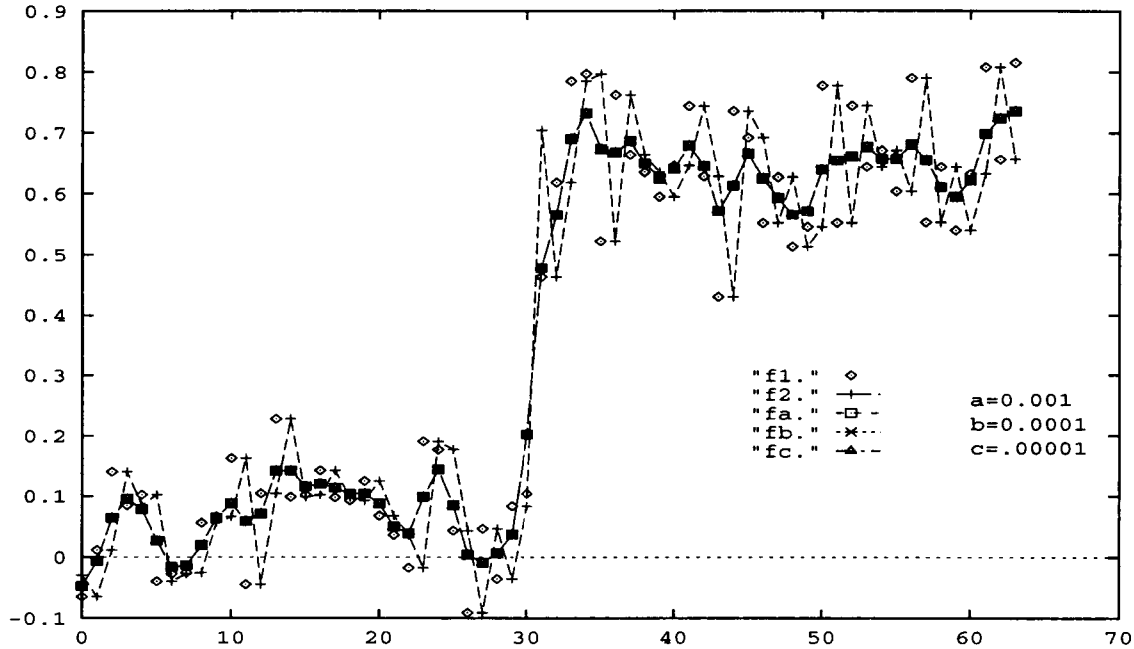


Figure 4.15: *1–D fusion with first–order stabilizer, noisy data, and small* λ.

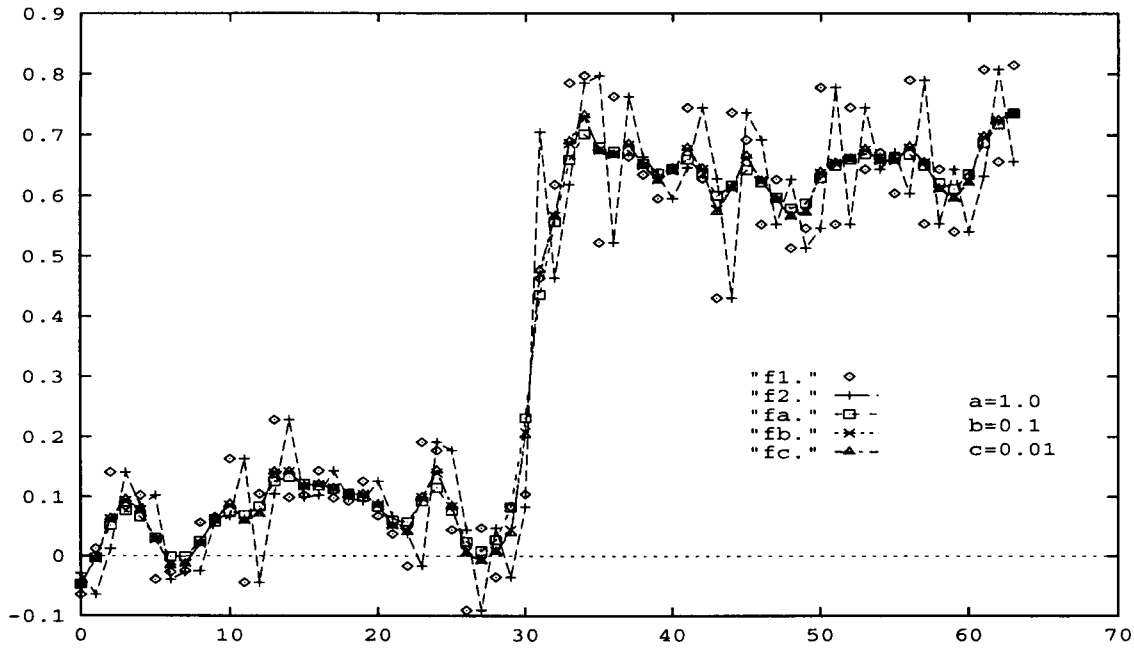


Figure 4.16: *1–D fusion with first–order stabilizer, noisy data, and medium* λ.

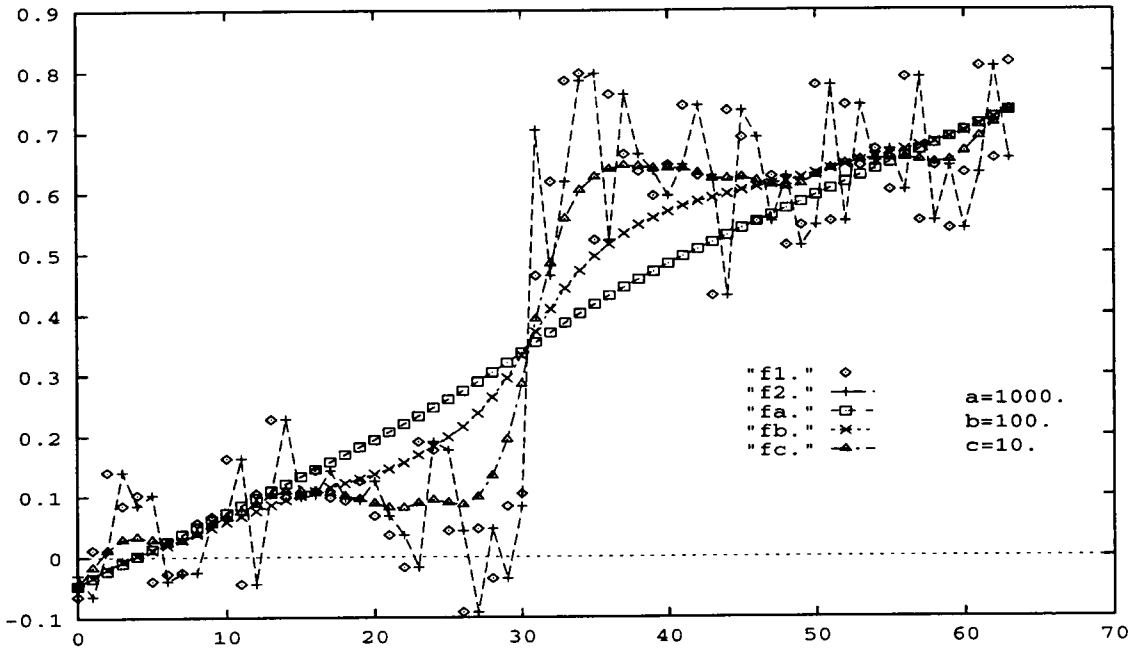


Figure 4.17: *1–D fusion with first–order stabilizer, noisy data, and large* λ.

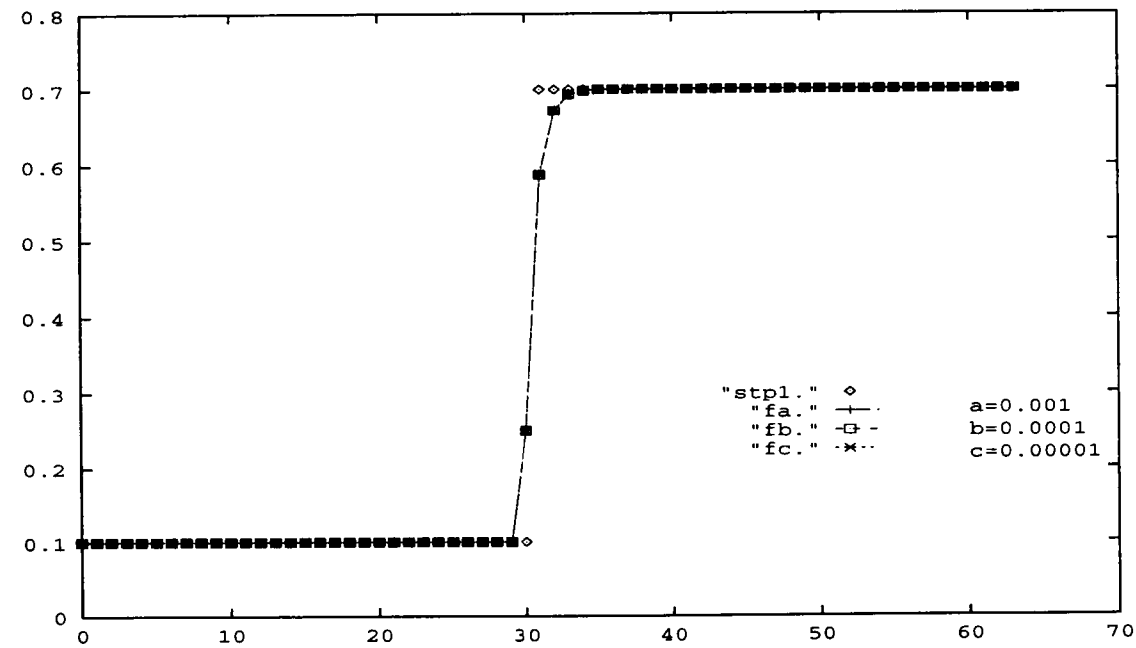


Figure 4.18: *1–D fusion with second–order stabilizer and small* λ.

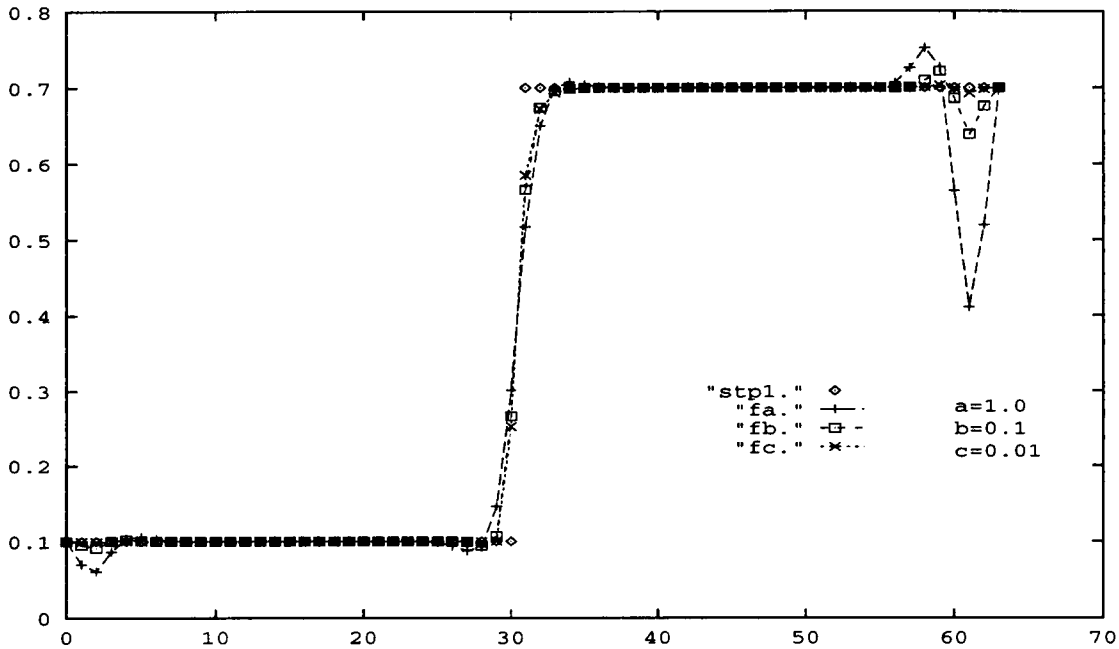


Figure 4.19: *1-D fusion with second-order stabilizer and large λ.*

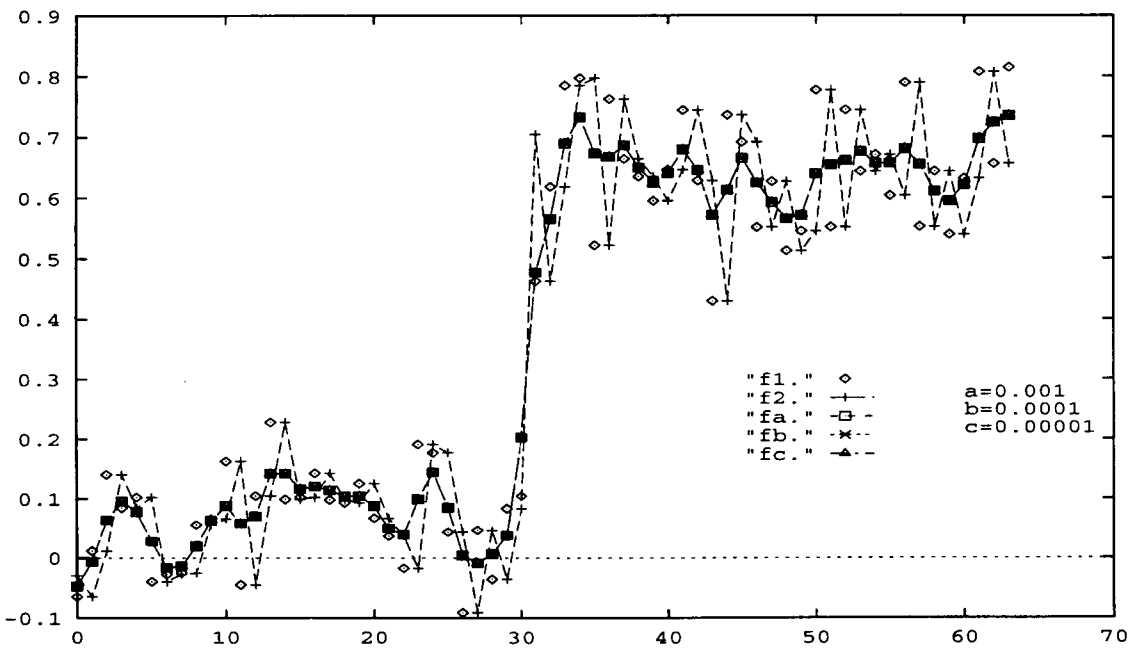


Figure 4.20: *1-D fusion with second-order stabilizer, noisy data, and small λ.*

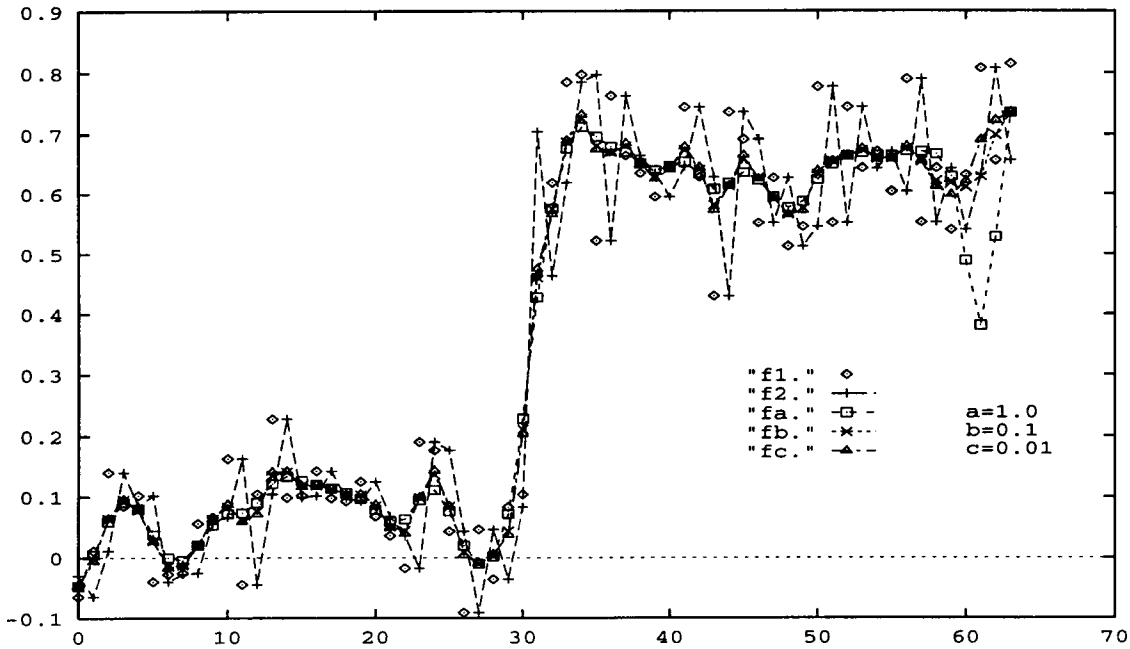


Figure 4.21: *1–D fusion with second–order stabilizer, noisy data, and big* λ.

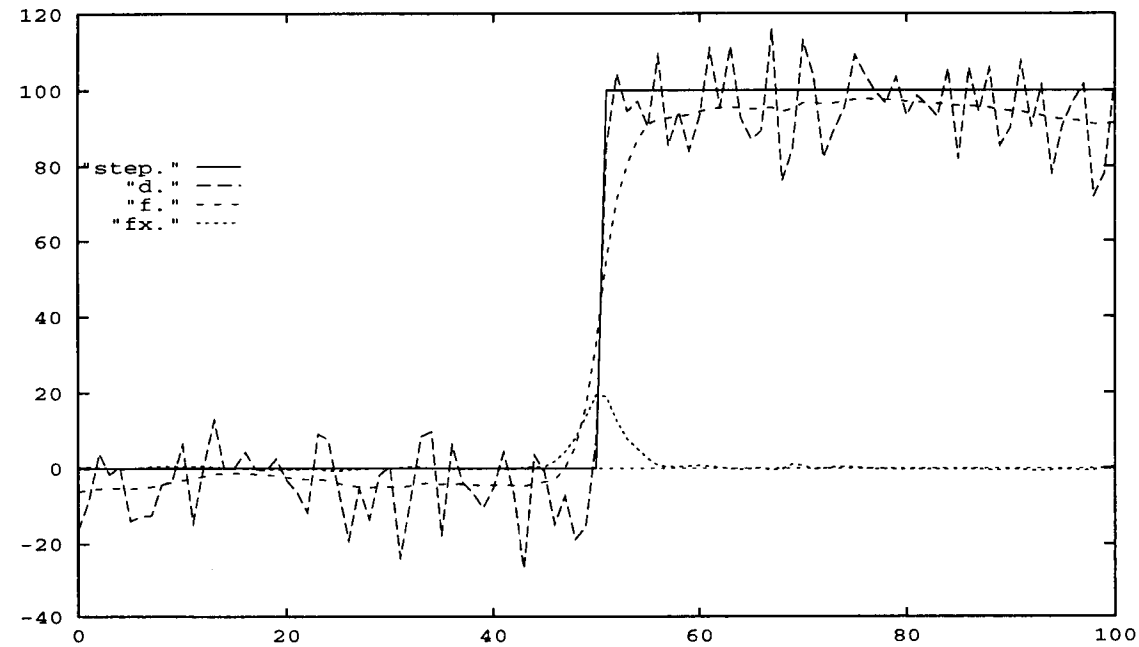
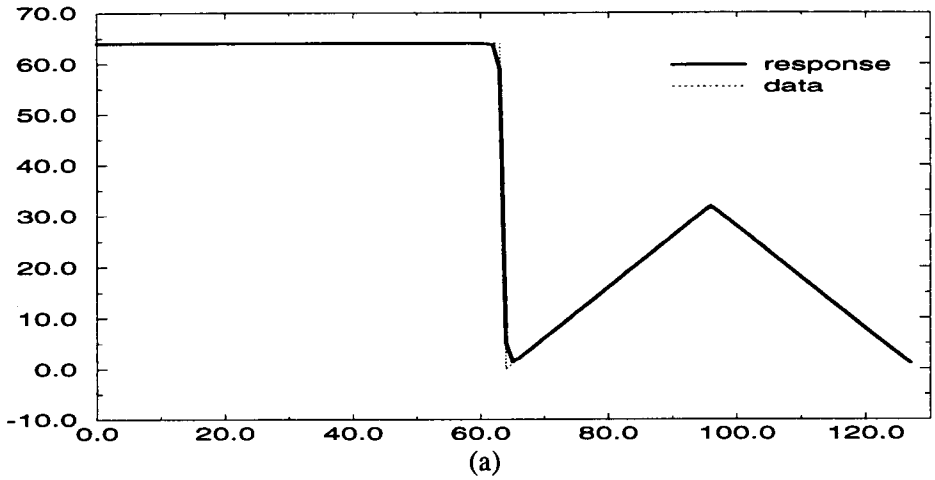
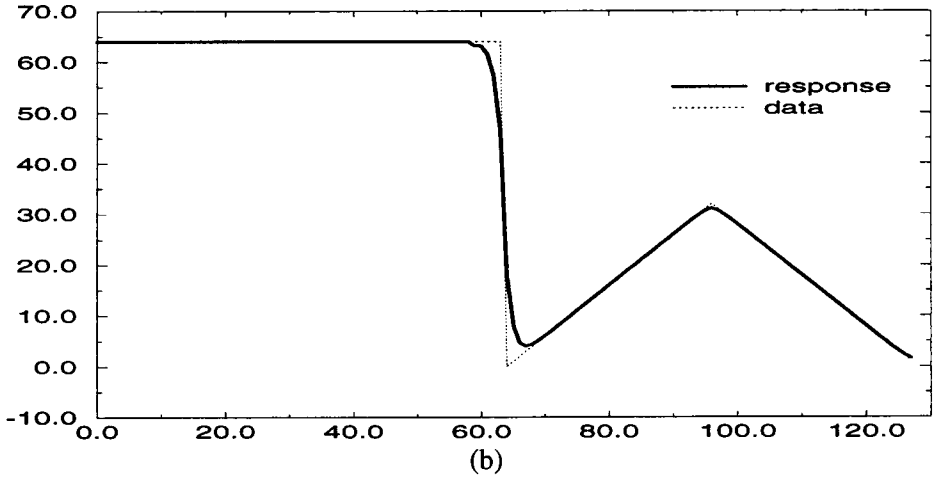


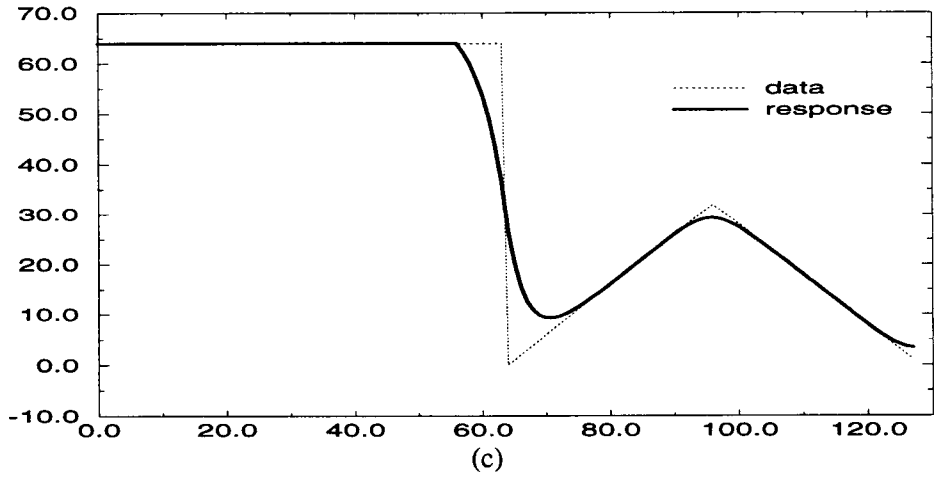
Figure 4.22: *One–dimensional regularization with varying regularization parameters. The input data is labeled "d", the regularized solution is "f", and the derivative of the solution is "fx".*



(a)

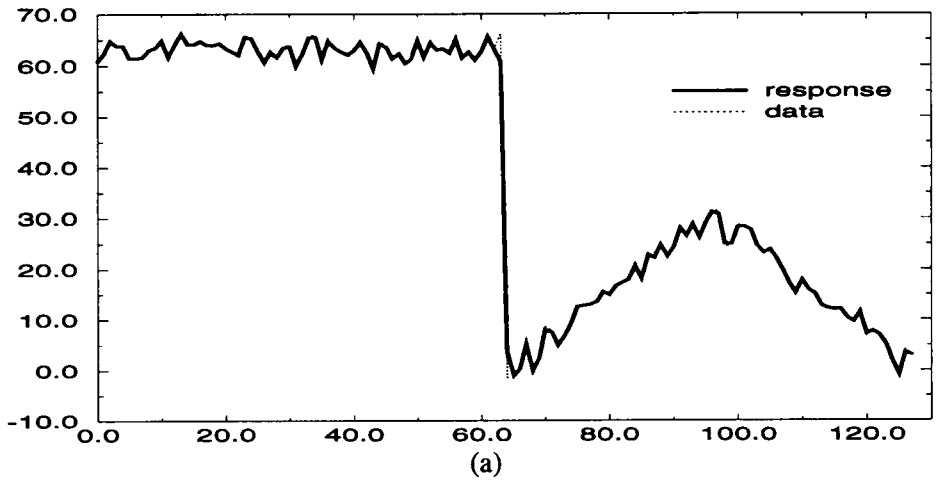


(b)

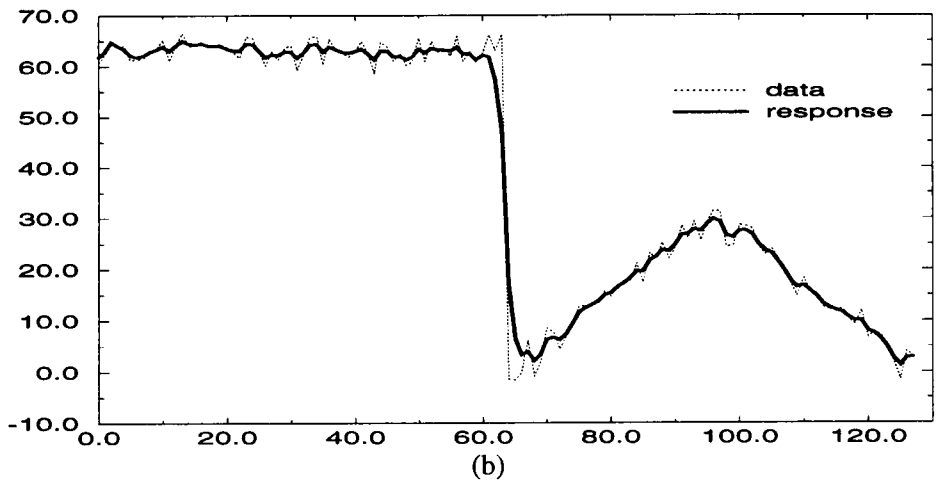


(c)

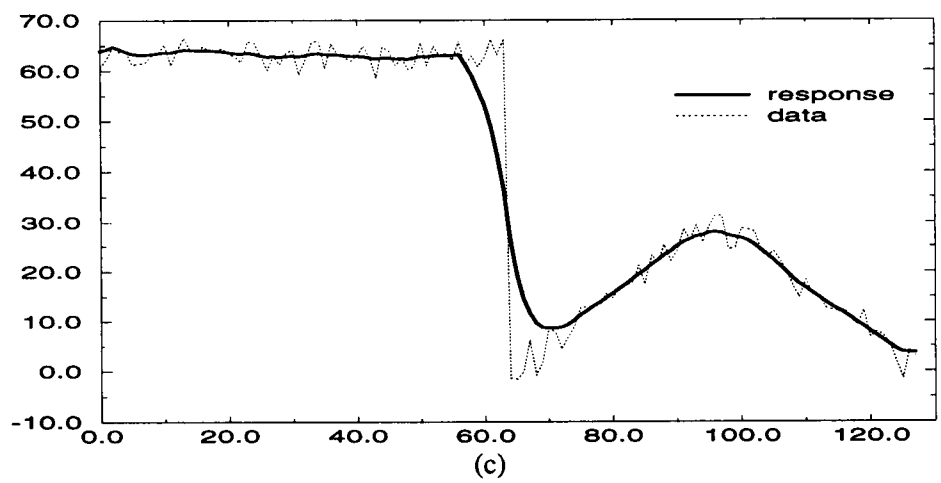
Figure 4.23: *Responses of the first functional (first-order stabilizer with fixed regularization parameter) and noise-free data: (a)* $\lambda = 0.1$, *(b)* $\lambda = 1.0$, *(c)* $\lambda = 10.0$.



(a)



(b)



(c)

Figure 4.24: *Responses of the first functional and noisy data: (a)* $\lambda = 0.1$, *(b)* $\lambda = 1.0$, *(c)* $\lambda = 10.0$.

Case 2: Second Functional

$$\int (f_1(x) - f(x))^2 \, dx + \int \lambda(x)(f'(x))^2 \, dx$$

For this functional, we show typical results for the two basic experiments: noise–free conditions and noisy conditions. Figures 4.25(a) and 4.25(b) show typical results for the noise–free and the noisy data source, respectively.

Case 3: Special Form of the Third Functional

$$\int (f_1(x) - f(x))^2 \, dx + \beta^2 \int (f''(x))^2 \, dx$$

We have added this case to our set of experiments in order to directly compare the responses of the first–order system against the responses of the second–order system for the chosen, or discrete, set of values of the regularization parameter. Again, we have included three subcases: low, medium, and high values of the regularization parameter. Figure 4.26 shows the responses for the noise–free conditions, while Fig. 4.27 shows the responses under noisy conditions.

Case 4: Third Functional

$$\int (f_1(x) - f(x))^2 \, dx + \int [\beta(x)(f''(x))^2] \, dx$$

As in Case 2, we have performed two basic experiments: response under noise–free conditions and response under noisy conditions. Figures 4.28(a) and 4.28(b) show typical results for the noise–free and noisy conditions, respectively.

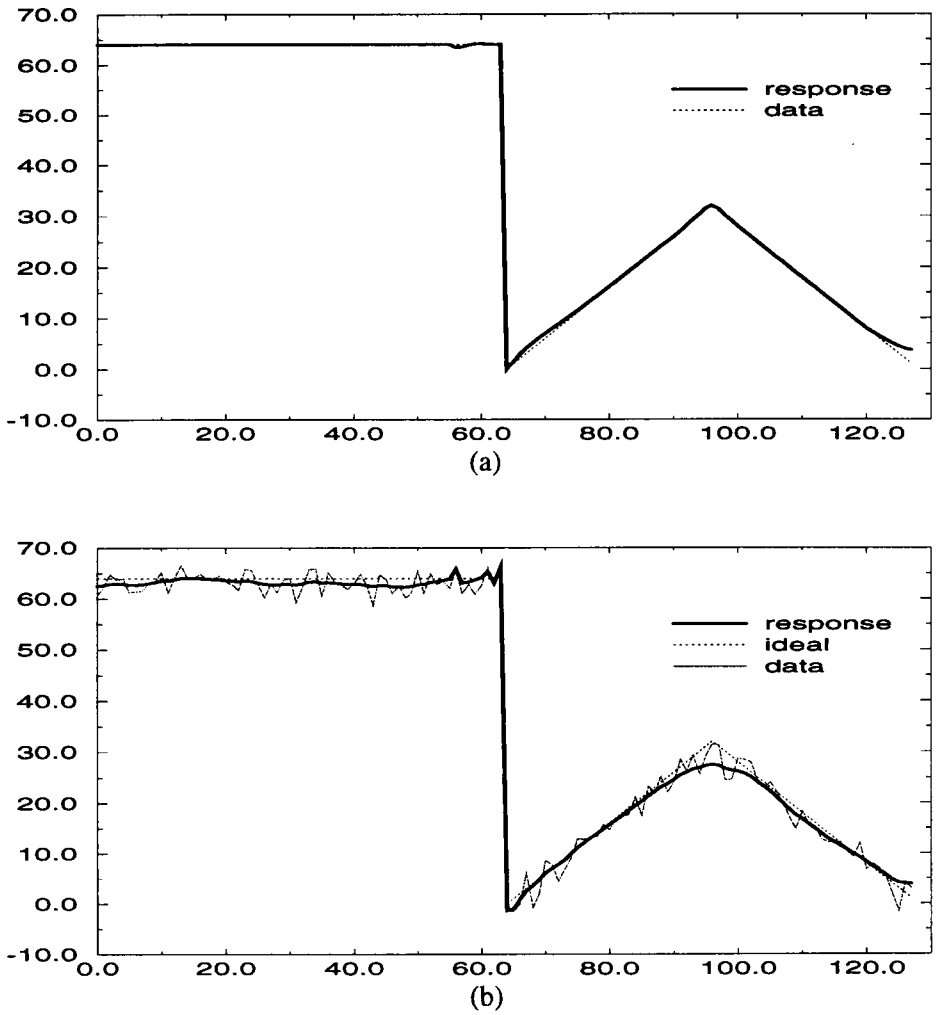


Figure 4.25: *Responses of the second functional (first-order stabilizer and variable regularization parameter $\lambda(x)$): (a) noise-free response, (b) response under noisy conditions.*

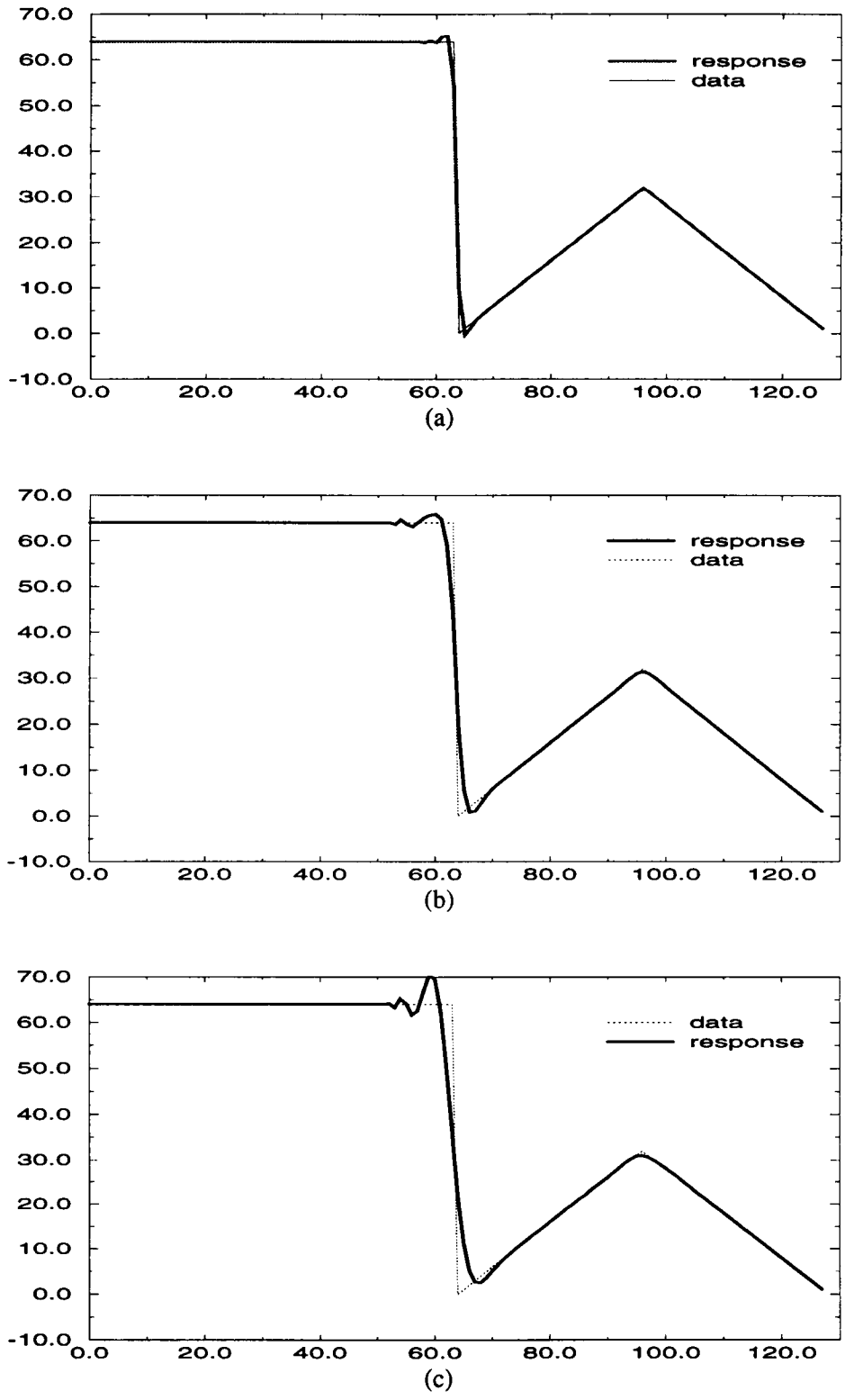


Figure 4.26: *Responses of the second–order stabilizer with fixed regularization parameter under noise–free conditions: (a)* $\beta = 0.1$, *(b)* $\beta = 1.0$, *(c)* $\beta = 10.0$.

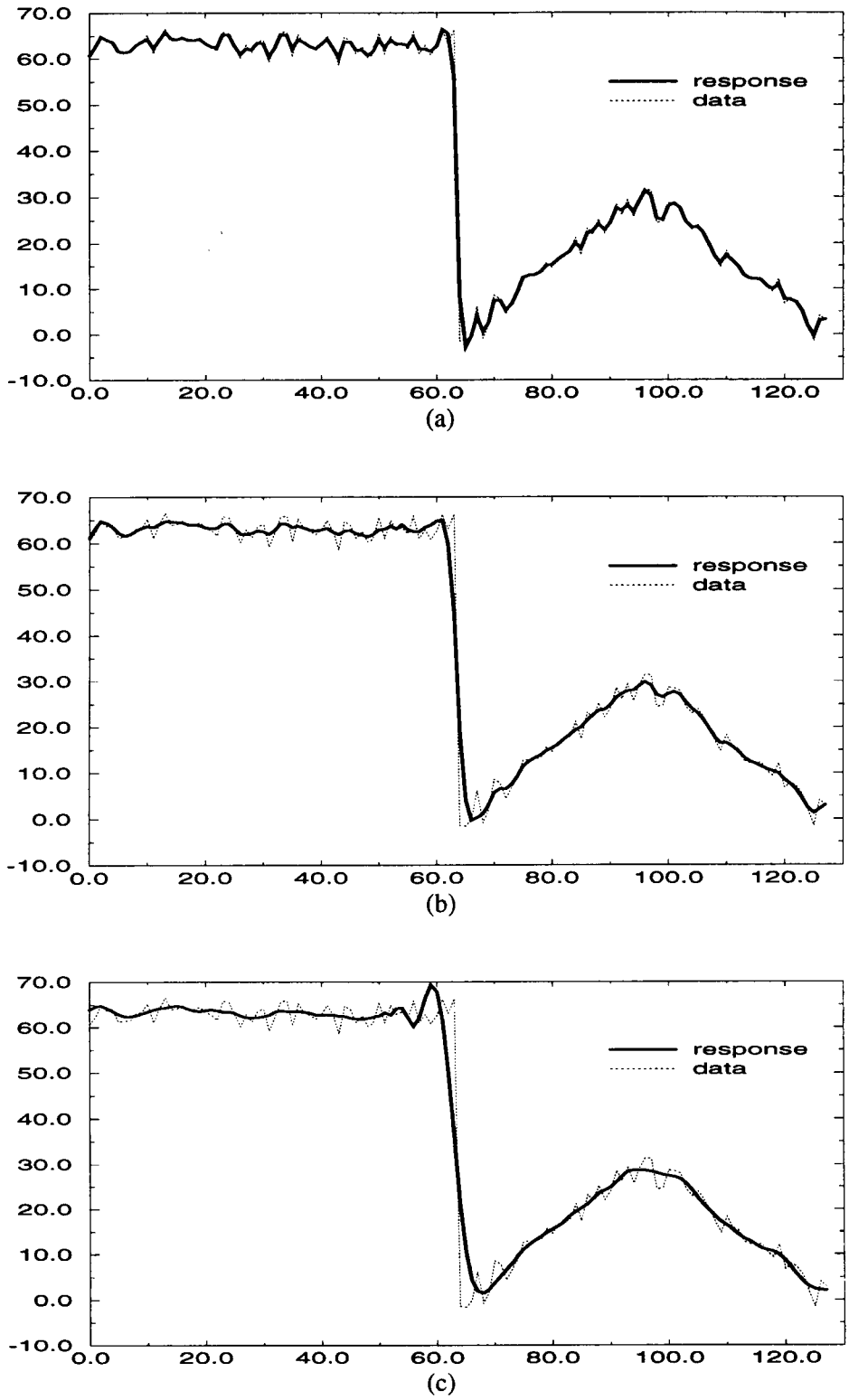


Figure 4.27: *Responses of the second–order stabilizer with fixed regularization parameter under noisy conditions: (a)* $\beta = 0.1$, *(b)* $\beta = 1.0$, *(c)* $\beta = 10.0$.

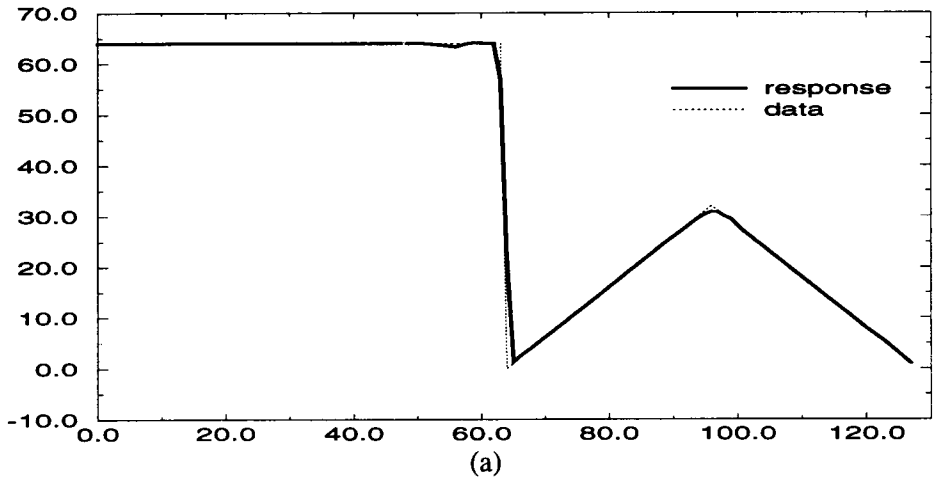


Figure 4.28: *Typical responses of the third functional (second–order stabilizer with variable regularization parameter): (a) noise–free data, (b) noisy data.*

Case 5: Fourth Functional

$$\int [(f_1(x) - f(x))^2]\, dx + \int [\lambda(x)(f'(x))^2 + \beta(x)(f''(x))^2]\, dx$$

For this hybrid, or mixed, functional we have also performed two basic experiments: response under noise–free conditions and noisy conditions. Figures 4.29(a) and 4.29(b) show typical responses for the the noise–free and noisy situations, respectively.

Case 6: Special Form of the Fourth Functional

$$\int (f_1(x) - f(x))^2\, dx + \lambda^2 \int (f'(x))^2\, dx + \beta^2 \int (f''(x))^2\, dx$$

This hybrid regularization functional was derived from the fourth functional by letting the regularization parameters be fixed through space. We now have a system with two free parameters, so we performed a series of experiments in which λ and β took values from the set $\{0.1, 1.0, 10.0\}$. Figure 4.30 shows the nine possible results for the noise–free case. Figure 4.31 shows the nine results obtained for the noisy conditions.

Case 7: Fifth Functional

$$\int [\omega_1(f_1(x) - f(x))^2 + \omega_2(f_2(x) - f(x))^2]\, dx + \int [\lambda(x)(f'(x))^2 + \beta(x)(f''(x))^2]\, dx$$

For this functional we performed a total of six experiments: three for the noise–free case and three for the noisy case. The basic idea was to observe the effect on

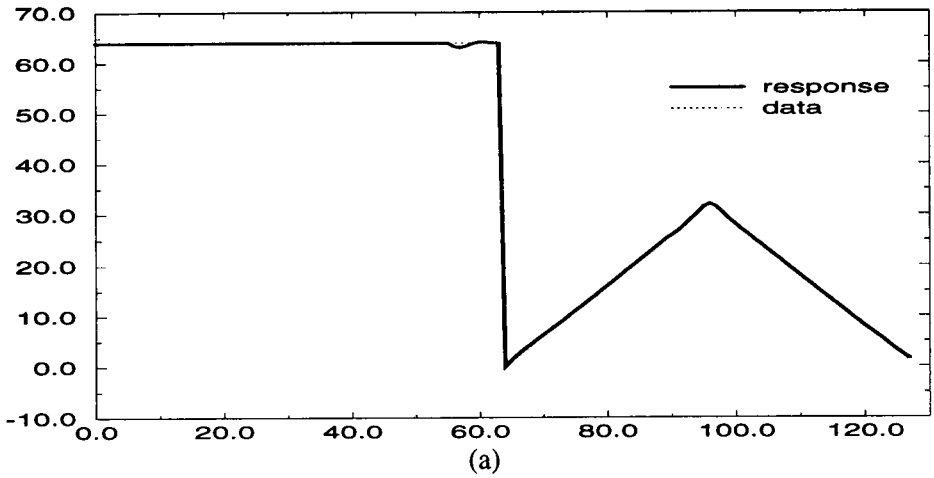


Figure 4.29: *Typical responses of the fourth functional (a hybrid system with first- and second-order stabilizers with variable regularization parameters): (a) noise-free data, (b) noisy data.*

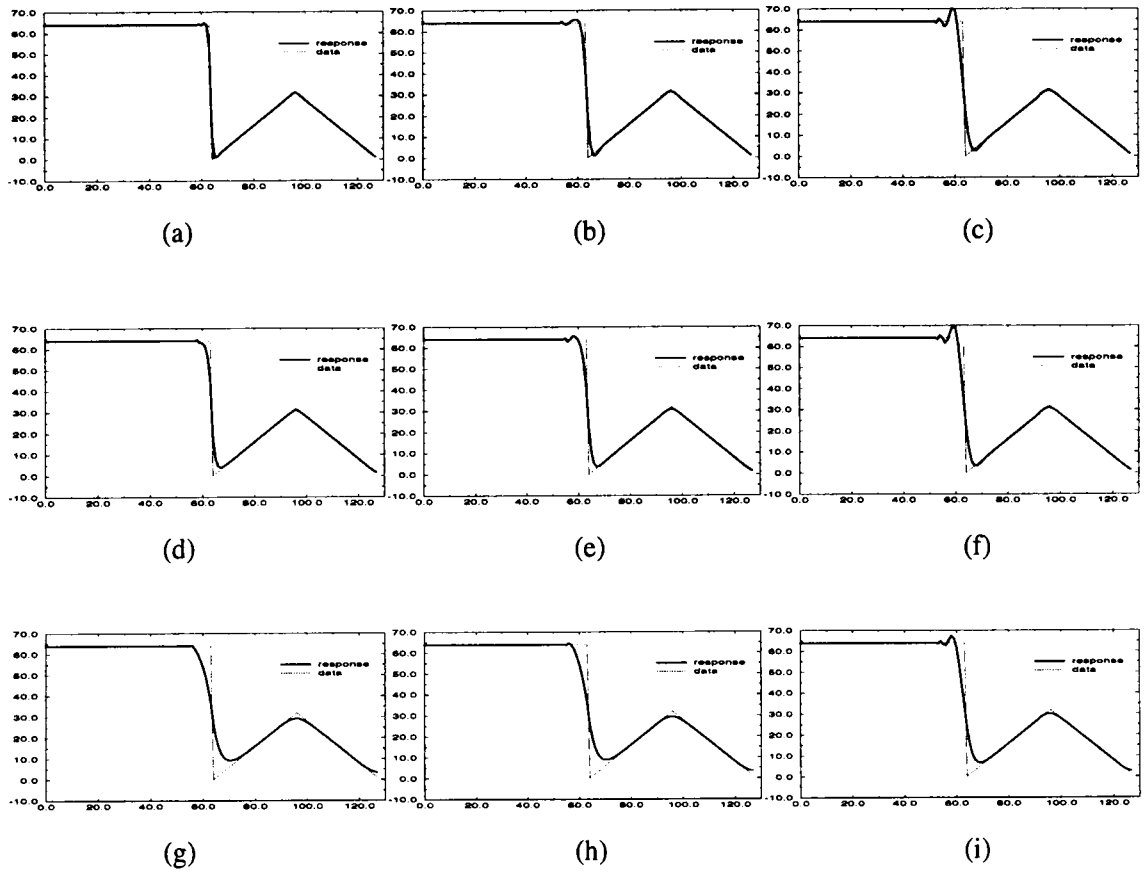


Figure 4.30: *Responses of the hybrid functional (first- and second-order stabilizers with fixed regularization parameters) under noise-free conditions: (a)* $\lambda = \beta = 0.1$, *(b)* $\lambda = 0.1$, $\beta = 1.0$; *(c)* $\lambda = 0.1$, $\beta = 10.0$; *(d)* $\lambda = 1.0$, $\beta = 0.1$; *(d)* $\lambda = \beta = 1.0$, *(f)* $\lambda = 1.0$, $\beta = 10.0$; *(g)* $\lambda = 10.0$, $\beta = 0.1$; *(h)* $\lambda = 10.0$, $\beta = 1.0$; *(i)* $\lambda = \beta = 10.0$.

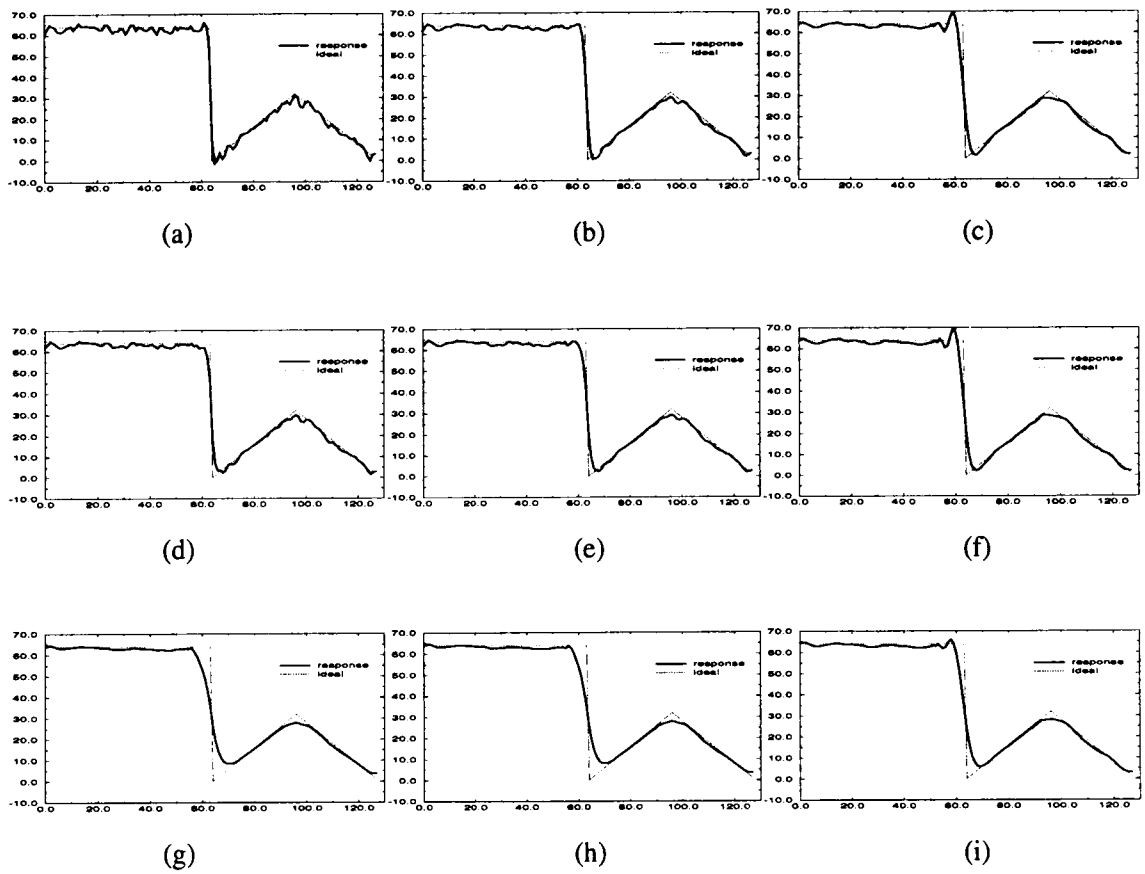


Figure 4.31: *Responses of the hybrid functional (first- and second-order stabilizers with fixed regularization parameters) under noisy conditions: (a)* $\lambda = \beta = 0.1$, *(b)* $\lambda = 0.1$, $\beta = 1.0$; *(c)* $\lambda = 0.1$, $\beta = 10.0$; *(d)* $\lambda = 1.0$, $\beta = 0.1$; *(d)* $\lambda = \beta = 1.0$, *(f)* $\lambda = 1.0$, $\beta = 10.0$; *(g)* $\lambda = 10.0$, $\beta = 0.1$; *(h)* $\lambda = 10.0$, $\beta = 1.0$; *(i)* $\lambda = \beta = 10.0$.

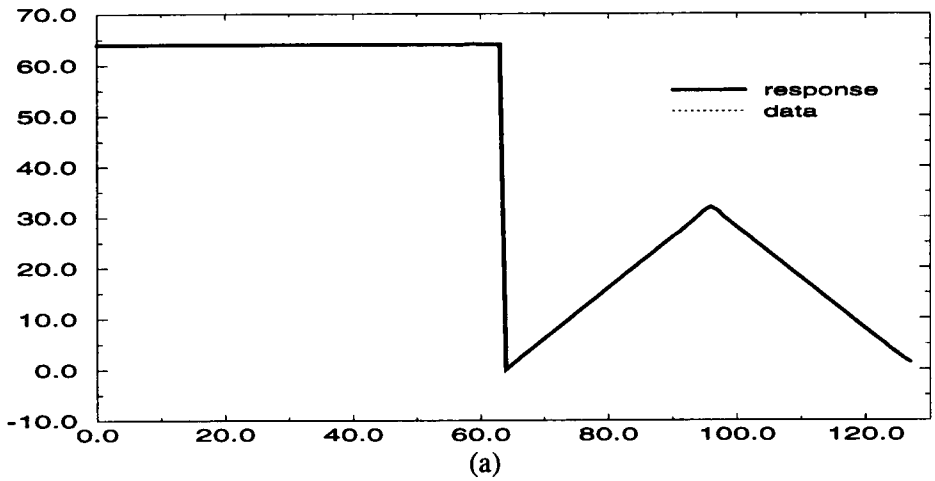
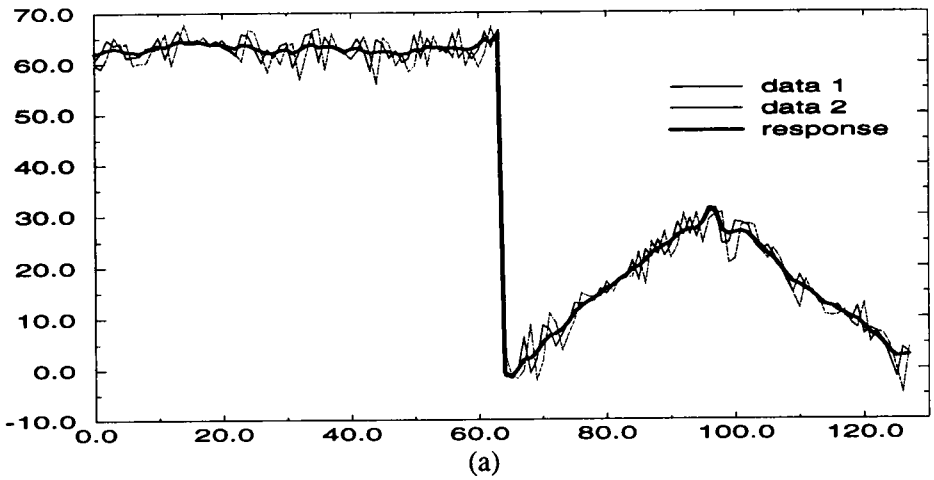


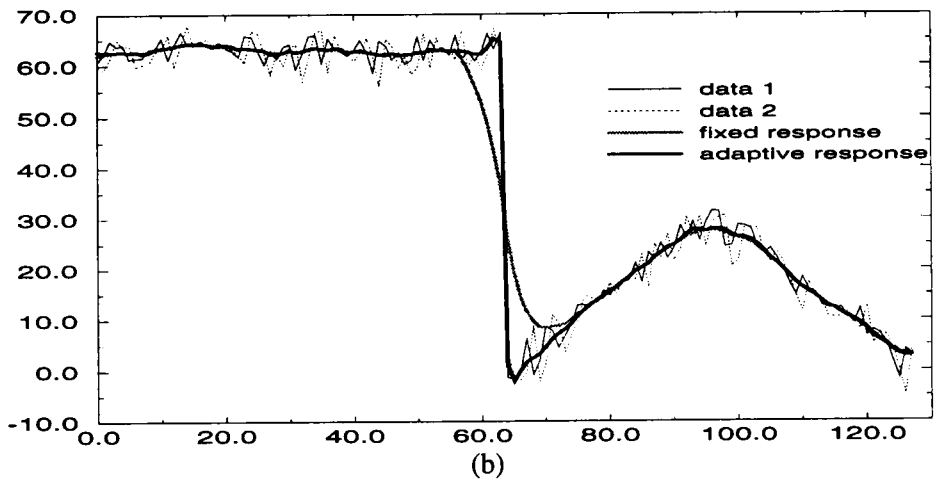
Figure 4.32: *Typical result for the fifth functional (a mixed stabilizer system with two data sources) under noise–free conditions:* $\omega_1 = \omega_2$.

the response due to the weights ω_1 and ω_2. The three subcases considered were $\omega_1 > \omega_2$, $\omega_1 = \omega_2$, and $\omega_1 < \omega_2$.

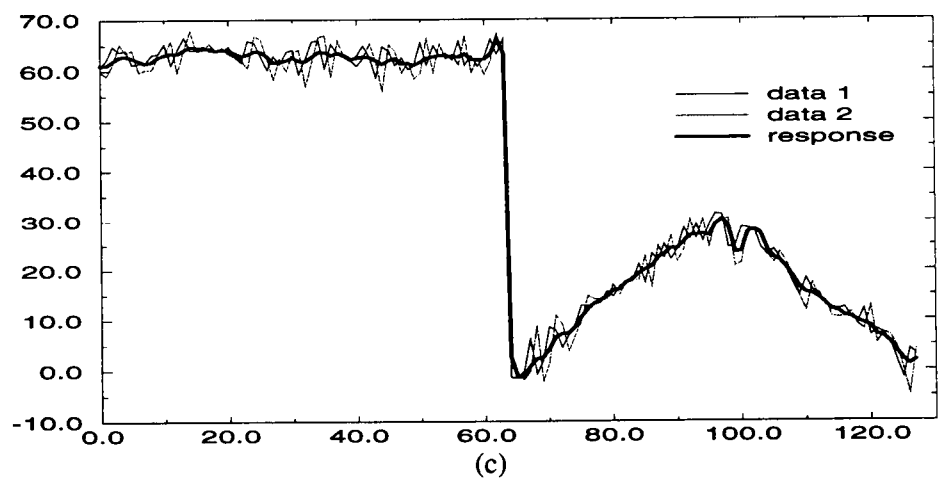
Figure 4.32 shows a typical result for the noise–free case, while Figure 4.33 shows the results for the noisy case. Since the responses for the noise–free case were identical for all practical purposes, we included only one of them. In Fig. 4.33(b) we have also included the response under similar conditions for the nonadaptive system. We can easily see that oversmoothing has occurred at the location of the jump discontinuity. We can appreciate, then, the improved response achieved with adaptive regularization parameters.



(a)



(b)



(c)

Figure 4.33: *Typical results for the fifth functional (a mixed stabilizer system with two data sources) under noisy conditions: (a)* $\omega_1 > \omega_2$, *(b)* $\omega_1 = \omega_2$, *(c)* $\omega_1 < \omega_2$.

4.3 Gray Edge Detection Results

We have developed and tested our 2–D fusion functionals for the first–order and second–order stabilizer cases. We will illustrate typical results obtained when the algorithm was applied first to synthetic data and then to real data. For the case of synthetic images, we have considered first intensity–intensity image pairs, i.e., images of the same modality. Then we considered a synthetic range–intensity image pair. The intensity–only image data has been contaminated with zero–mean Gaussian noise.

Figure 4.34 shows results for the noisy "square" image pair. Figure 4.34.a shows the first input image and Fig. 4.34(b) shows the second input image. The first input image has a signal–to–noise Ratio (SNR) of 13.01, while the second input image has an SNR of 10. The fused edges detected by the zero–crossing approach are shown in Fig. 4.34(c), while Fig. 4.34(d) shows the fused edges employing the gradient technique. Figs. 4.34(e) and 4.34(f) show typical results obtained using a conventional (nonfused) zero–crossing method on each of the input images, respectively. Figures 4.34(g) and 4.34(h) show typical results obtained using a conventional (nonfused) gradient method on each of the input images, respectively. Clearly, both fusion approaches performed better than the conventional edge detectors.

Figures 4.35 through 4.37 show the results for the "cake," "checkers," and "wake" image pairs, respectively. The "cake" image (see Fig. 4.35) illustrates the

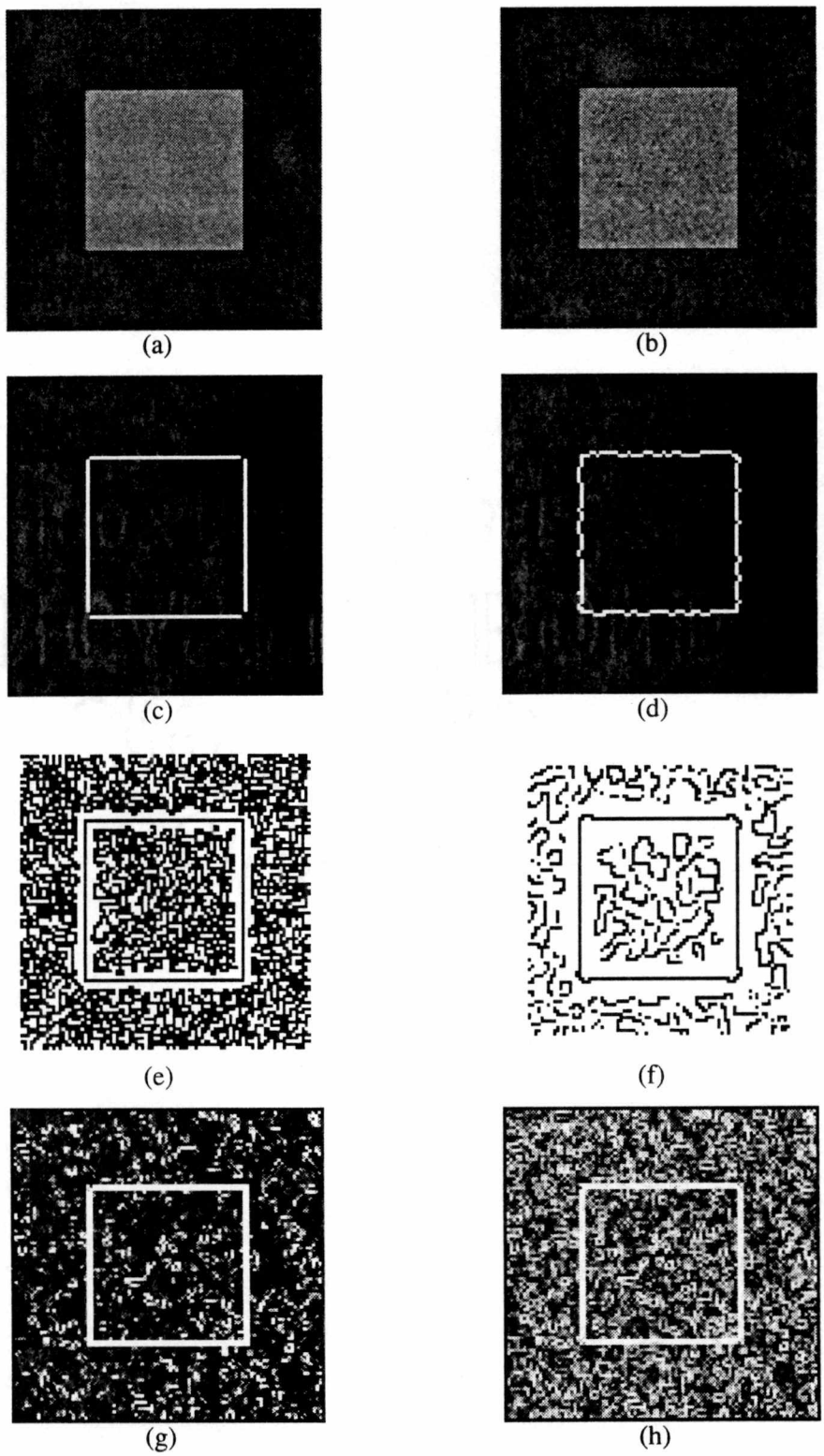


Figure 4.34: *Square; fusion with first–order stabilizer: (a) input 1, $\sigma_n = 5$, (b) input 2, $\sigma_n = 10$, (c) zero–crossing map, (d) gradient edge map, (e) conventional zero–crossing on input 1, (f) conventional zero-crossing on input 2, (g) gradient on input 1, (h) gradient on input 2.*

problems that appear when the SNR is decreased for a given feature. The ideal cake has three regions with intensity levels of 75, 175, and 215 respectively. So the signal–to–noise ratios are 8.24 and 4.26, respectively for the worst case. This effective decrease in the SNR (although the image has been contaminated with noise of a fixed standard deviation) causes the zero–crossing based approach to miss the innermost edge.

The "checkers" image (see Fig. 4.36) generalizes the problem of the square to an array of subproblems. We can appreciate now the problems of proximity of features in both the vertical and horizontal directions. Since the SNR is kept above 9, the performance of the zero–crossing based technique is correct.

The "wake" image (see Fig. 4.37) represents a minimal model of the trail left in the water by a moving ship. This experiment was included to illustrate the capabilities of the method for a task that traditionally has been performed using Hough or Radon transforms. We can clearly see from Fig. 4.37(c) or (d) that the major features have been extracted from the noisy scene. Although the problem of wake detection for real ships from SAR imagery is more complex than this, we do not discard the possibility of employing our method as a front–end processor for these kinds of images as part of a bigger system encompassing knowledge about wake signals.

Figure 4.38 shows results for a synthetic range–intensity image pair when the first–order stabilizer algorithm is applied. This experiment serves two purposes. The first, is to evaluate the behavior of the algorithm for data of different sensing

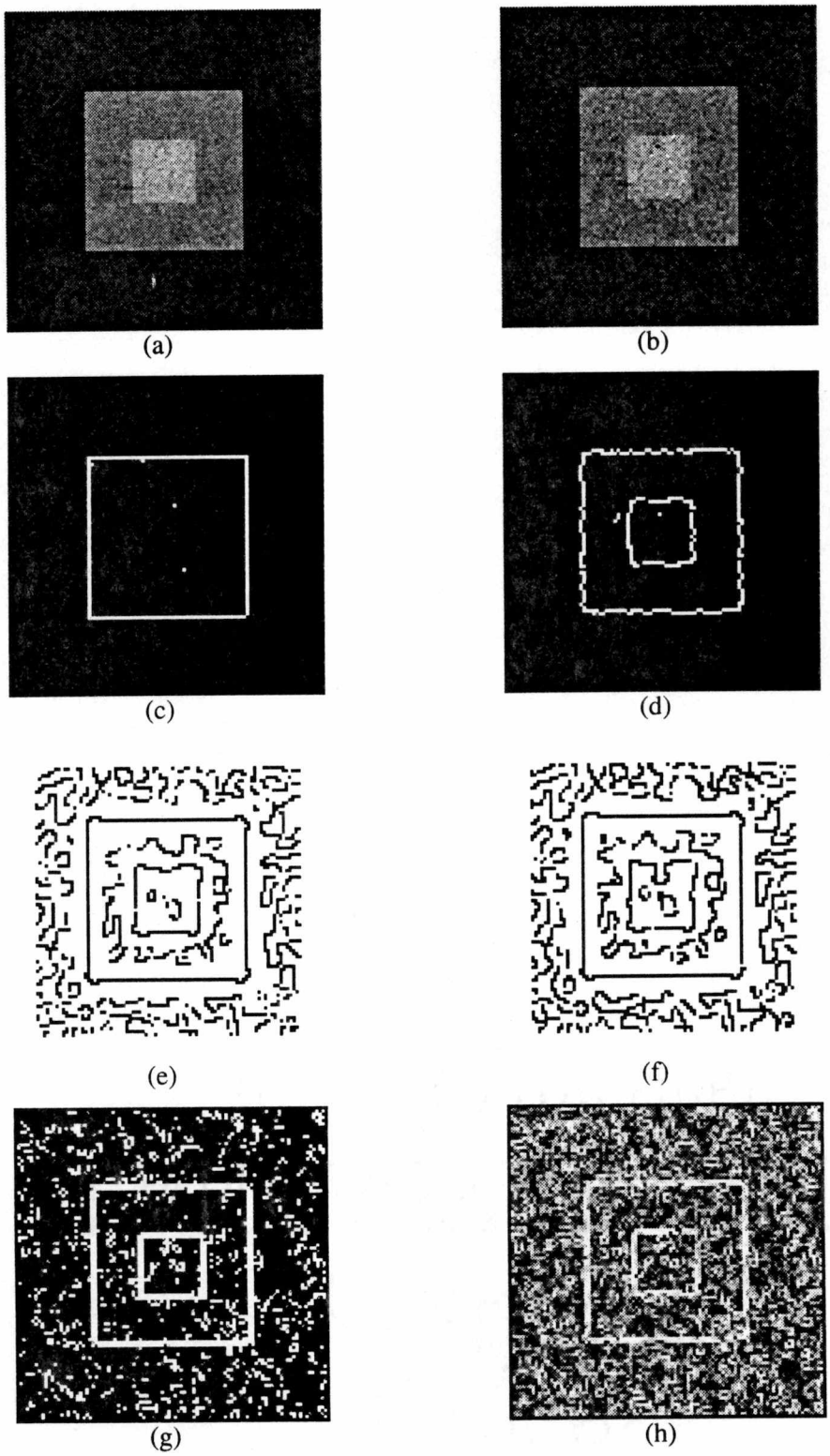


Figure 4.35: *Cake; fusion with first–order stabilizer: (a) input 1,* $\sigma_n = 10$*, (b) input 2,* $\sigma_n = 15$*, (c) zero–crossing map, (d) gradient edge map, (e) conventional zero–crossing on input 1, (f) conventional zero–crossing on input 2, (g) gradient on input 1, (h) gradient on input 2.*

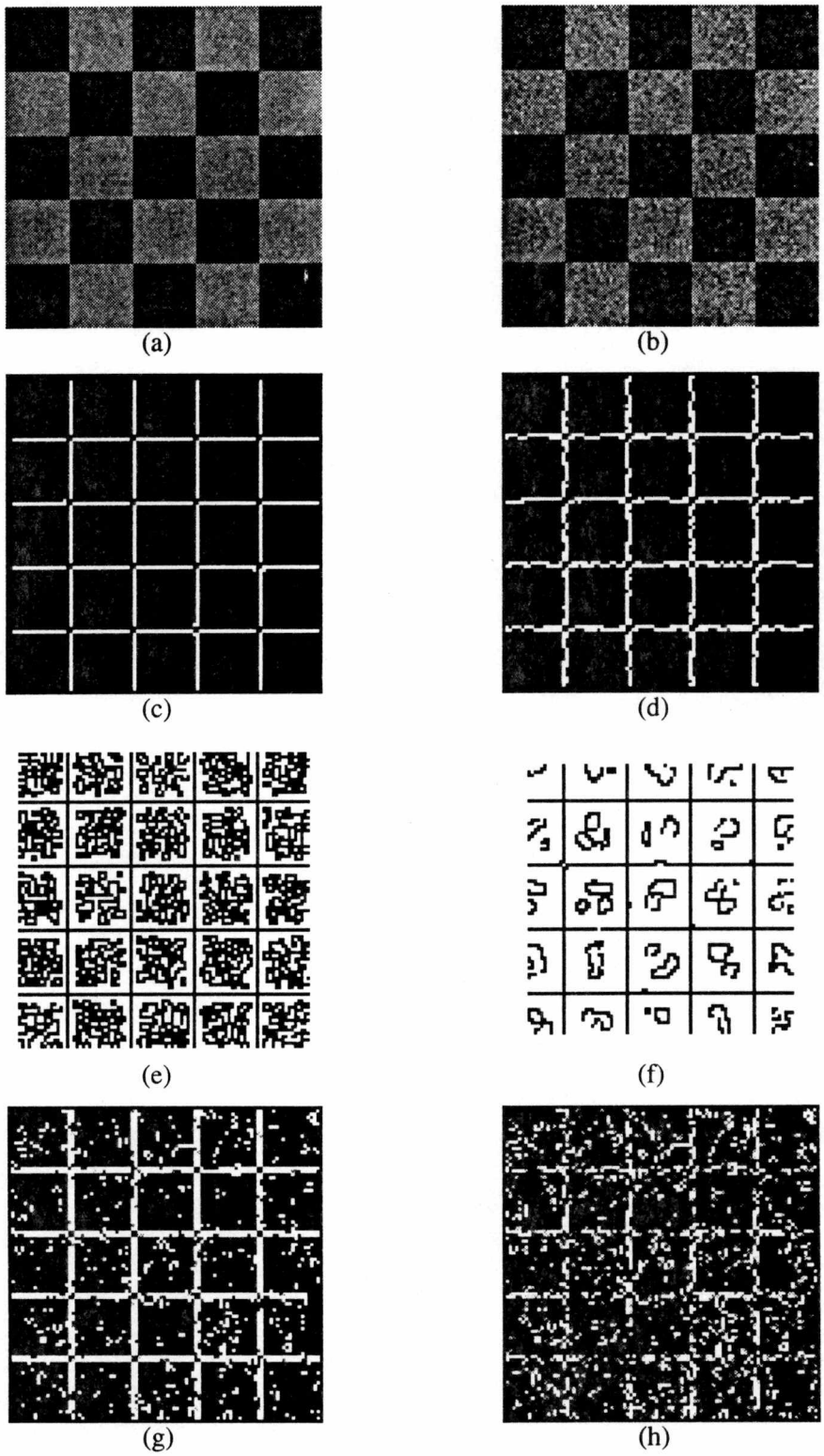


Figure 4.36: *Checkers; fusion with first-order stabilizer: (a) input 1,* $\sigma_n = 10$*, (b) input 2,* $\sigma_n = 20$ *(c) zero-crossing map, (d) gradient edge map, (e) conventional zero-crossing on input 1, (f) conventional zero-crossing on input 2, (g) gradient on input 1, (h) gradient on input 2.*

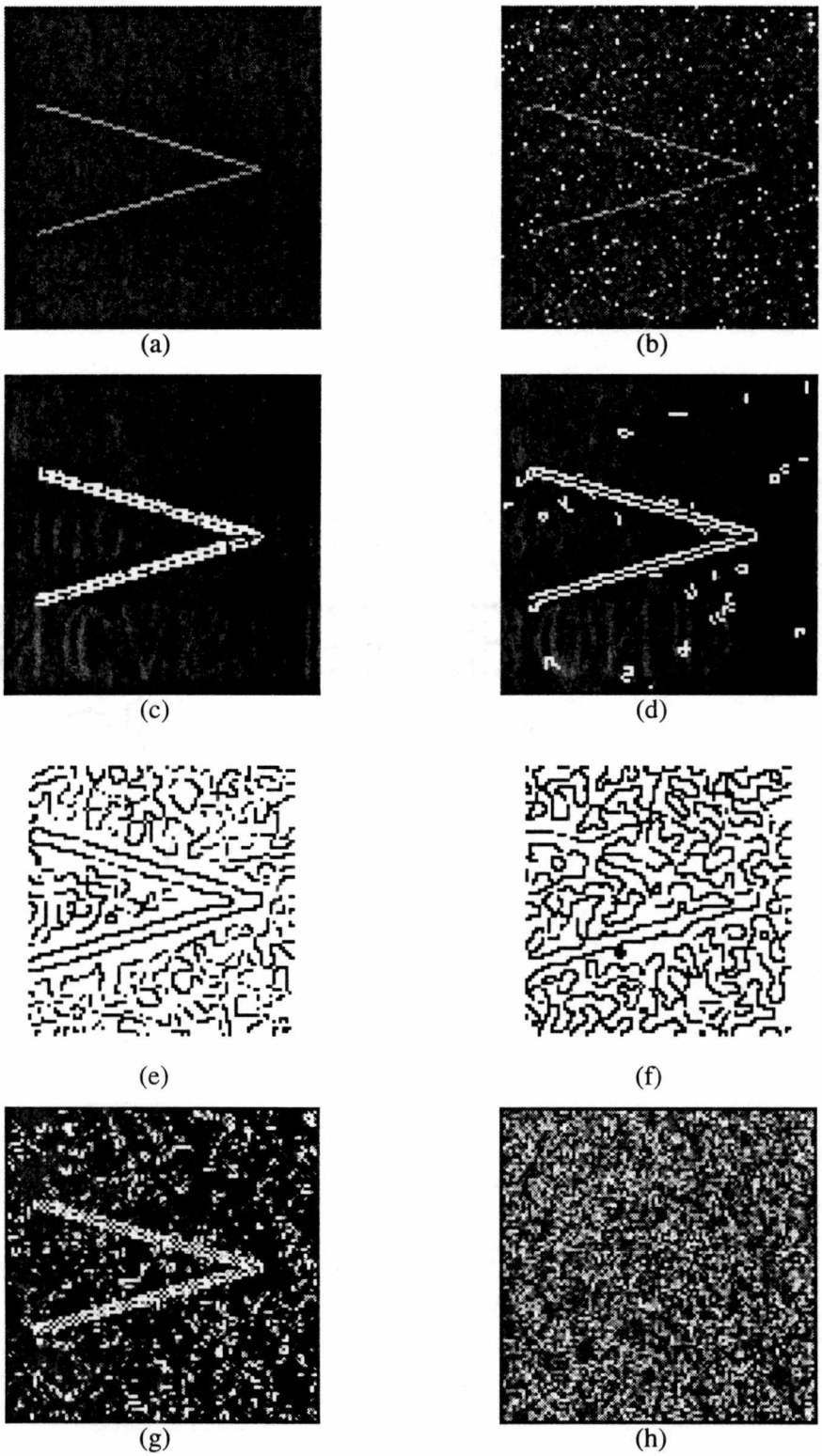


Figure 4.37: *Wake; fusion with first–order stabilizer: (a) input 1,* $\sigma_n = 10$*, (b) input 2,* $\sigma_n = 30$*, (c) zero–crossing map, (d) gradient edge map, (e) conventional zero–crossing on input 1, (f) conventional zero–crossing on input 2, (g) gradient on input 1, (h) gradient on input 2.*

modalities. The second is to evaluate the final edge map when there is information available from one sensory modality but not from the other. An important aspect of this image pair is the fact that the letters "UT" are only "visible" in the intensity image. As seen in Figs. 4.38(c) and (d), most of the edges present in the scene were detected, with the exception of the bottom edge of the cone and the horizon. The bottom of the cone is especially difficult to detect because it simply does not show up in the range image. We have noticed that the horizon tends to disappear when employing first–order stabilizers, and we shall return to this issue later. The benefits of the fusion process are evident, after comparing the nonfused edges to the fused edges.

Figures 4.39 and 4.40 demonstrate the effectiveness of the proposed fusion technique as applied to real data. Figure 4.39 shows fusion results for a pair of real range and intensity images. (Notice that the letter "T" is only "visible" in the intensity image.) These images were captured with an Odetics laser range scanner and have a resolution of 128×128 with 256 gray levels. This kind of scanner provides registered range and intensity images, while at the same time avoiding the triangulation problem that affects other ranging systems.

Normal preprocessing operations for range images include cosine–correction and median filtering with a small size mask (e.g., 3×3). We have not performed these two operations, however, to test the fusion technique under worst case conditions. Naturally, some improvements are to be expected if more pre– and post–processing steps are added to the fusion process.

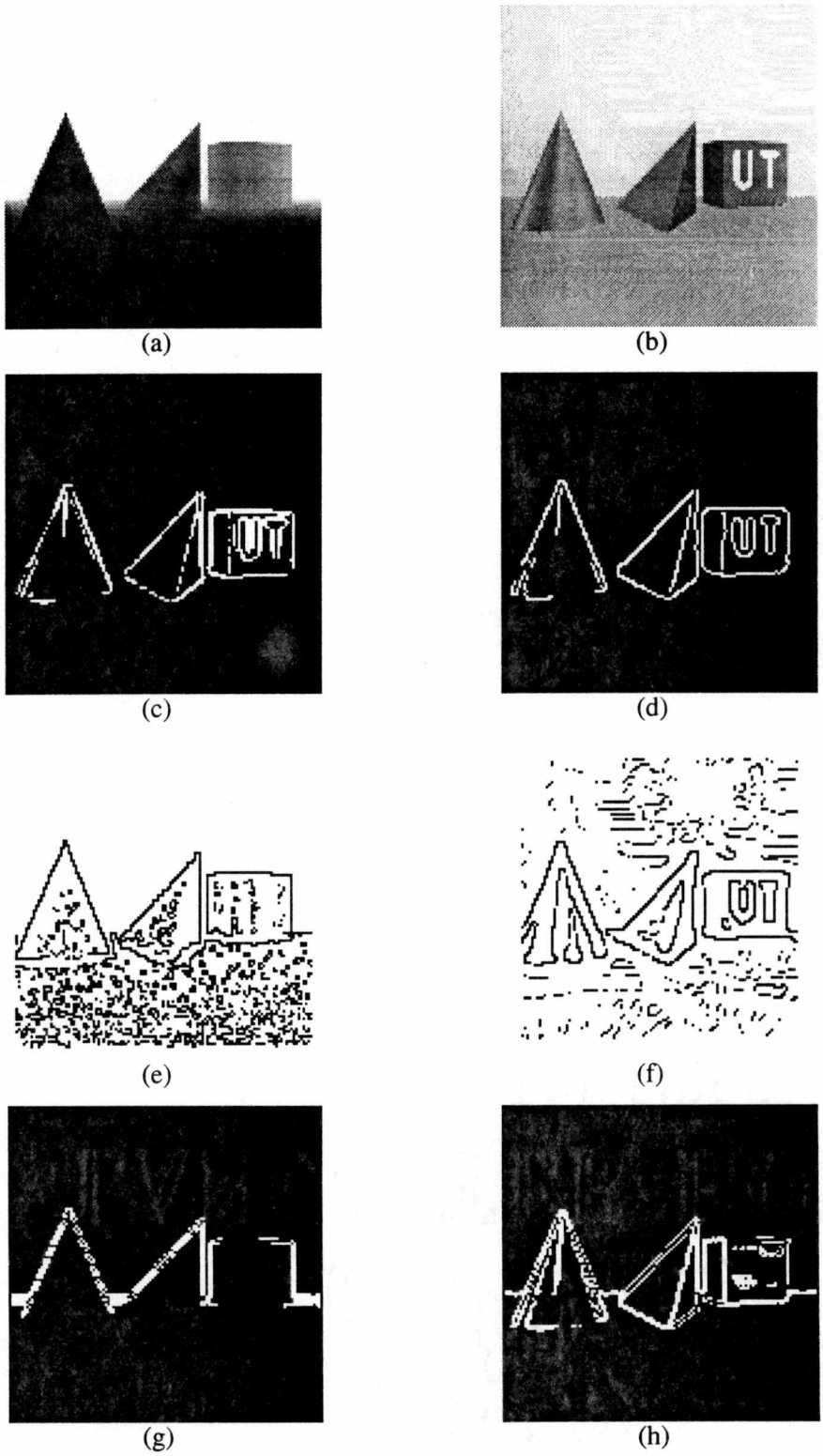


Figure 4.38: *Synthetic objects; fusion with first–order stabilizer: (a) range image, (b) intensity image, (c) zero–crossing map, (d) gradient edge map, (e) conventional zero–crossing on input 1, (f) conventional zero–crossing on input 2, (g) gradient on input 1, (h) gradient on input 2.*

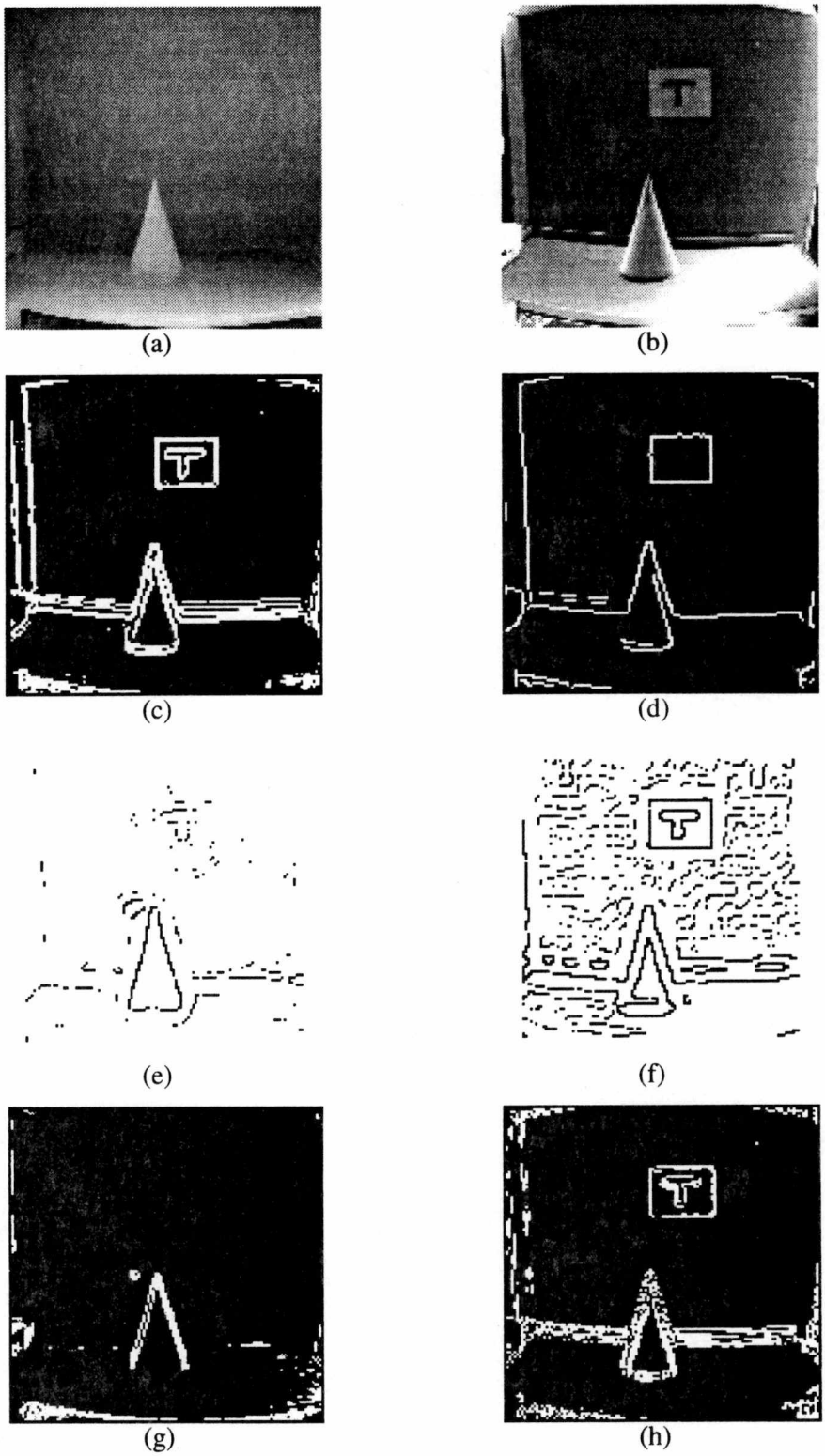


Figure 4.39: *Real intensity–range data; fusion with first–order stabilizer: (a) range image, (b) intensity image, (c) zero–crossing map, (d) gradient edge map, (e) conventional zero–crossing on input 1, (f) conventional zero–crossing on input 2, (g) gradient on input 1, (h) gradient on input 2.*

Figure 4.40 depicts a cluttered scene containing two vehicles in a natural environment. These images are part of a large database available at our Laboratory. By careful examination, we can see that neither the intensity image by itself nor the infrared image by itself provide enough information to detect the features of interest (the vehicles). We can also see, from Figs. 4.40(e) and (f) that conventional zero–crossing techniques were unsuccessful in isolating the vehicles from from the background. The same conclusion can be reached after observing the edges detected by conventional gradient methods (see Figs. 4.40(g) and (h)). The gradient–based edge fusion technique outperformed the zero–crossing based method, and it will be the method of choice for the second–order stabilizer. It was also clear that the first–order stabilizer had difficulties detecting the boundary between the background forest and the smooth foreground. The performance of the zero-crossing based approach was inferior for real cluttered scenes that contained spatially variant signal–to–noise ratios. Further improvements are to be expected if we include knowledge about the sensors and knowledge about the scenes, however, this kind of meta–knowledge has not been incorporated here to make our results valid in general.

The following set of results from Figs. 4.41 through 4.46 illustrate the results of the fusion algorithm for the second–order stabilizer case. (The same set of images was employed for the first–order stabilizer to allow us to compare results).

Figure 4.41 shows results for the "square" image. Notice that Fig. 4.41(c) shows the edges obtained by the gradient method when applied to the ideal square

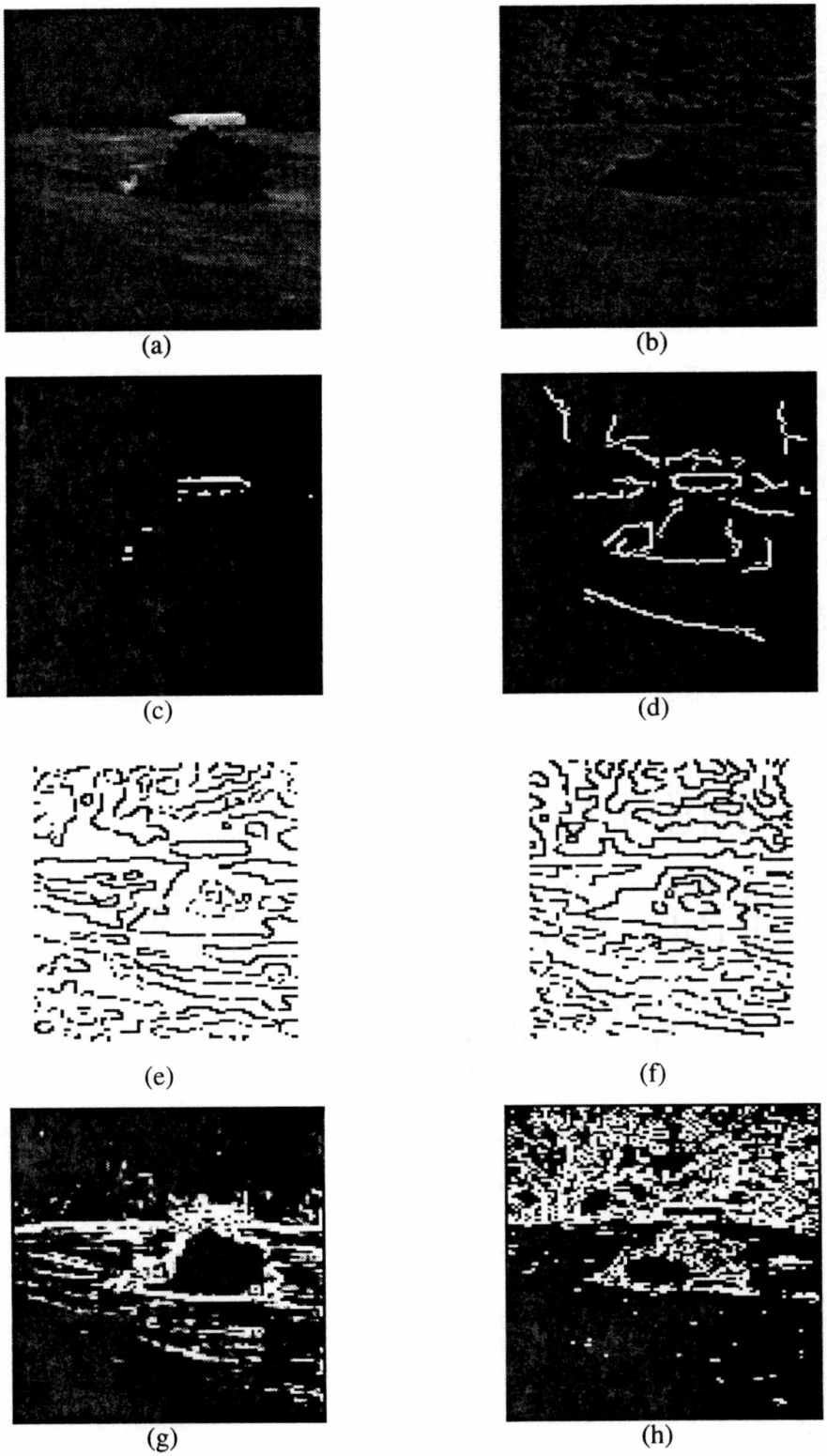


Figure 4.40: *Real infrared–intensity data; fusion with first–order stabilizer: (a) infrared image, (b) intensity image, (c) zero–crossing map, (d) gradient edge map, (e) conventional zero–crossing on input 1, (f) conventional zero–crossing on input 2, (g) gradient on input 1, (h) gradient on input 2.*

image. In a sense, it represents the upper limit of the achievable results for these input images. Figure 4.41(d) shows a typical result obtained from our second–order fusion method. As before, Figs. 4.41(e) and (f) show the results obtained with a conventional zero–crossing algorithm. Figures 4.41(g) and (h) show the edges detected using conventional gradient techniques. It is our conclusion that only the fused edges are clean enough to be useful for a subsequent machine processing stage without human intervention.

Figures 4.42(a) and (b) show the noisy "cake" image pair. Figure 4.42(c) depicts the edges for the noise–free case. Notice that these are not ideal edges, but the best edges that the gradient method could achieve. Figure 4.42(d) shows the edges obtained from the fusion process; although the SNR is very low (4.26) for the top region, we could recover its boundary properly.

Figures 4.43(a) and (b) show the checker images, contaminated with zero–mean Gaussian noise and standard deviation of 10 and 20, respectively. Figures 4.43(c) and (d) show the gradient edges from the noise–free image and the fused edges, respectively. The zero–crossing edges, shown in Figures 4.43(e) and (f) have better resolution at the corner junctions (especially at low σ) but the choice of the optimal sigma is an open problem. As seen in Fig. 4.43(h), conventional gradient edges would be useless to a higher–level processing stage.

Next experiment, Fig. 4.44 shows the results of applying the second–order stabilizer method to a pair of synthetic range–intensity images. The important point here is that most of the edges have been detected, including the "horizon,"—

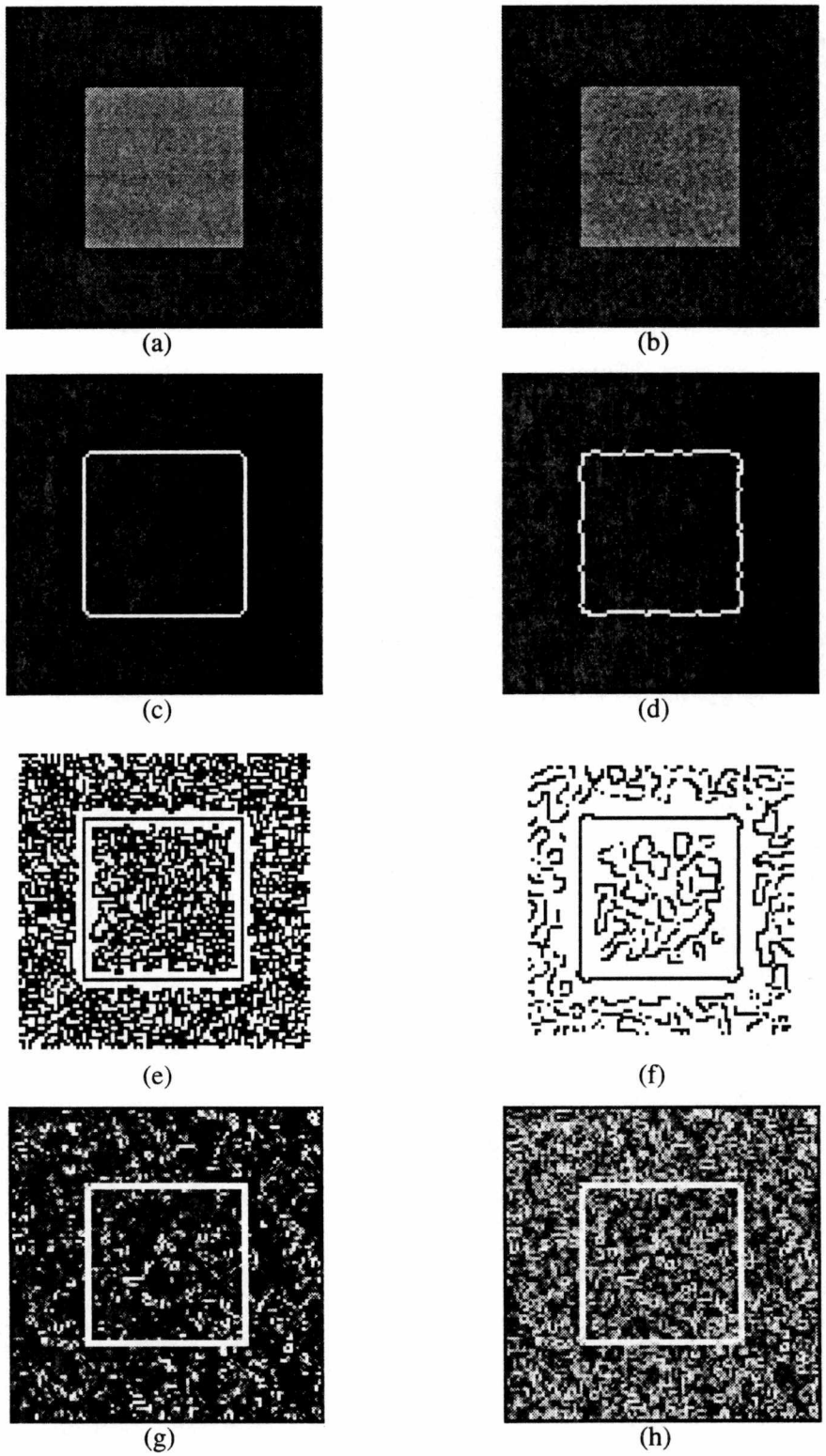


Figure 4.41: *Square; fusion with second–order stabilizer: (a) input 1,* $\sigma_n = 5$*, (b) input 2,* $\sigma_n = 10$*, (c) gradient on ideal image, (d) gradient edge map, (e) conventional zero–crossing on input 1, (f) conventional zero–crossing on input 2, (g) gradient on input 1, (h) gradient on input 2.*

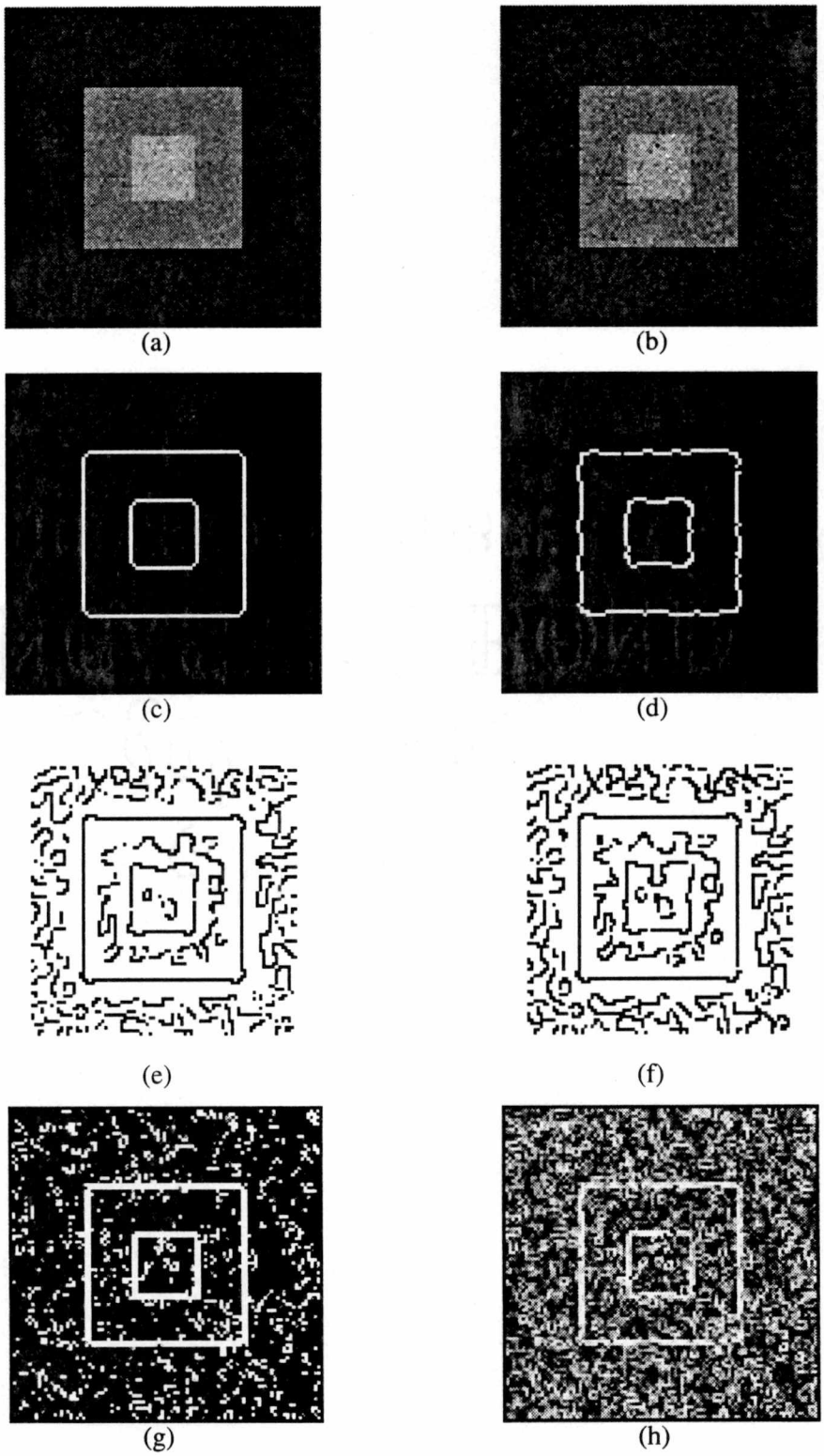


Figure 4.42: *Cake; fusion with second–order stabilizer: (a) input 1,* $\sigma_n = 10$*, (b) input 2,* $\sigma_n = 15$*, (c) gradient on ideal image, (d) gradient edge map, (e) conventional zero–crossing on input 1, (f) conventional zero–crossing on input 2, (g) gradient on input 1, (h) gradient on input 2.*

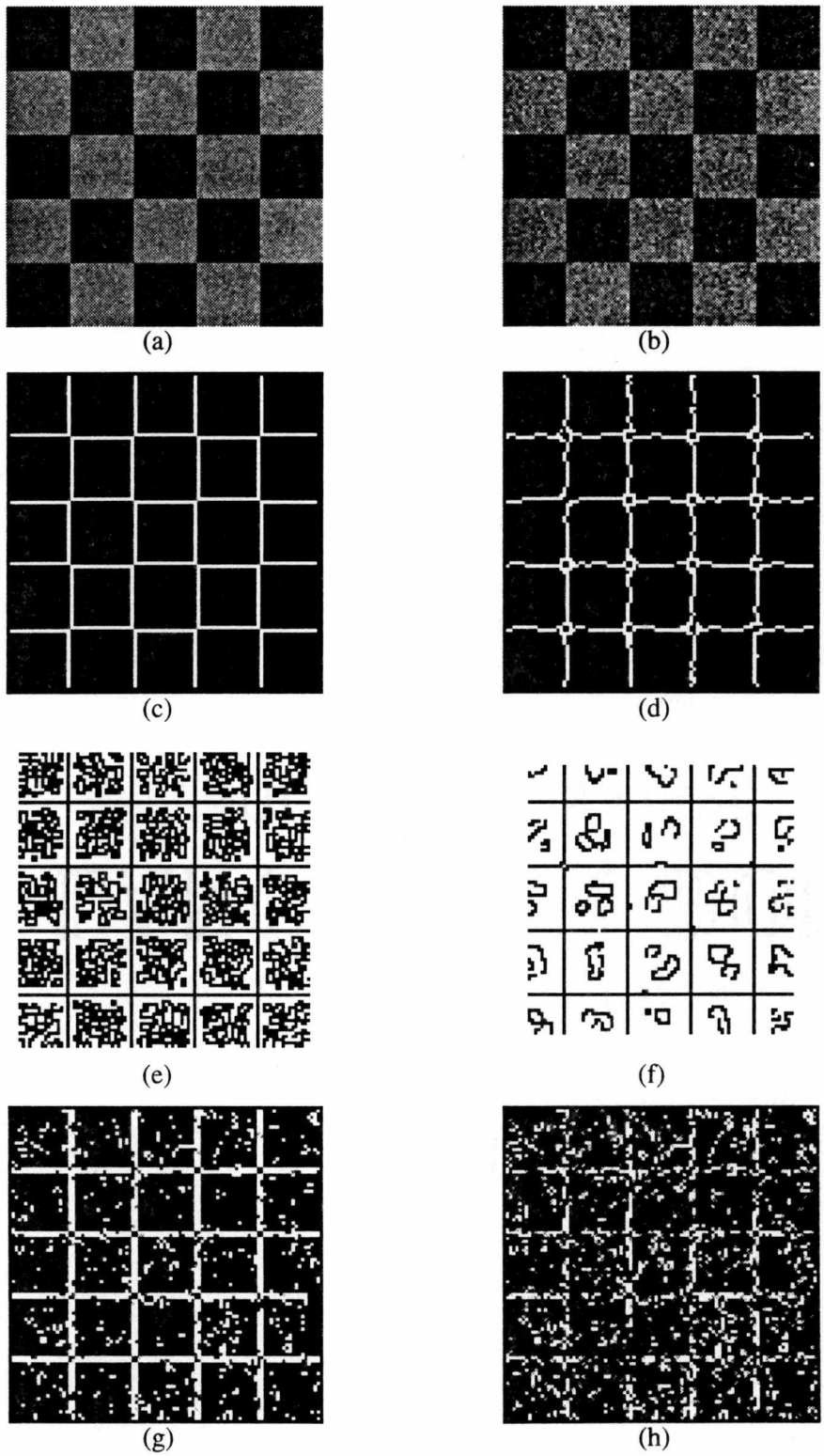


Figure 4.43: *Checkers; fusion with second–order stabilizer: (a) input 1,* $\sigma_n = 10$*, (b) input 2,* $\sigma_n = 20$*, (c) gradient on ideal image, (d) gradient edge map, (e) conventional zero–crossing on input 1, (f) conventional zero–crossing on input 2, (g) gradient on input 1, (h) gradient on input 2.*

the boundary between the floor and the background wall. Additionally, the letters "UT" that are only visible in the intensity image were also detected.

Figures 4.45(a) and (b) show a pair of real range and intensity images. Figure 4.45(d) shows the fused edges obtained with the second–order stabilizer. Notice that the letter "T" was correctly detected. This result is better than any of the edge maps obtained from either zero–crossing method or gradient method. Additional processing that could improve the results even further include cosine correction and median filtering.

The last fusion experiment using SOS (Figure 4.46) shows a real infrared–intensity image pair (vehicles in natural environment). Notice that, as shown in Fig. 4.46(d), for the same input data as in Fig. 4.40, the border between the flat land and the forest is now apparent in the fused edges, a desirable property for edge detection. In later sections we shall illustrate experimentally the reason why the second–order stabilizer has better edge detection results as compared to the first–order stabilizer. This effect becomes evident for edges that humans perceive easily but are not clearly defined for a machine. (There are at least two mechanisms that may be playing a role here. First, lateral inhibition and winner–take–all networks behave approximately as second–order systems, and second, Gestalt groupings and expectancy. (The problem of detecting globally salient structures has been studied by Ullman *et al.* [52]).

The last experiment of this section (see Fig. 4.47) shows the effect that a different set of weights on the input data sources has on the final result. The

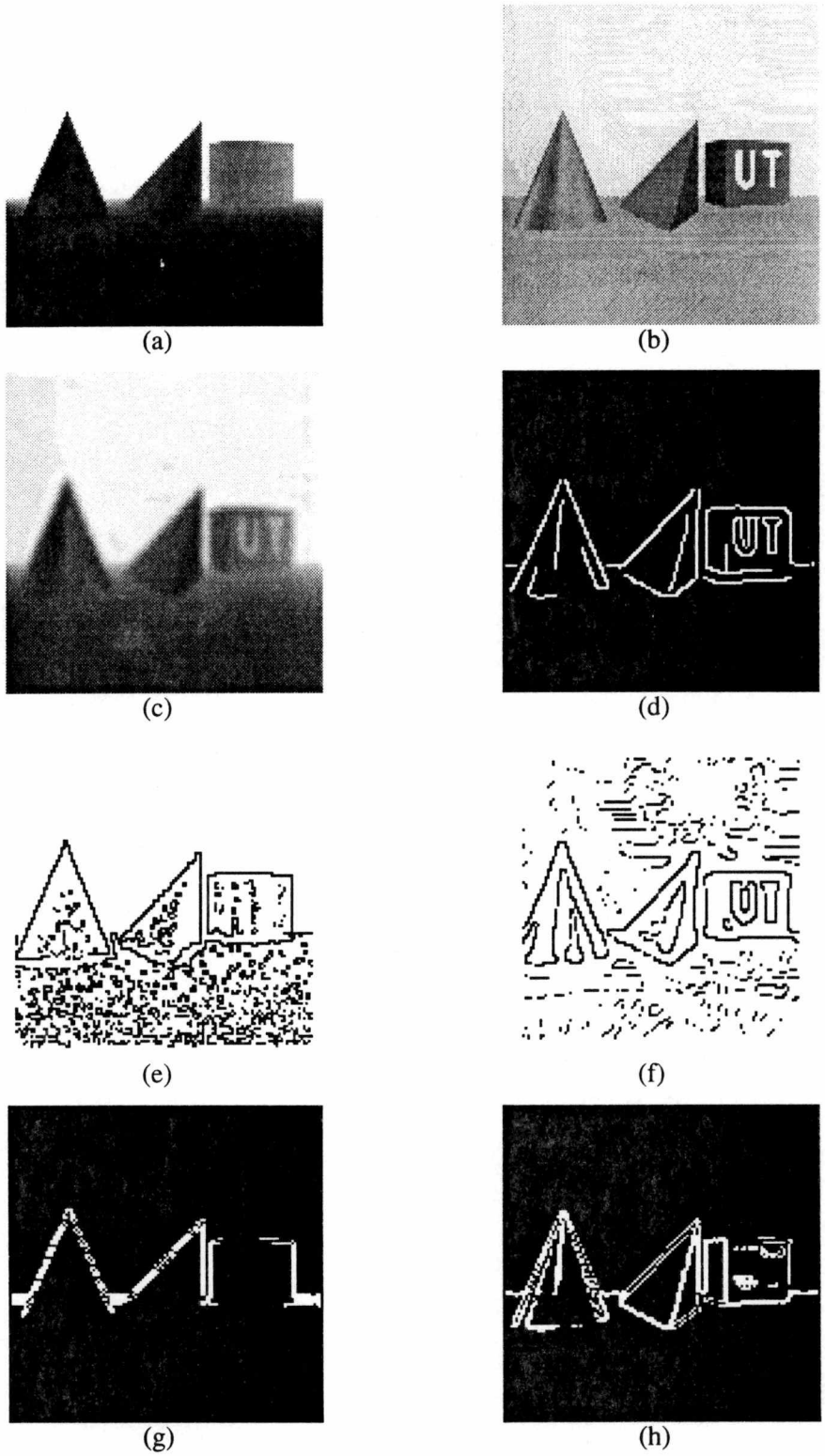


Figure 4.44: *Synthetic objects; fusion with second–order stabilizer: (a) range image, (b) intensity image, (c) fused surface, (d) gradient edge map, (e) conventional zero–crossing on input 1, (f) conventional zero–crossing on input 2, (g) gradient on input 1, (h) gradient on input 2.*

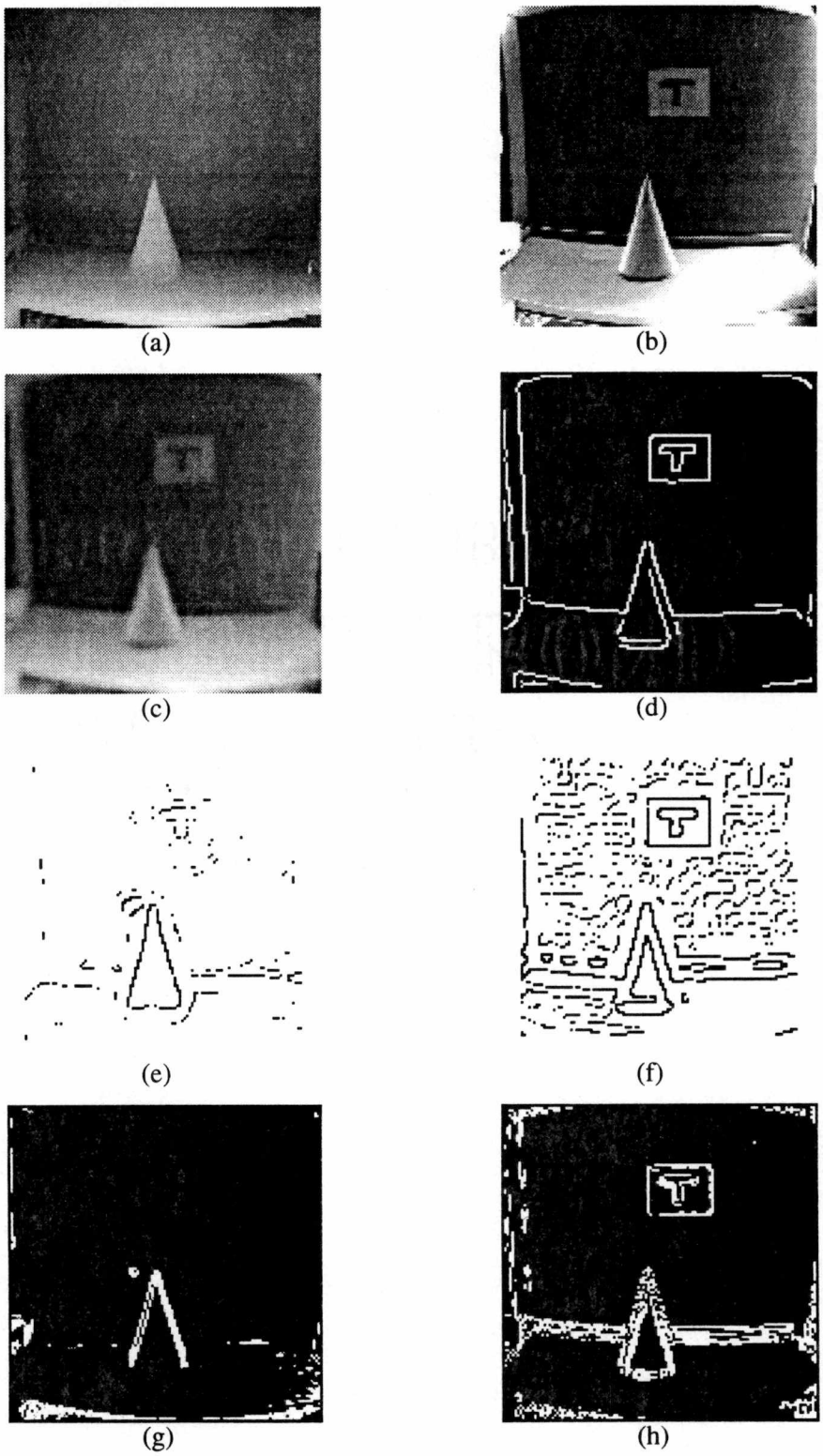


Figure 4.45: *Real intensity–range data; fusion with second–order stabilizer: (a) range image, (b) intensity image, (c) fused surface, (d) gradient edge map, (e) conventional zero–crossing on input 1, (f) conventional zero–crossing on input 2, (g) gradient on input 1, (h) gradient on input 2.*

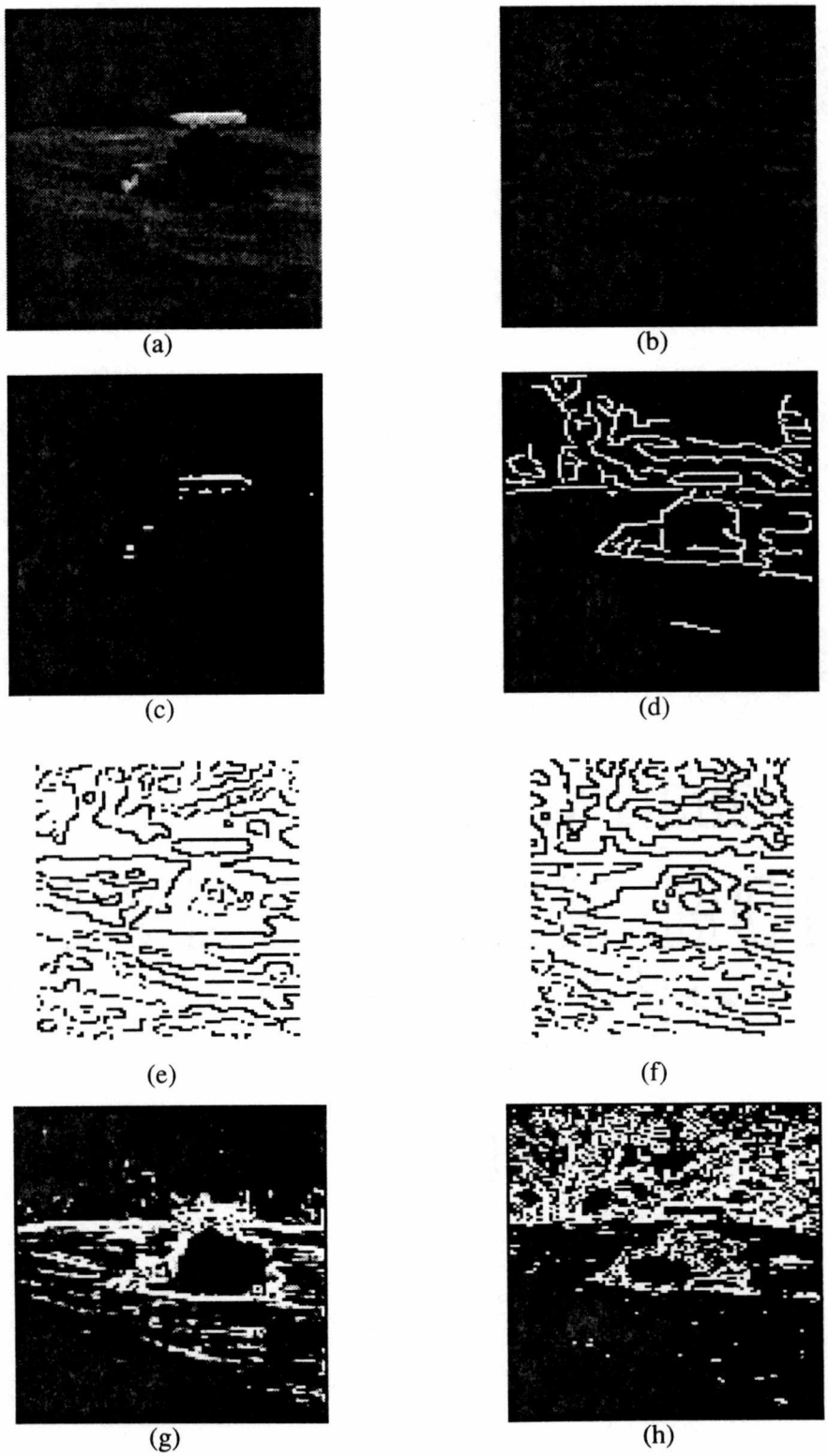


Figure 4.46: *Real infrared–intensity data; fusion with second–order stabilizer: (a) infrared image, (b) intensity image, (c) zero–crossing map, (d) gradient edge map, (e) conventional zero–crossing on input 1, (f) conventional zero–crossing on input 2, (g) gradient on input 1, (h) gradient on input 2.*

problem that we are trying to solve is the detection of the boundary between the flat land and the forest in the infrared–intensity image pair. Since for equally weighted data sources the results were less than satisfactory for the first–order stabilizer case, the following question arised: could it be possible to improve the result by differentially weighting the input sources? Let ω_1 and ω_2 represent the weighting factors associated with the infrared and intensity images, respectively. We found experimentally that the best results were obtained by choosing $\omega_1 = 2$ and $\omega_2 = 1$. We do not expect this ratio to hold in general, because in most cases it is the intensity image which contains the majority of the useful information. The selection of optimal weights is an open problem and we can only give some suggestions towards its resolution. If a large database of images was available we could hope making some reasonable statistical measurements. By partitioning the sample space into a training set and a test set, some cross–validation measures could be obtained. We do not envision a system that will work in general, so we think that knowledge for particular kinds of images, and especially fuzzy reasoning is expected to play a role in this problem.

As seen in Fig. 4.47(d) the border between the flat land and the forest is now apparent in the fused edges. This is in contrast to Fig. 4.40, which did not detect this border properly.

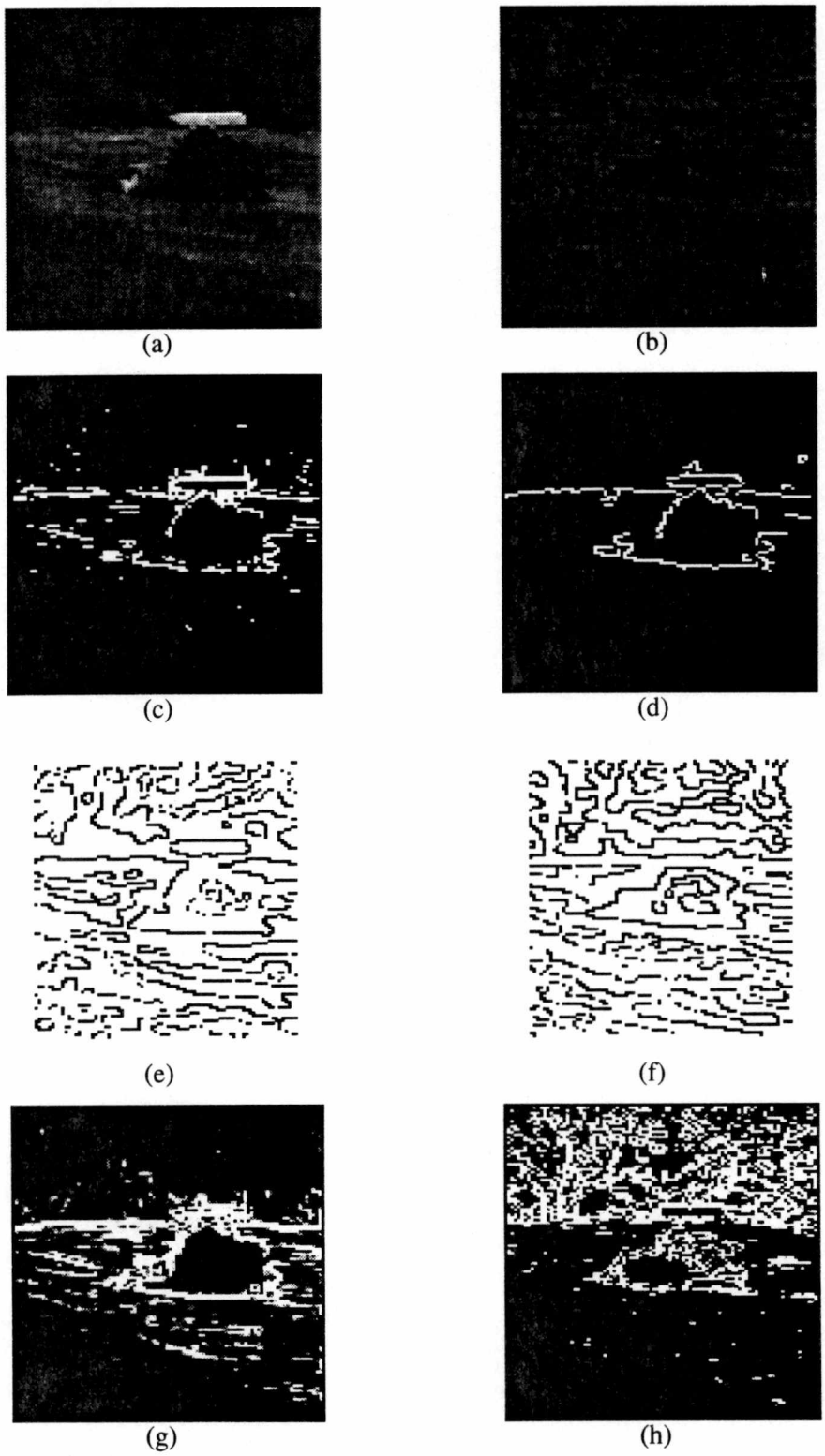


Figure 4.47: *Fusion with first–order stabilizer and weighted inputs: (a) infrared image* $w_1 = 2$, *(b) intensity image,* $w_2 = 1$ *(c) zero–crossing map, (d) gradient edge map, (e) conventional zero–crossing on input 1, (f) conventional zero–crossing on input 2, (g) gradient on input 1, (h) gradient on input 2.*

4.4 Surface Reconstruction Results

In this section we present the surface reconstruction results obtained by applying our paradigm. Following the sequencing introduced in Section 3.5, we will present results in the same order. First, we will show results using the first–order (membrane) stabilizer for both dense and sparse data. Then we will show results using the second–order (thin plate) stabilizer for both dense and sparse data. Finally, we will present results of reconstructed surfaces using dense range data from two noisy knowledge sources. Our first experiment, shown in Fig. 4.48, illustrates the behavior of the first–order stabilizer. Figure 4.48(a) shows the input data, a dense array contaminated with zero–mean Gaussian noise of standard deviation of 10 units. Figure 4.48(b) shows a typical result from the membrane stabilizer. We may notice that a global regularization parameter tends to smooth over the discontinuities. Figure 4.48(c) shows the surface reconstructed after employing variable regularization parameters. We can see that sharp edges are preserved. Figure 4.48(d) shows sparse input data similar to the data that can be acquired from stereo data. Figure 4.48(e) shows the surface reconstructed by the membrane stabilizer. Figure 4.48(f) shows randomized input data with 30% density, and Fig. 4.48(g) shows the corresponding reconstructed surface. Our second experiment, shown in Fig. 4.49, illustrates the behavior of the second–order stabilizer. Figure 4.49(a) shows the input data (same data used in our previous experiment). Figure 4.49(b) shows a typical result using our thin plate stabilizer

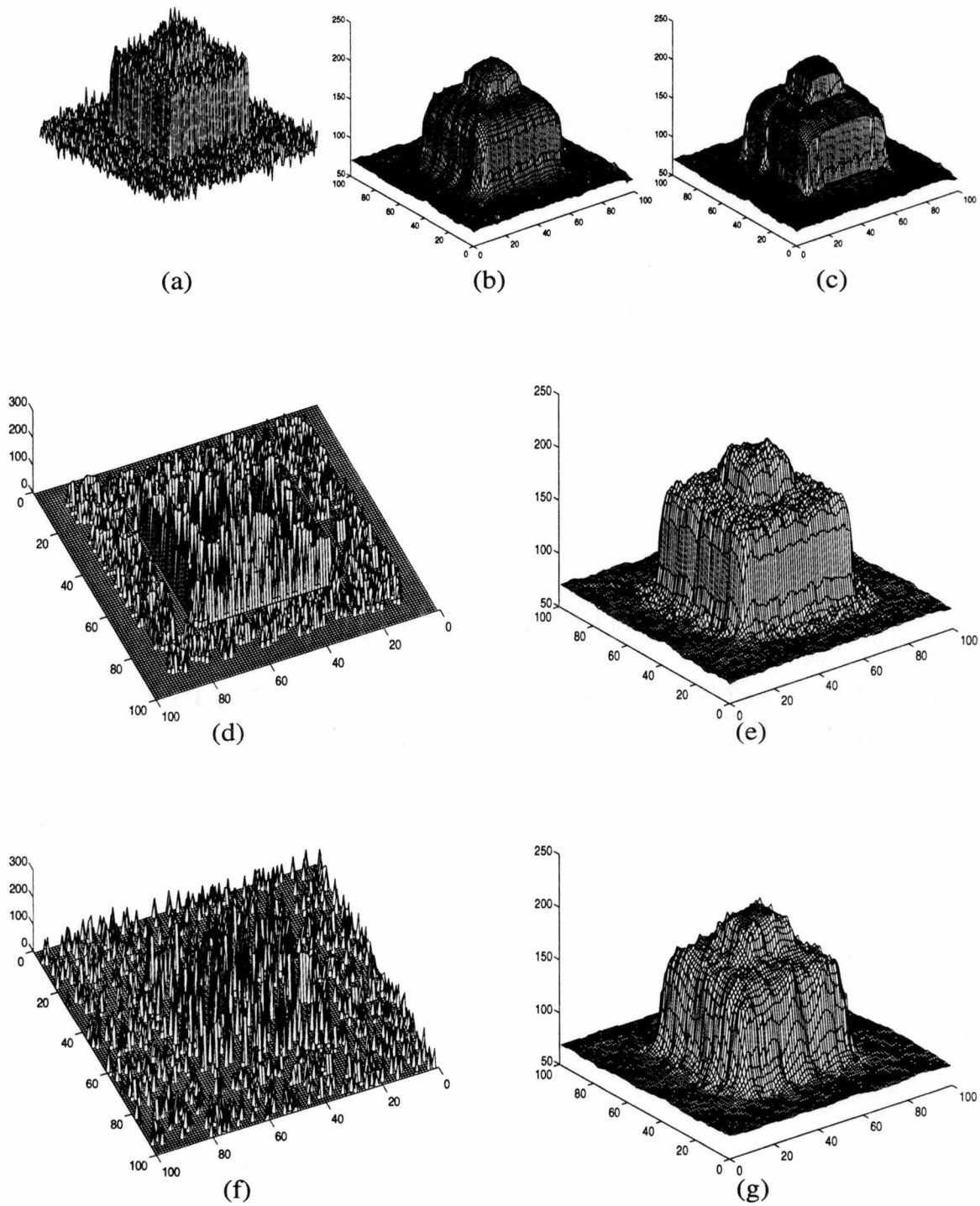


Figure 4.48: *Surface reconstruction with membrane stabilizer: (a) input data, $\sigma_n = 10$, (b) surface recovered with global regularization parameter, (c) surface recovered with space-variant regularization parameters, (d) sparse input data, (e) surface recovered, (f) random input data with 30% density, (g) recovered surface.*

with global regularization parameter. The edges are now better defined at the cost of overshooting (Gibbs phenomenon). Figure 4.49(c) shows improved results around discontinuities when we allow for space–variant regularization parameters. Figure 4.49(d) shows sparse input data similar to the data that can be acquired from stereo data. Figure 4.49(e) shows the surface reconstructed by the thin plate stabilizer. Figure 4.49(f) shows randomized input data with 30% density. Figure 4.49(g) shows the surface reconstructed by the thin plate stabilizer for the random data case.

Our next series of experiments deals with surfaces recovered from information provided by two data sources. In our first experiment, shown in Fig. 4.50, a block object contaminated with zero–mean Gaussian noise of standard deviation $\sigma = 5$ (Fig. 4.50(a)), and a second block contaminated with noise of $\sigma = 10$ (Fig. 4.50(b)), are jointly reconstructed using either a first–order stabilizer (Figure 4.50(d)) or a second–order stabilizer (Figure 4.50(e)). For comparison purposes, Fig. 4.50(c) shows the ideal (noise–free) block. The surface reconstructed using the first–order stabilizer is, in general, smoother than the surface reconstructed using the second–order stabilizer. In fact, it would be the best choice under the given conditions if the discontinuities were better preserved. However, the recovered surface for the second–order stabilizer has better characteristics around the edges, but it created some noticeable oscillations in the smooth regions (regions away from the discontinuities in the ideal case).

In our second experiment, two Gaussian contaminated cake objects are joint-

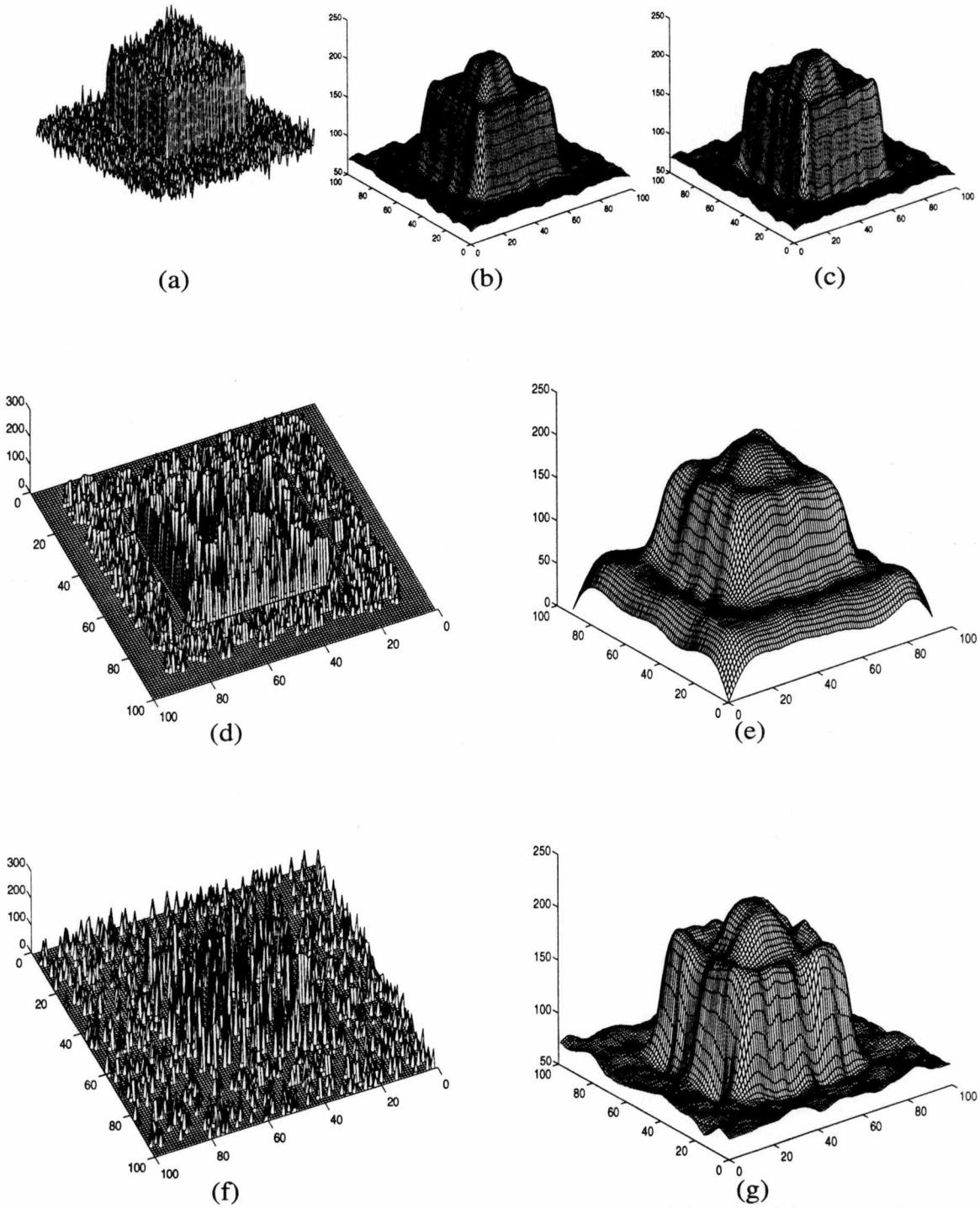


Figure 4.49: *Surface reconstruction with thin plate stabilizer: (a) input data, $\sigma_n = 10$, (b) recovered surface with global regularization parameter, (c) recovered surface with space-variant regularization parameters, (d) sparse input data, (e) surface recovered, (f) random input data with 30% density, (g) recovered surface.*

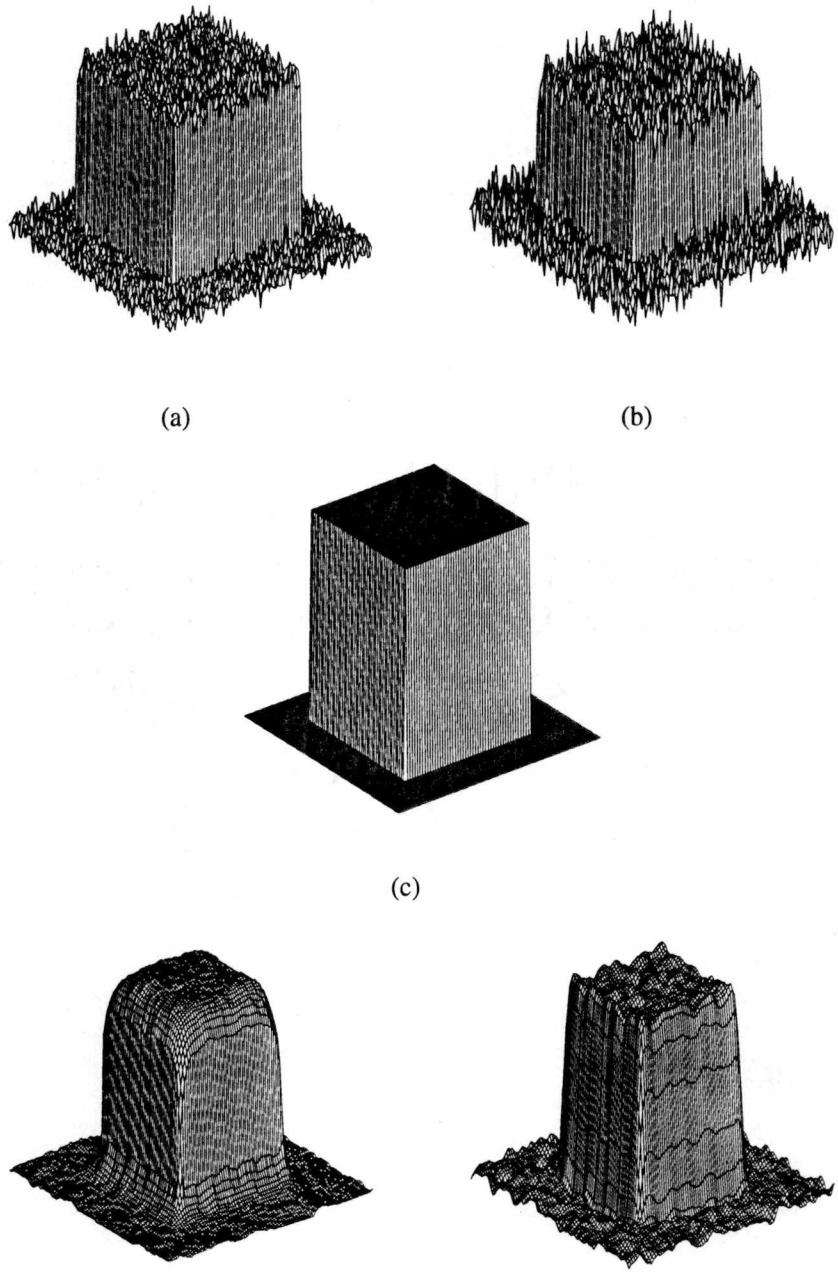


Figure 4.50: *Square; surface reconstruction: (a) input 1, $\sigma_n = 5$, (b) input 2, $\sigma_n = 10$, (c) ideal surface, (d) recovered surface, first–order stabilizer, (e) recovered surface, second–order stabilizer.*

ly reconstructed. The first input, Fig. 4.51(a), was contaminated with noise of standard deviation $\sigma = 10$; the second input, Fig. 4.51(b), was contaminated with noise of standard deviation $\sigma = 15$. The ideal cake is shown in Fig. 4.51(c). Figure 4.51(d) shows the reconstruction achieved using the first–order stabilizer. Figure 4.51(d) shows the reconstruction achieved using the second–order stabilizer. We can see that the smaller box tended to blend in. This happened because it has a SNR of 4.3 as opposed to the bigger box (the base of the cake), which has a SNR of 8.2. As in our previous experiment, the second–order response shows more oscillations and steeper edges. A natural extension of our work would be to keep the first–order responses in regions away from discontinuities and to keep the second–order responses at or near discontinuities. (This extension would be obvious to implement if the segmentation problem had been solved previously, but this is not generally the case.)

Our third experiment, Fig. 4.52, shows reconstruction results for a more complex object. The first input, Fig. 4.52(a), was contaminated with zero–mean Gaussian noise with a standard deviation of $\sigma = 10$. The second input was contaminated with the same kind of noise, but $\sigma = 20$. The reconstructed surface using the first–order stabilizer is shown in Figure 4.52(d). The reconstructed surface using the second–order stabilizer is shown in Figure 4.52(e). Although the input data is severely contaminated, the recovered surfaces from both stabilizers approximate the ideal surface fairly well. All of the boxes were detected and the reconstructed surfaces provided enough information for an autonomous manipu-

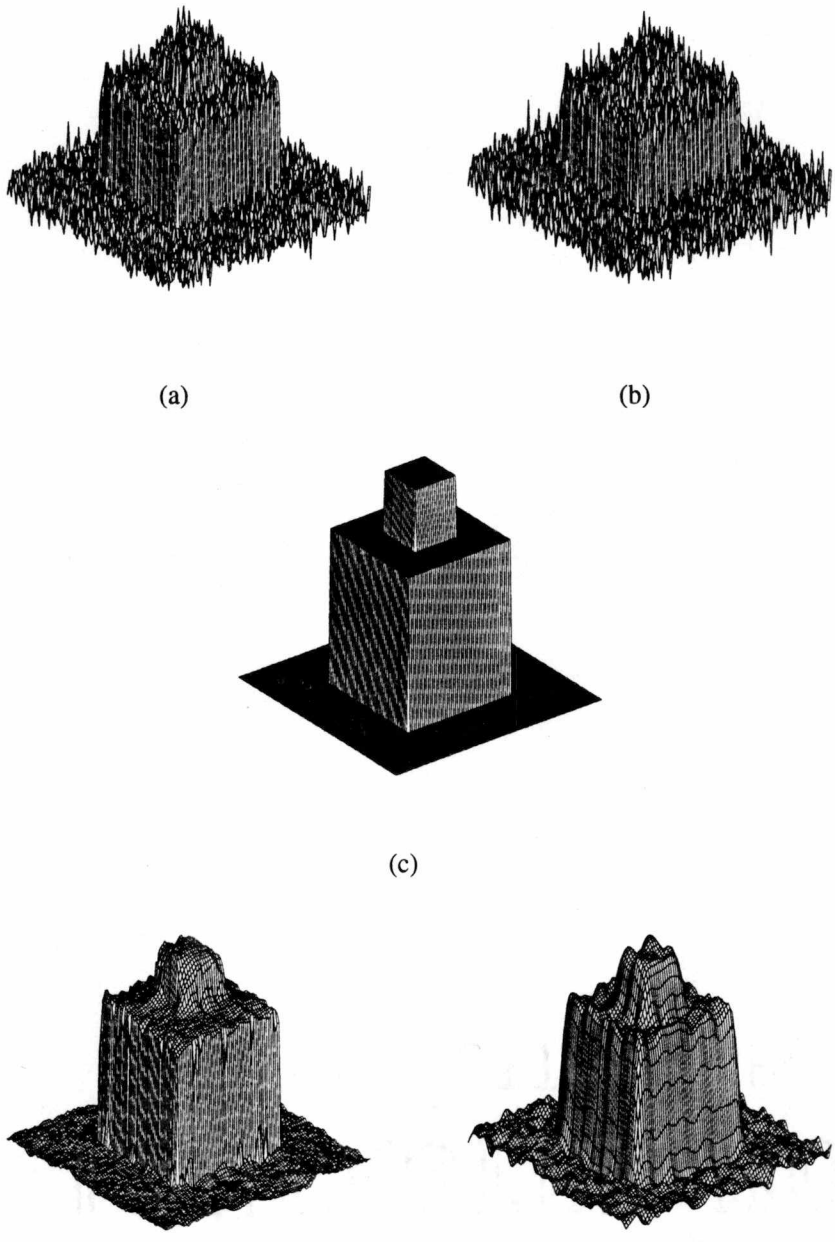


Figure 4.51: *Cake; surface reconstruction: (a) input 1, $\sigma_n = 10$, (b) input 2, $\sigma_n = 15$, (c) ideal surface, (d) recovered surface, first–order stabilizer, (e) recovered surface, second–order stabilizer.*

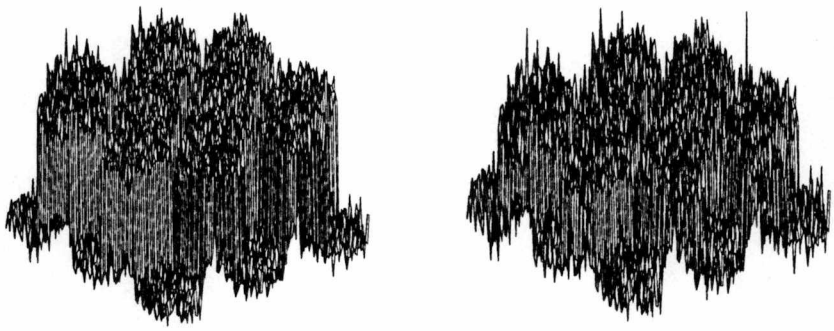
lation task or for navigation purposes. We assume that higher–level processing modules would be available to provide world coordinates, but this problem is beyond the scope of this dissertation.

Our final surface reconstruction experiment involves the range image of a cup (generously provided by Jean–Michel Jolion, Université Claude Bernard, France). The purpose of this experiment is to measure the effectiveness of our surface reconstruction algorithms for a real man–made object containing curved surfaces. As seen from Figs. 4.53(d) and (f) the handle of the cup was not recovered. This was to be expected because the outliers in the data obscured this weak feature.

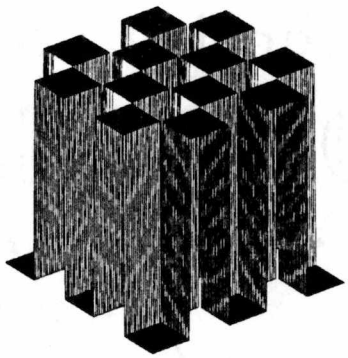
4.5 Color Edge Detection Results

We have tested our color edge detection technique on several color images. Both synthetic and real images have been employed. For each color image the red, green, and blue channels are extracted. This monochromatic information is then processed to produce the final fused edge map. Although a zero–crossing edge map was produced for each input image, only the gradient edge map is shown in the pictures to avoid cluttering.

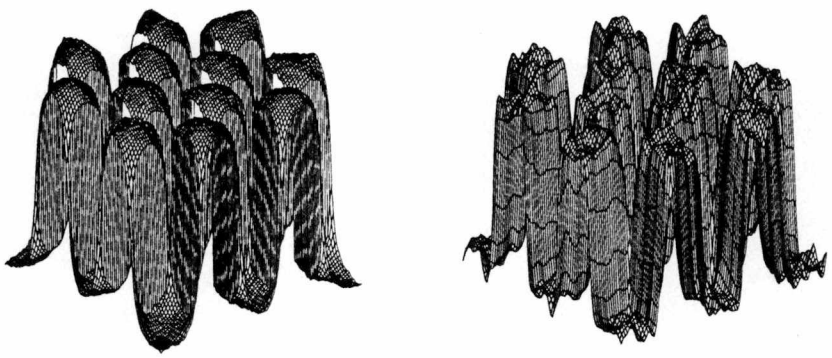
Our first (synthetic) color image (Fig. 4.54(a)) shows three vertical color bands, red, green, and blue, respectively, from left to right. Since these are isoluminance bands, any intensity–only edge detector would miss both inner boundaries (i.e., it would miss the color edge between the red and green strips, and the edge between



(a) (b)



(c)



(d) (e)

Figure 4.52: *Checkers; surface reconstruction: (a) input 1, $\sigma_n = 10$, (b) input 2, $\sigma_n = 20$, (c) ideal surface, (d) recovered surface, first–order stabilizer, (e) recovered surface, second–order stabilizer.*

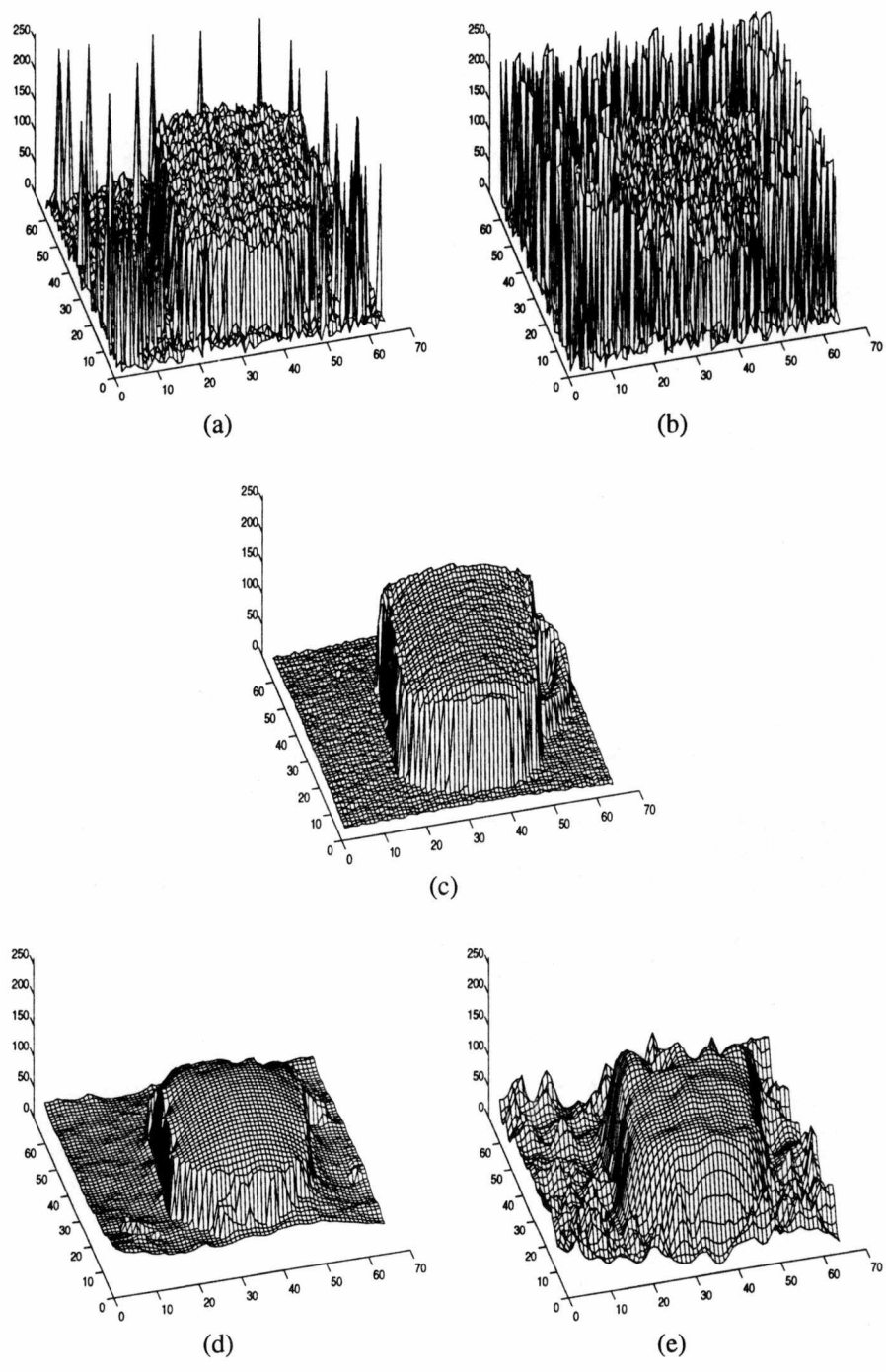
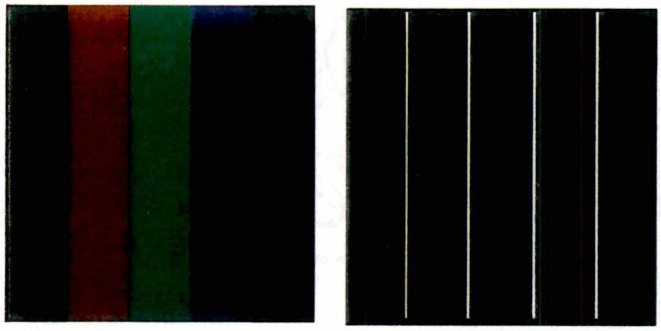


Figure 4.53: *Surface reconstruction of cup: (a) input 1, $\sigma_n = 10$, (b) input 2, $\sigma_n = 20$, (c) ideal surface, (d) recovered surface, first–order stabilizer, (e) recovered surface, second–order stabilizer.*



(a) (b)

Figure 4.54: *Synthetic image of isoluminant color bands: (a) input color image, (b) color edges detected by our fusion technique.*

the green and the blue strips). Since the intensity for a multichannel image is typically computed as $I = \sqrt{R^2 + G^2 + B^2}$, and since each of the color bands have the same individual intensity for this particular test image, we conclude that a color edge detector is needed to produce the correct results.

Although it is unlikely that every real scene will posses many isoluminant edges, we cannot rule this possibility out. Our color edge detector, as seen in Fig. 4.54(b), responded correctly in detecting isoluminant color edges.

Figures 4.55 through 4.59 show typical results obtained by our method for both synthetic and real color scenes. In all cases, the topmost row shows the (color) input image; the second row from the top shows the red, green, and blue channels, respectively; the third row shows typical gradient edge maps for each channel data; finally at the bottom, we have the fused edge map. It is apparent from these experiments that the fused result is consistently better than each of the individual edge maps.

Our second color image, Fig. 4.55, corresponds to a synthetic scene depicting a robot manipulator and several colored objects. To better understand the results, we will describe the major surfaces in the scene. The floor is gray and shows specular reflections and shadows. The base of the robot is blue–gray. The arm is mostly gray a with blue wrist. The object in the gripper is yellow, the cylindrical object is green, and the flat object is red. The arm casts a shadow on the floor and on the base. From the fused edges we see that almost all of the major edge features of the scene were detected. Missing are the roof edge on the flat red object and the perceived edge in the middle section of the arm. The reasons for the omissions are that our algorithm was designed to detect step discontinuities, and that the bottom edge of the square middle section of the arm is indistinguishable from its self–shadows.

The issues of shadow identification and specularity are beyond the scope of this dissertation. The interested reader may wish to consult the paper by Jiang and Ward [26] and references therein.

Figure 4.56 shows the color image of Lena. The background is mostly light brown, with the exception of the dark brown arc on the right, the yellowish inverted V above the arc, and the yellowish area near the right bottom corner. The hat has two tones of beige and the feathers on the hat are purple. All major edges were detected with two not–so–obvious omissions: the gap at the top of the hat, and the vertical edge that goes from the top of the hat to the top of the image. These perceptual edges are very difficult to detect because there are only minor

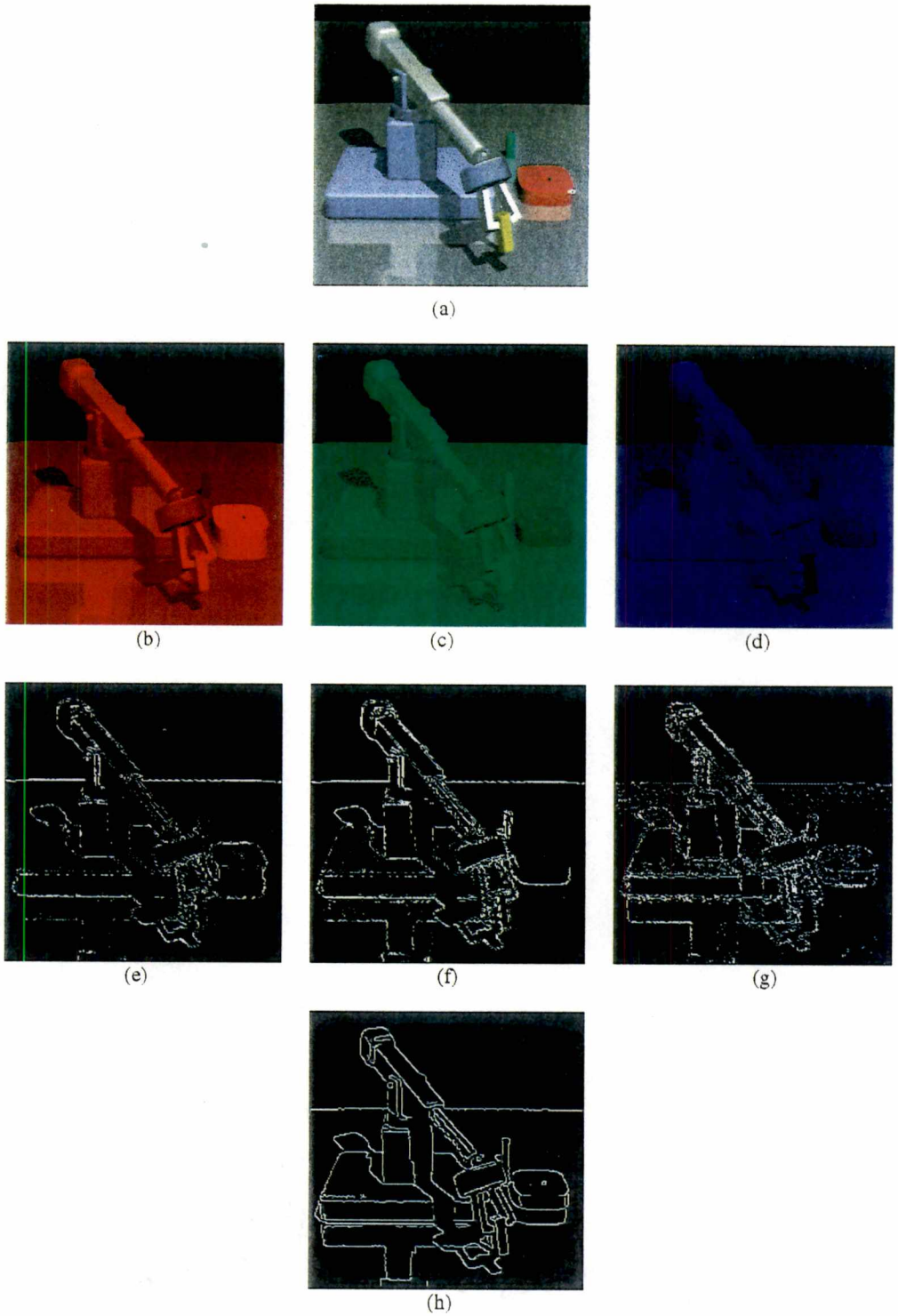


Figure 4.55: *Synthetic arm; color edge detection: (a) input color image (b) red channel, (c) green channel, (d) blue channel, (e) red edge map, (f) green edge map, (g) blue edge map, (h) fused edges.*

color variations between the regions across these boundaries. Even though we could adjust the thresholds to make sure that these edges do appear, we have not done so; rather, we have shown typical results. Otherwise, the solution would be too problem dependent and would lose generality.

Figure 4.57 shows a variety of red and green colored peppers. We shall describe the six principal areas in this image. From top to bottom and left to right we have a red pepper, another red pepper, a light green chile pepper, a dark shadow, a dark green pepper, and a third red pepper. We should also note the specular reflections of the red and dark green peppers. If specular reflections needed to be removed, perhaps the best options are the following: (1) in controlled settings, illumination can be arranged to minimize specular reflections, and (2) in less structured environments, polarization filters are a viable option. Our color edge detection algorithm successfully detected most edges (see Fig. 4.57(h)). We can point to three exceptions: (1) the discontinuity between the vertical light green pepper and the horizontal green pepper located near the bottom left of the picture was not detected. The reason for this was that both peppers share almost the same color and illumination. The second and third exceptions correspond to expected edges at the lower half of the central dark green pepper. In both cases there is a gradual (and fuzzy) transition from light to dark. We believe that physical insight obtained from shadow analysis can improve the interpretation of these fuzzy edges. The proper way to add this knowledge to our current formulation is still an open problem.

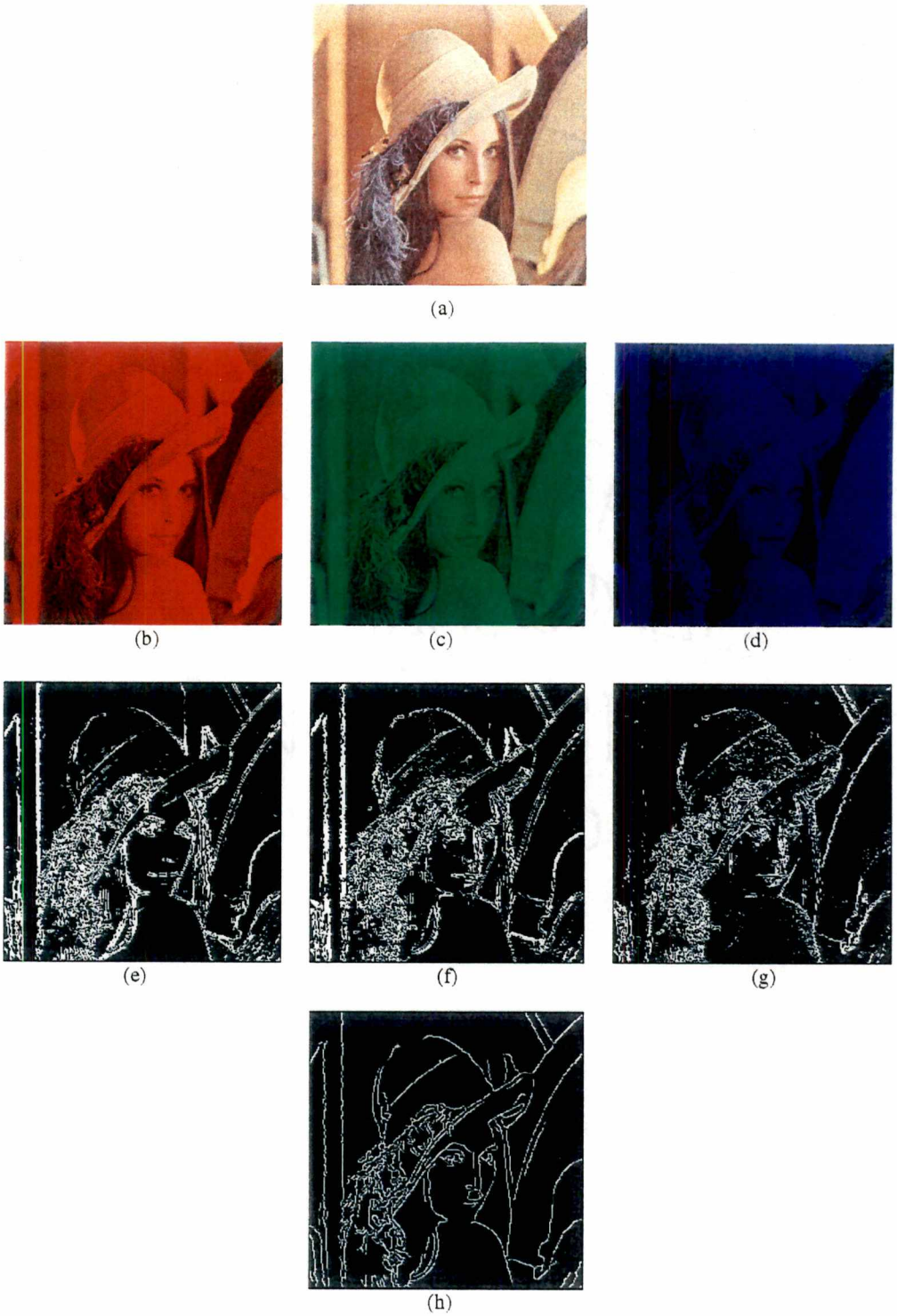


Figure 4.56: *Lena; color edge detection: (a) input color image, (b) red channel, (c) green channel, (d) blue channel, (e) red edge map, (f) green edge map, (g) blue edge map, (h) fused edges.*

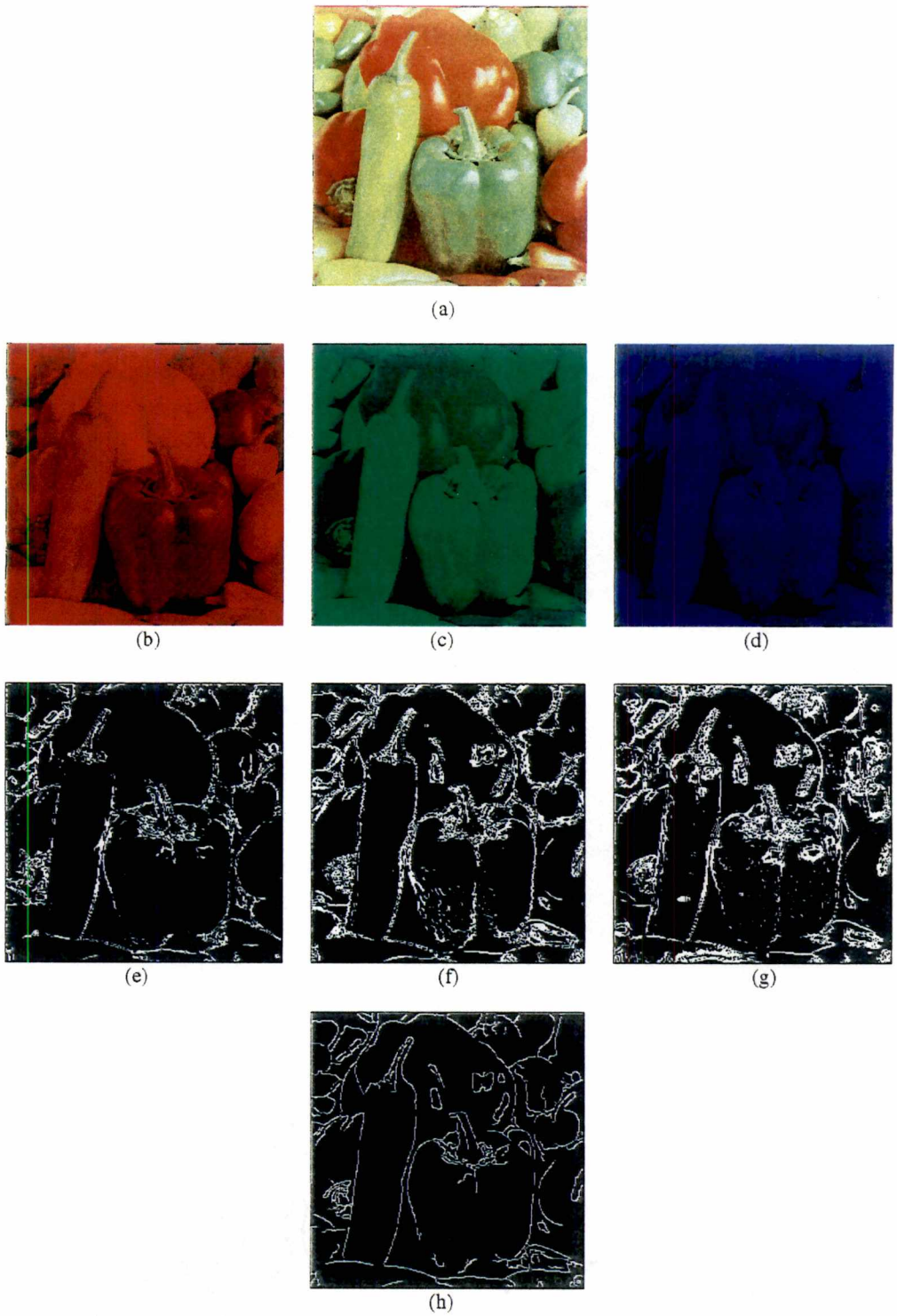


Figure 4.57: *Peppers; color edge detection: (a) input color image, (b) red channel, (c) green channel, (d) blue channel, (e) red edge map, (f) green edge map, (g) blue edge map, (h) fused edges.*

Figure 4.58 shows a parrot perched on a big branch in the outdoors. This image contains edges at a variety of scales, which will allow us to test the strengths and limitations of our color edge detection method in an unstructured setting. (We mentioned earlier the major assumptions under which our technique operates; those assumptions are not enforceable in natural scenes, so the question is how severe is the degradation in the response going to be when the assumptions do not hold.) The parrot body has red feathers, the long wing feathers are copper, the tail is blue, the branch is greenish–brown, and the surrounding foliage is green. As shown in Fig. 4.58(h), all major edge features were detected. Some of the fine textured chest feathers are missing as well as a better defined edge for the chest. We believe that color constancy and/or area–based color segmentation techniques would be useful additions to our color edge detection method to help improve these results. Further improvements would also require better noise estimation techniques that could operate for textured images such as this one.

Figure 4.59 shows another outdoor scene containing a color–banded hot air balloon and the author and his children. Since we do not have control over the illumination (the sun), some reflections and hard shadows are evident. As seen from the fused edges, Fig. 4.59(h), the white letters on the blue portion of the balloon were detected quite well (the entire message reads News–Sentinel). The strong illumination from the sun blended the rest of the letters, the balloon, and the sky into one color. Therefore, no features can be seen in the area surrounding the top right corner of the image. The addition of another kind of sensory

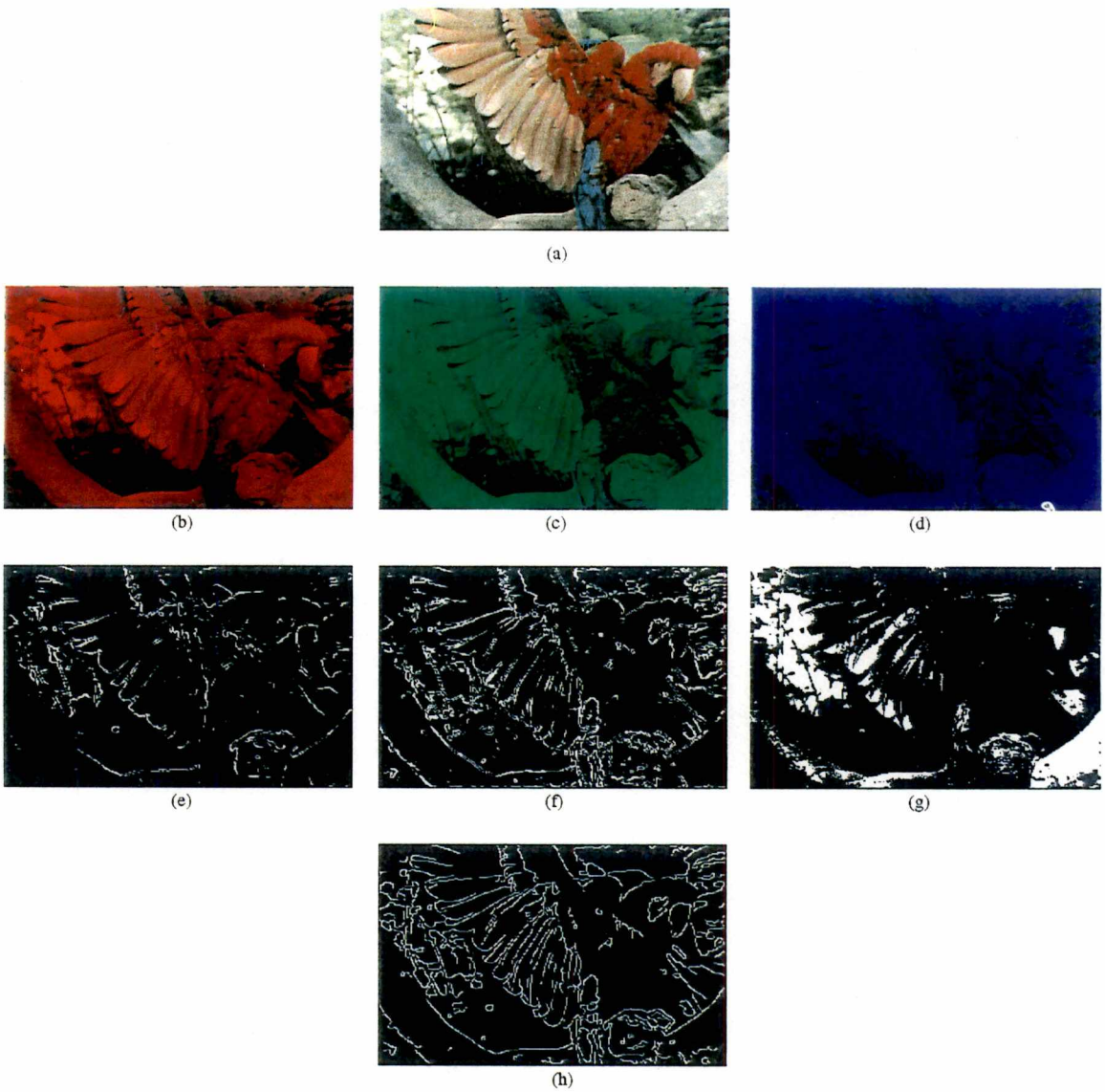


Figure 4.58: *Parrot; color edge detection: (a) input color image, (b) red channel, (c) green channel, (d) blue channel, (e) red edge map, (f) green edge map, (g) blue edge map, (h) fused edges.*

information, such as range sensing, would allow improving the definition of the right curved portion of the balloon against the sky, however, it would not help to detect the letters.

Our final color image, Fig. 4.60, shows a Cincinnati–Milacron robotic manipulator and a control panel existent in our laboratory. The control panel is gray with white indicators and knobs. The base and most of the manipulator is beige, the next section is orange, the force/torque wrist assembly is light gray (aluminum), and the gripper is black. The picture contains some strong specular reflections and dark shadows. We have chosen this image as a typical example of the conditions under which a (color) machine vision system has to operate. The problems of shadowing and specular reflections can be handled by separate software and/or hardware modules (such as polarization filters). An analysis of the results and problems encountered with our color edge detection method indicates that a single sensory modality by itself can not explain all the physical causes of a given (edge) feature. Shadow edges, specular edges, etc., require separate physical interpretations. A more complete system would need to integrate color edge detection, color–region detection, shadow analysis and range sensing.

4.6 Roof Edge Detection Results

In this section, we present a simple experiment that illustrates the operation of a new roof edge detection algorithm. We have chosen a synthetic range image

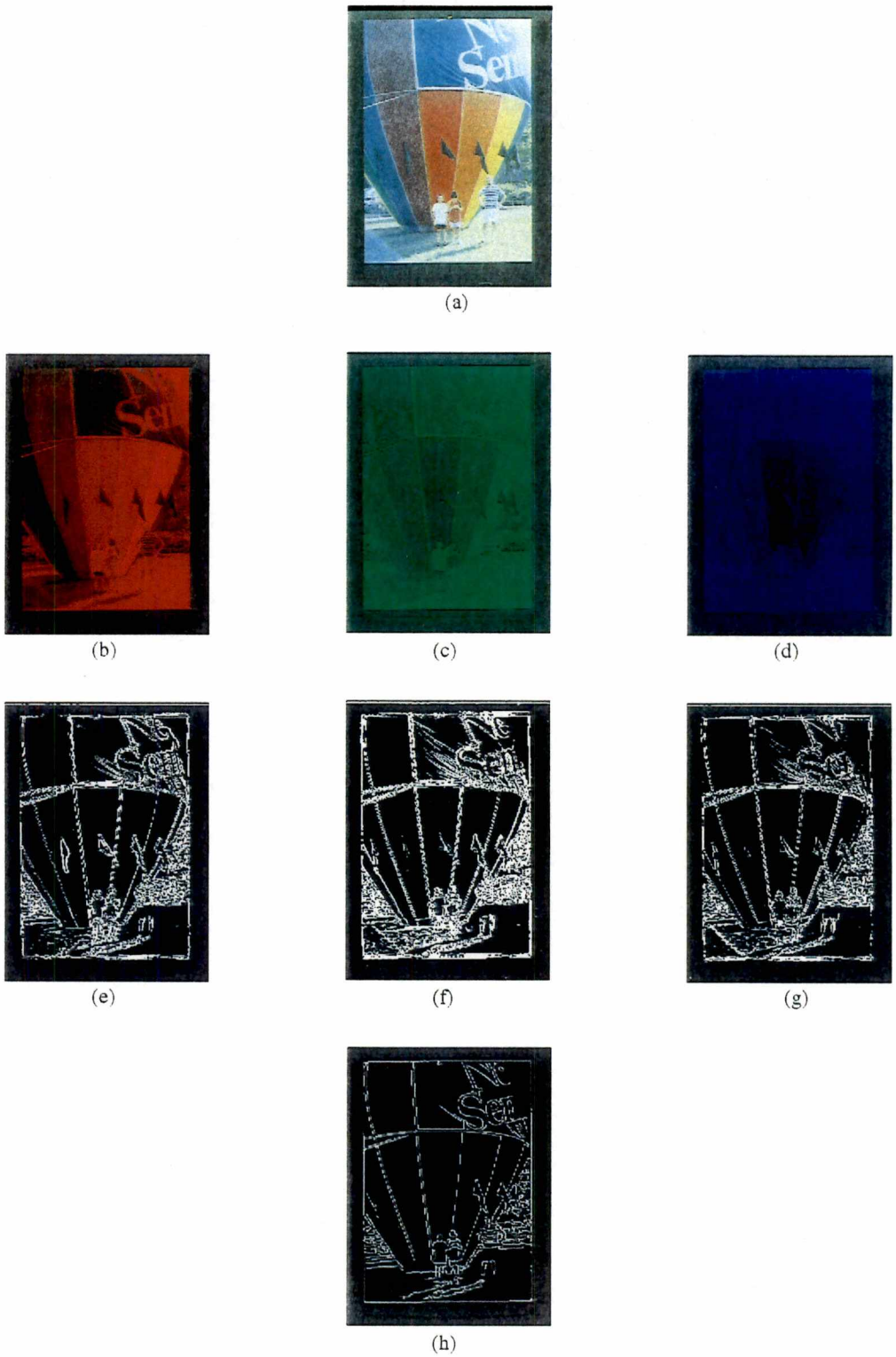


Figure 4.59: *Balloon; color edge detection: (a) input color image, (b) red channel, (c) green channel, (d) blue channel, (e) red edge map, (f) green edge map, (g) blue edge map, (h) fused edges.*

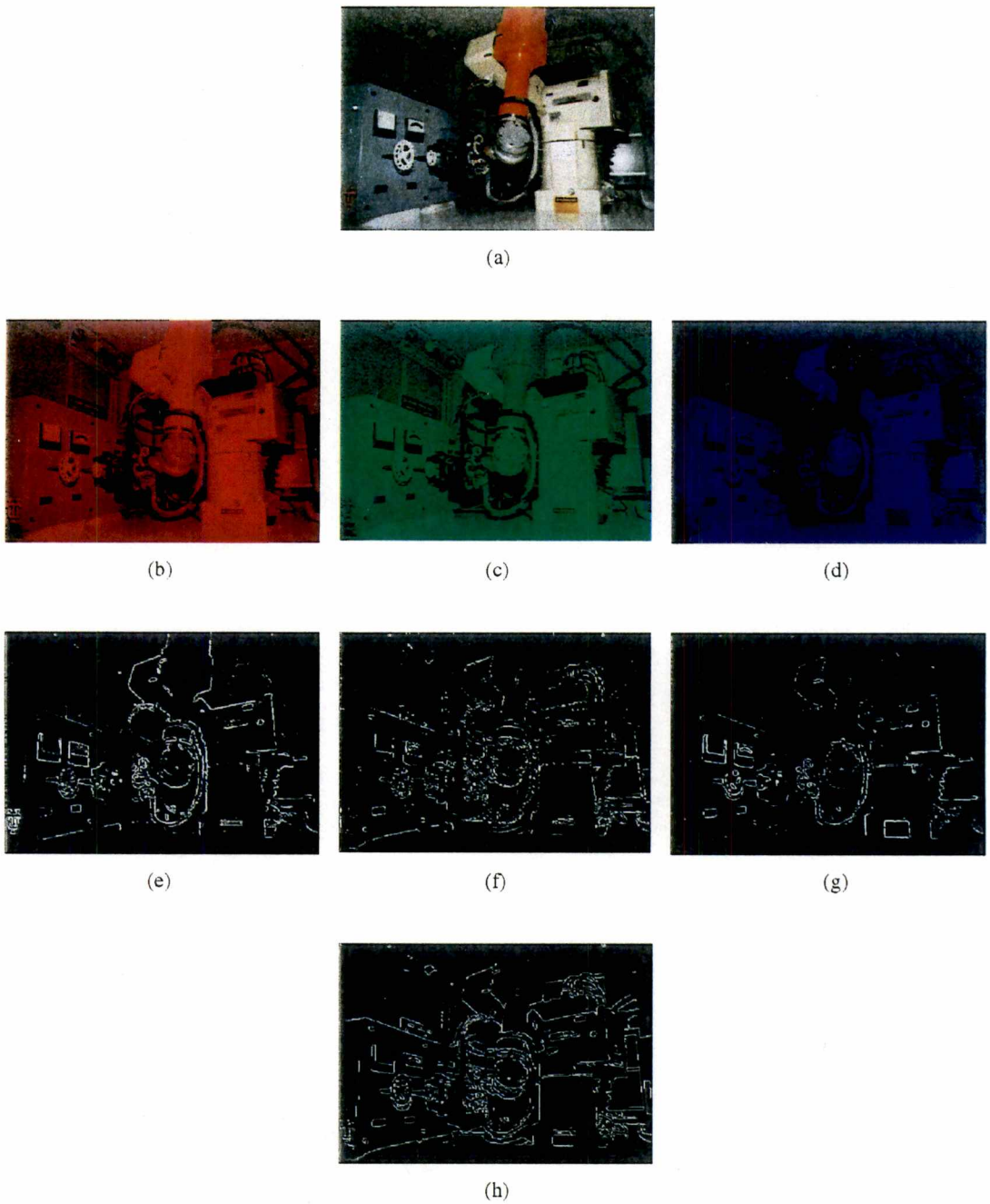


Figure 4.60: *Robotics manipulator; color edge detection: (a) input color image, (b) red channel, (c) green channel, (d) blue channel, (e) red edge map, (f) green edge map, (g) blue edge map, (h) fused edges.*

as a tool to show both the intermediate results and the final detected edges by the algorithm.

Figure 4.61(a) shows a 128×128 synthetic range image and corresponds to the input data to the algorithm. Figure 4.61(b) shows the vertical edges detected by the algorithm. Notice that the range image is not perfectly smooth to begin with, so edges not clearly visible at the resolution of the print were detected. Next, the input image was transposed by a simple matrix operation, resulting in Fig. 4.61(c). Then the vertical edges were extracted again (Fig. 4.61(d)). Transposing this edge map, we obtained the horizontal zero–crossings of the input image (Fig. 4.61(e)). Finally, the complete edge map is given by the union of the vertical and horizontal edge maps and is shown in Fig. 4.61(f).

4.7 Scale–Space for the First– and Second–Order Systems

In this section we present experiments relating the influence of the regularization parameter (λ) on the regularized responses of the first– and second–order stabilizers. For these experiments, we selected a single slice (row) of a 100×100 two–dimensional image. The input signal was contaminated with zero–mean Gaussian noise. Figure 4.62 shows the responses of the first–order system for two different values of λ.

Figure 4.63 shows the corresponding responses for the case of the second–order stabilizer. Figure 4.64 shows the input data, the first–order and the second–order

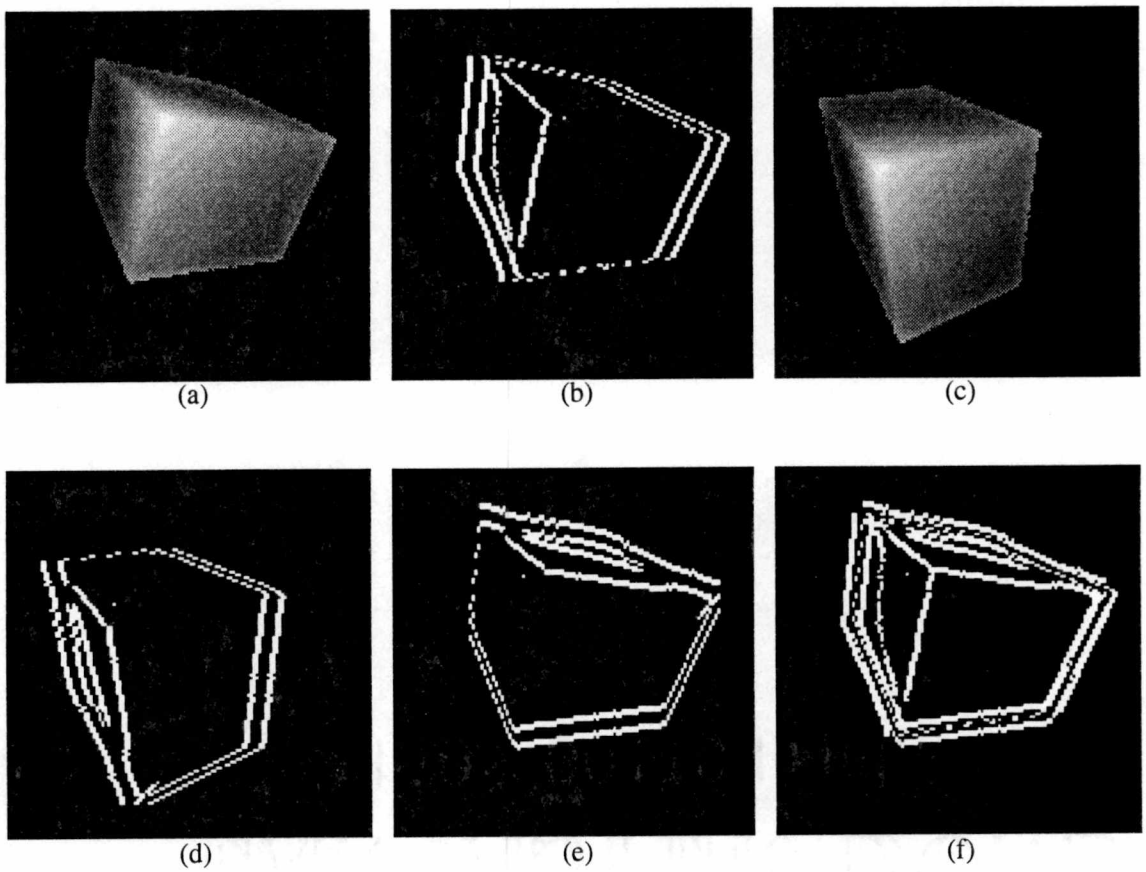


Figure 4.61: *Roof edge detection: (a) synthetic range input image, (b) vertical zero–crossings, (c) transposed of input image, (d) vertical zero–crossings of transposed image, (e) horizontal zero–crossings, (f) combined vertical and horizontal edges.*

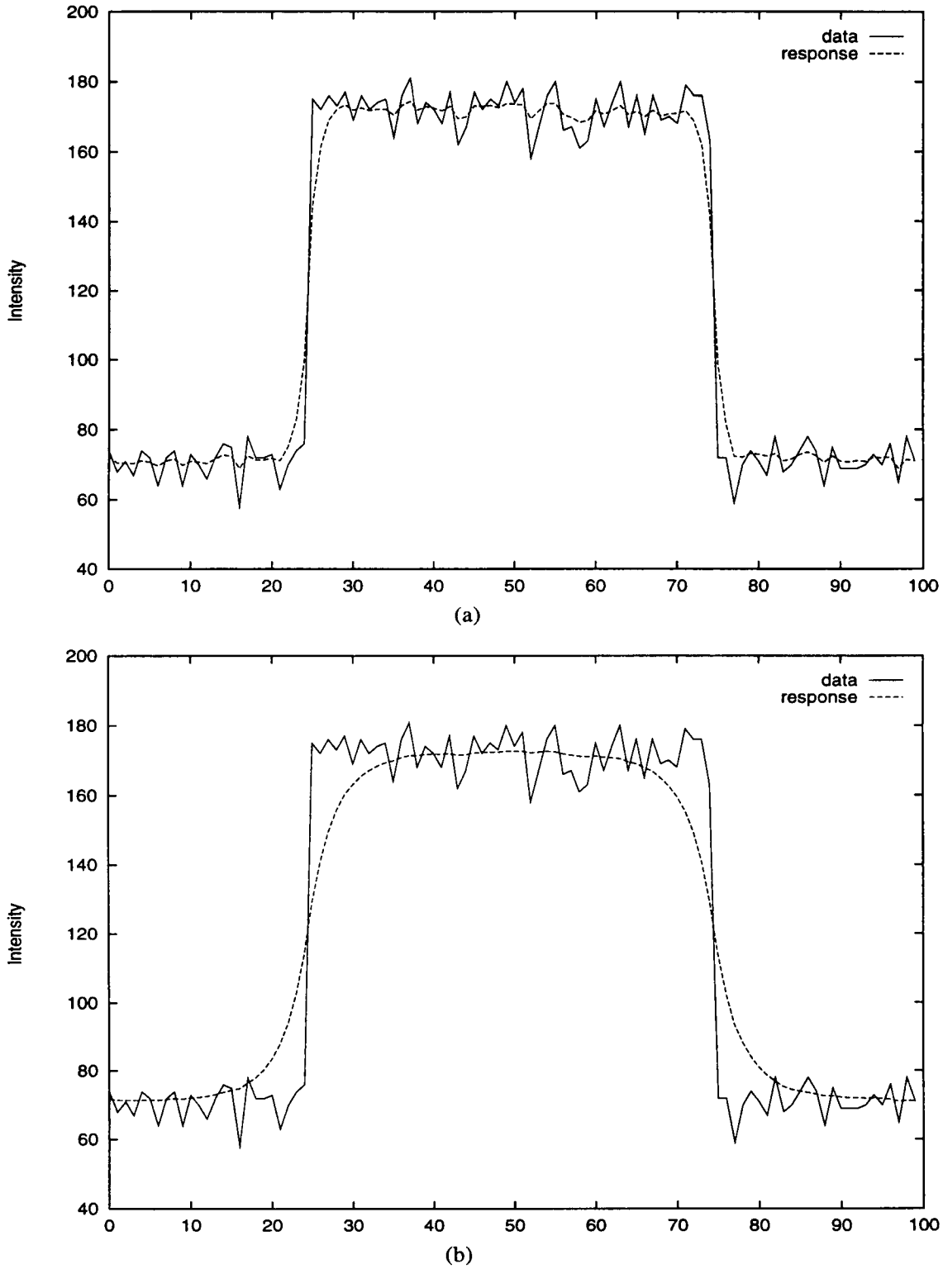


Figure 4.62: *First-order responses: (a)* $\lambda = 1$, *(b)* $\lambda = 10$.

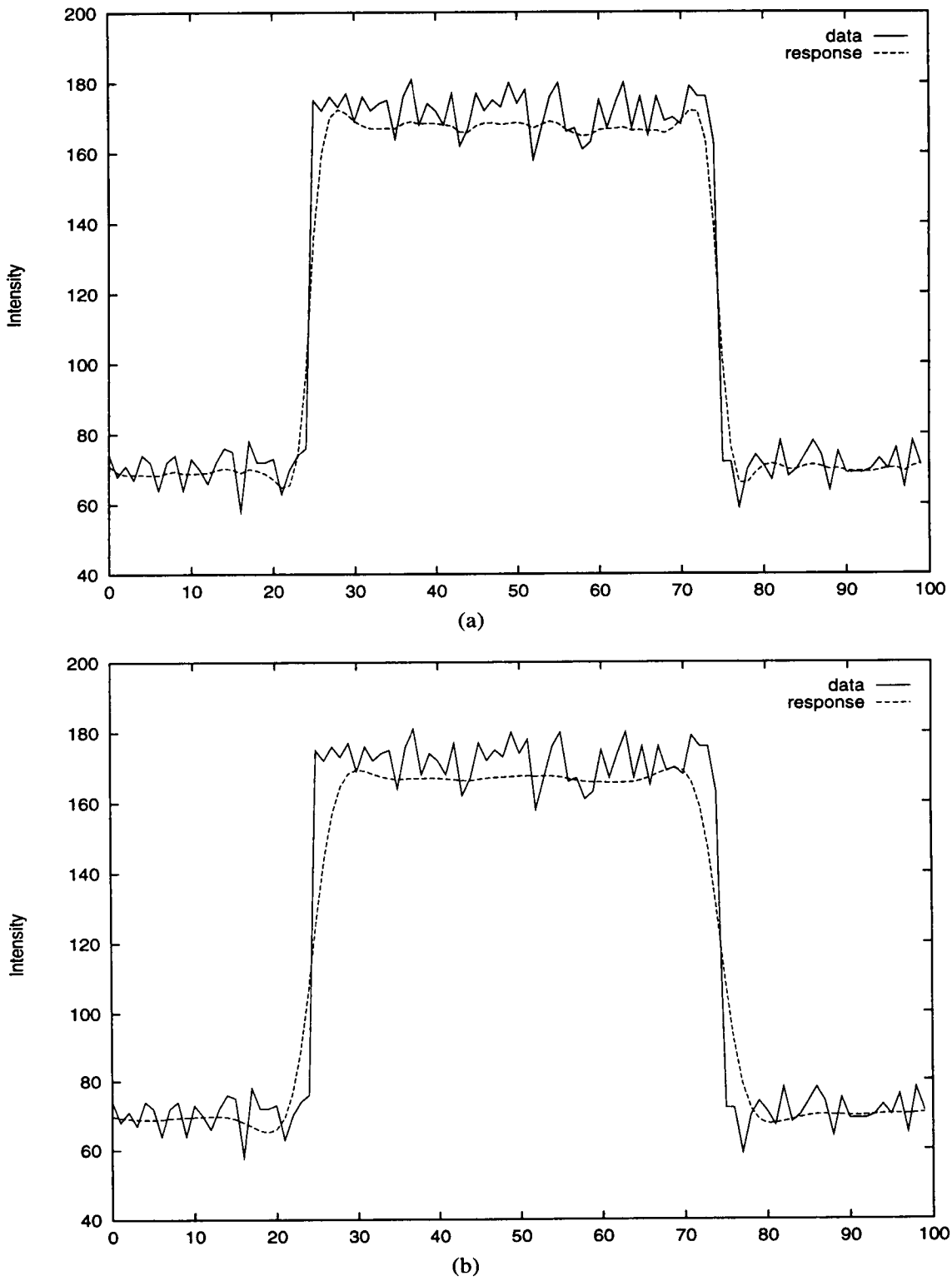


Figure 4.63: *Second-order responses: (a)* $\lambda = 1$, *(b)* $\lambda = 10$.

responses for the same value of the regularization parameter. In order to have a global appreciation of the behavior of the responses for an interval of λ, we need to plot in the same graph the responses of each system for a number of λs. In this representation, the horizontal coordinate (x–axis) represents space, the orthogonal axis in the plane (y–axis) represents λ, and the height over the x–y plane corresponds to the response (magnitude). Figures 4.65 and 4.66 show the scale–space representation for the first– and second–order systems, respectively.

4.8 Effect of Noise on the First– and Second–Order Systems

In this section we present experiments relating the effect of the noise (for a fixed value of the regularization parameter λ) on the regularized responses of the first– and second–order stabilizers. For these experiments, we selected a single slice (row) of a 100×100 two–dimensional image. The input signal was contaminated with zero–mean Gaussian noise. The value of λ was 5 and the standard deviation of the noise was varied from $\sigma_n = 10$ to $\sigma_n = 35$ in 5 units increments. We found that for σ_n beyond 35, the quality of the response decayed rapidly, making the edge detection procedure unreliable. We have used the usual definition of SNR $= 10\log(h/\sigma_n)$, where h is the magnitude of the step discontinuity. In terms of SNR, we tested the response for the following set: $\{10, 8.24, 6.99, 6.02, 5.23, 4.56\}$. Figure 4.67 shows the responses for the first–order stabilizer case. Figure 4.68 shows the responses for the second–order stabilizer case.

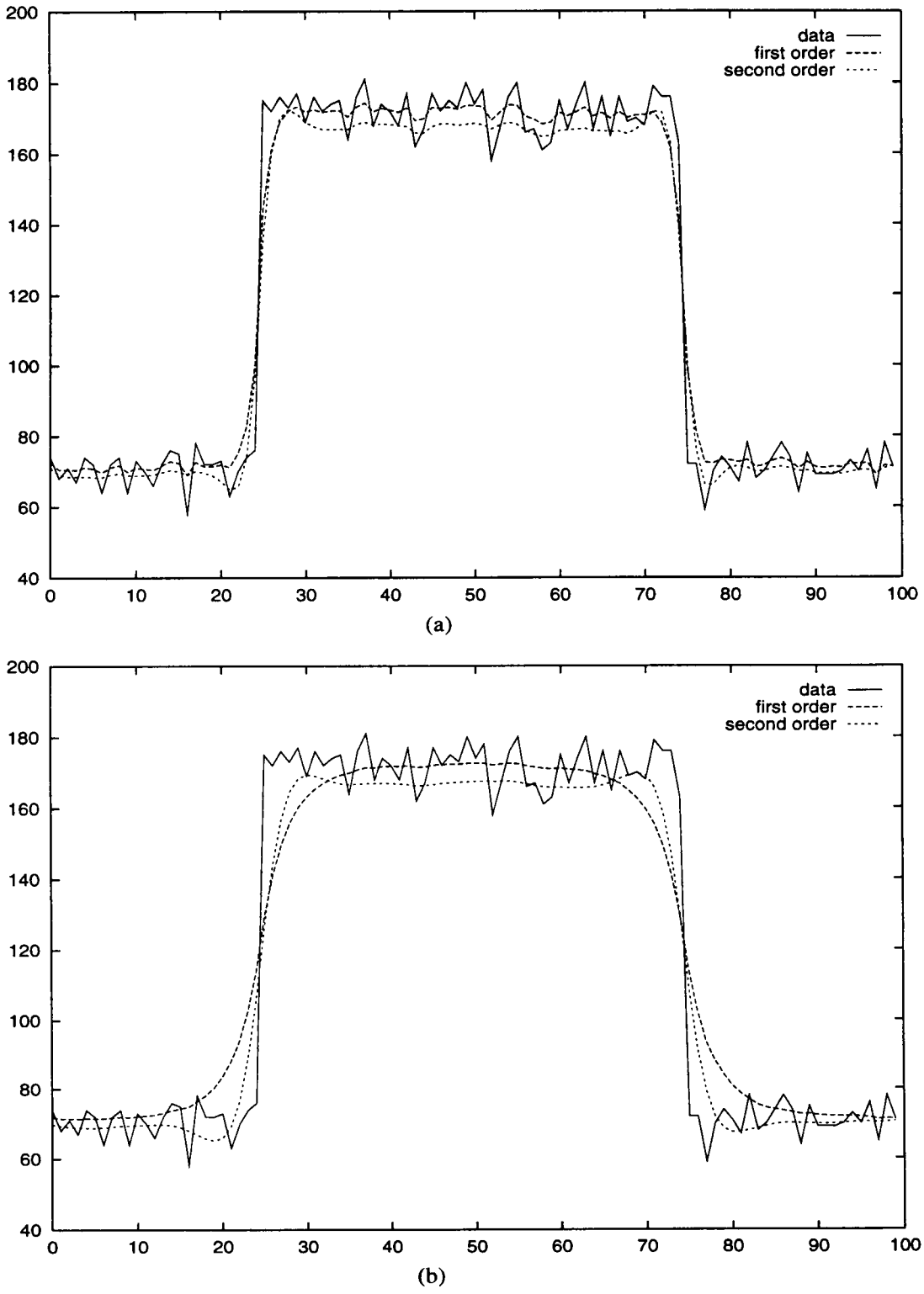


Figure 4.64: *First- and second-order responses: (a)* $\lambda = 1$, *(b)* $\lambda = 10$.

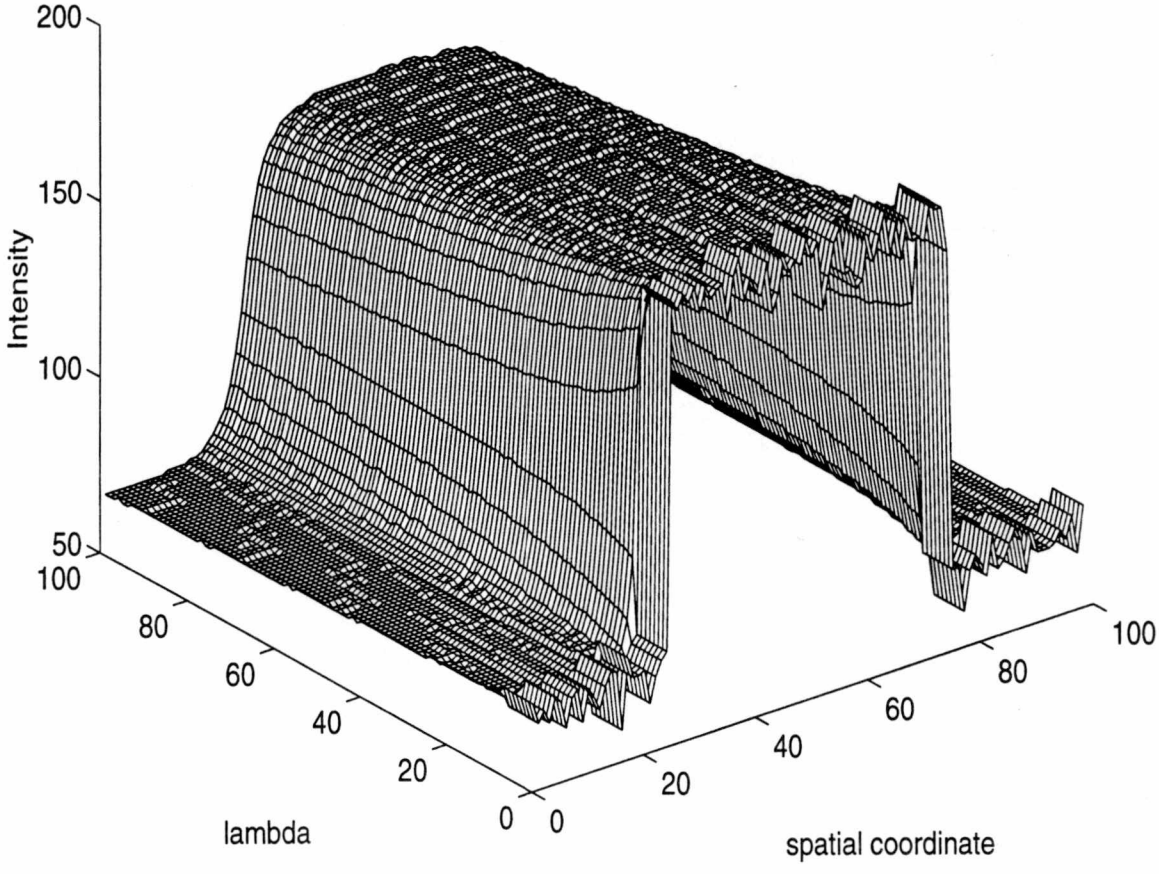


Figure 4.65: *Scale–space representation for the first–order stabilizer.*

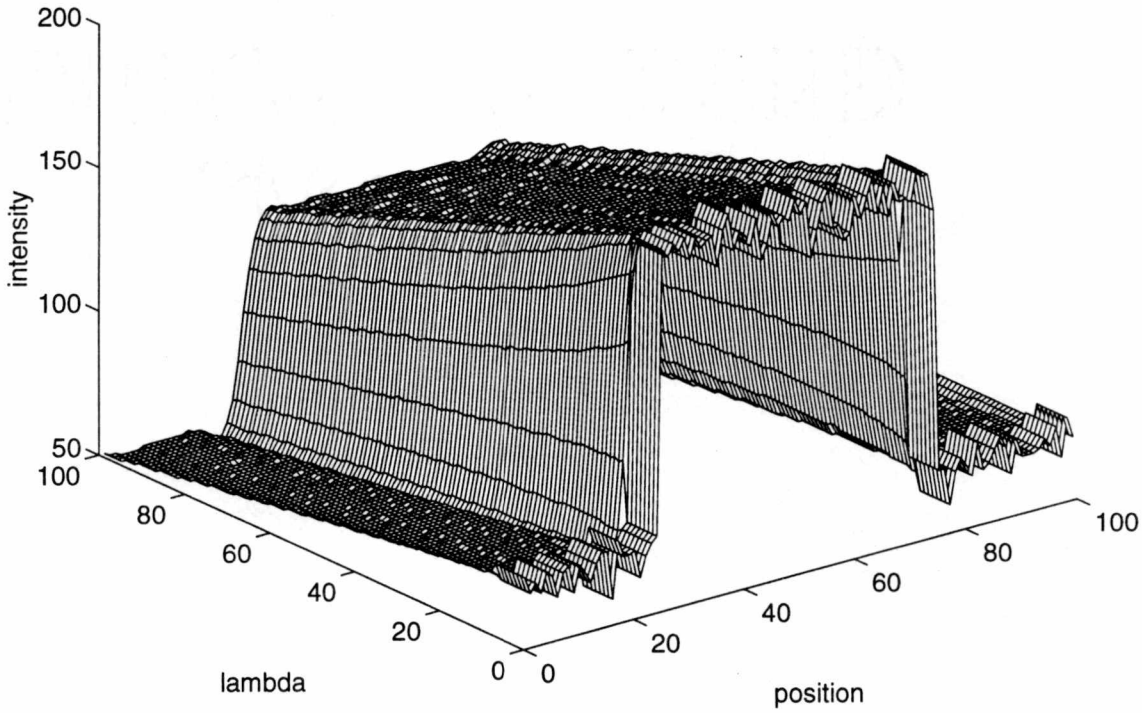


Figure 4.66: *Scale–space representation for the second–order stabilizer.*

An analysis of these plots indicates that for high SNR the second–order response produced better edge localization, whereas for low SNR the first–order response was a better approximation to the underlying noise–free input signal.

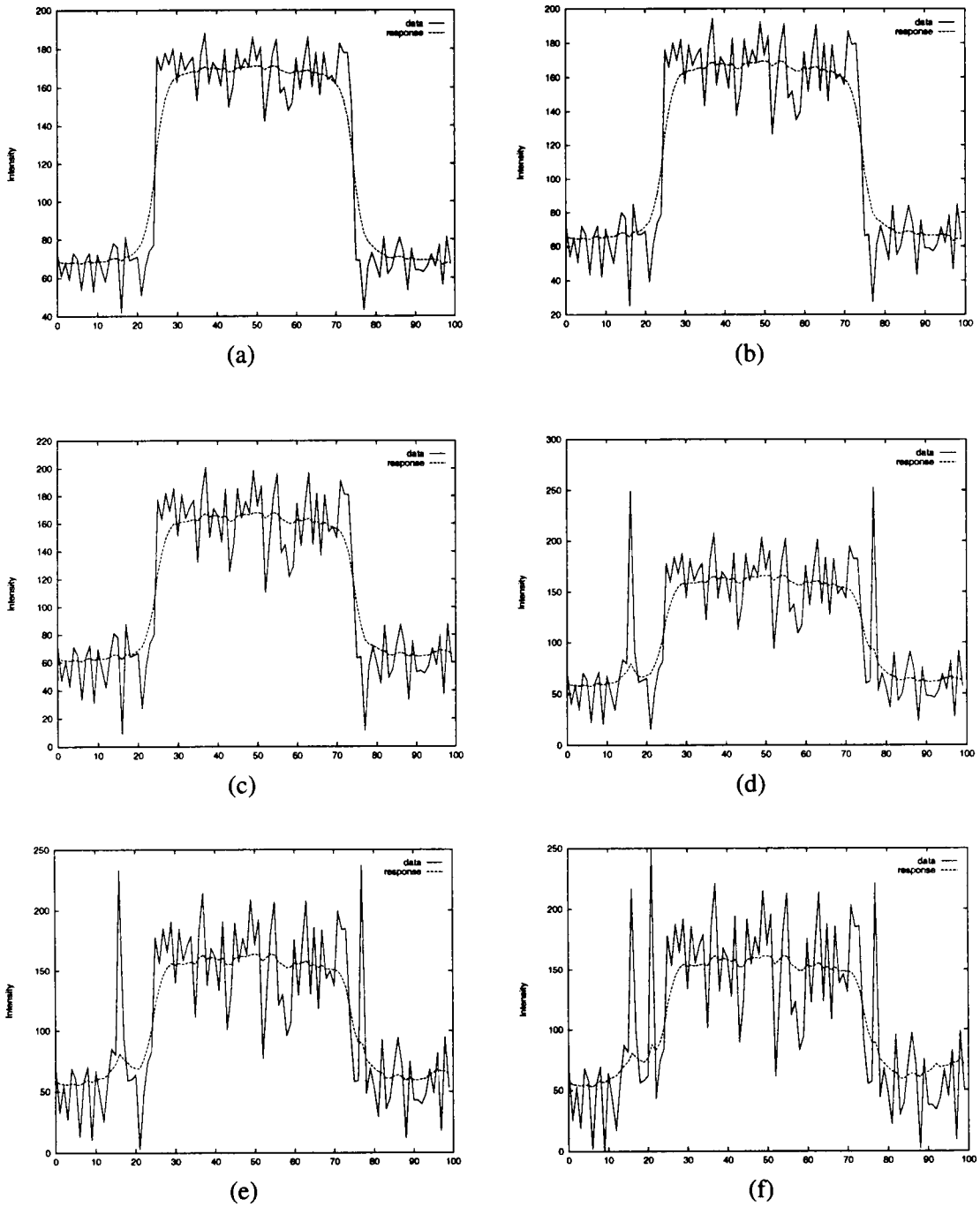


Figure 4.67: *First order stabilizer results under increasing noise levels: (a)* $\sigma_n = 10$, *(b)* $\sigma_n = 15$, *(c)* $\sigma_n = 20$, *(d)* , $\sigma_n = 25$ *(e)* $\sigma_n = 30$, *(f)* $\sigma_n = 35$.

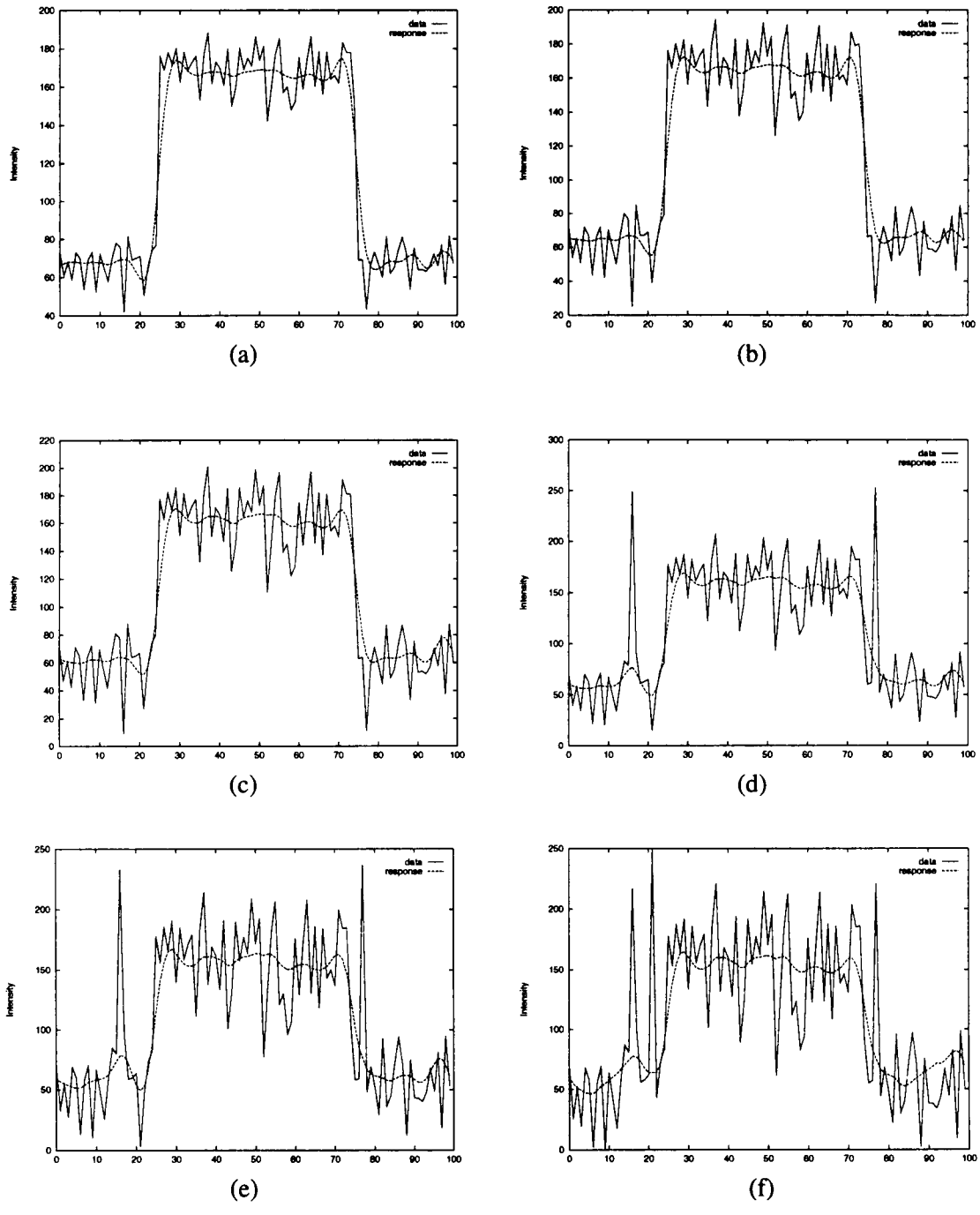


Figure 4.68: *Second order stabilizer results under increasing noise levels: (a)* $\sigma_n = 10$, *(b)* $\sigma_n = 15$, *(c)* $\sigma_n = 20$, *(d)* , $\sigma_n = 25$ *(e)* $\sigma_n = 30$, *(f)* $\sigma_n = 35$.

CHAPTER 5

CONCLUSION

This chapter presents a summary of the dissertation, the contributions which it makes to the areas of early vision processing and data fusion, and a list of recommendations for future research.

5.1 Summary and Contributions

Early vision is a necessary step prior to high–level vision processing. Data fusion can help in solving the inversion problem of finding adequate representations of the real world from unreliable sensory data. This happens because multiple data of the same or diverse sensing modalities provide additional constraints and complement each other and help in making the problem well–posed.

In this dissertation, three early vision problems have been considered: gray edge detection, color edge detection, and surface reconstruction. The unifying approach has been a new analytical data fusion technique that has been employed in all three cases.

In summary, work accomplished in this dissertation includes:

- A review of the basic concepts and categories of methods for data fusion,

- A review of previous work in data fusion applications, in roof edge detection, and in the three early vision problems mentioned before,
- An introduction to a new analytical fusion technique that can be solved employing Calculus of Variations and Euler–Lagrange techniques,
- A presentation of a family of functionals to solve the data fusion problem, as applied to the fusion at the level of edge features,
- A validation of the new formulation by applying it to various early vision problems, including gray edge detection, color edge detection, and surface reconstruction.

Among the contributions of this work, we can mention:

- A suite of regularized data fusion functionals that extends standard regularized techniques.
- Extensive testing on both synthetic and real images.
- Experiments using both fixed regularization parameters and space-variant regularization parameters.
- Implementations and tests for both first–order stabilizer (FOS) and second–order stabilizer (SOS) for 1–D and 2–D signals.
- A new roof edge detection algorithm has been proposed and tested.

The advantages of the new formulation, as compared to the analytical fusion technique described in Section 2.1 are the following:

- The new method provides a natural generalization to the multi–source case. The earlier analytical technique required pairwise fusion to achieve the same effect, therefore adding extra layers of processing before the final fused result.
- The new formulation does not require derivatives of the input data. This is especially important when dealing with noisy data, which is the usual case when data represents sensory information.
- The new method also allows for space–variant regularization parameters, thereby extending conventional regularization.

5.2 Recommendations for Future Research

We conclude this dissertation by mentioning several extensions to our work that may prove to be beneficial. We will list both general and specific extensions.

- Higher sensor data rates will require faster processing speeds. There are two natural ways to achieve this goal: faster hardware and optimized software. The trend towards dedicated VLSI processors, analog and/or digital, for early vision problems is an example of the hardware approach. Multigrid methods and direct Poisson solvers are two examples of the software approach.

- There are other mathematical models that deserve attention, including probabilistic models of random Markov fields, physical models of diffusion, and multi–scale wavelet representations.

- The fusion of multi–dimensional data, for example 1–D, 2–D, and 3–D, is a predictable extension of our work. Also, the fusion through time (e.g., using extended Kalman filters) needs to be considered.

- The use of specific knowledge about the sensors and about the application is another area that needs to be addressed. For example, median filtering is almost sure to improve range data. Although the mathematical model that performs the fusion does not need to include this kind of information, a complete sensor fusion system needs to be tuned to the proposed environment and application.

- Another area that needs attention is robust statistics because the assumption of Gaussian noise is too restrictive. Also, efficient methods for measuring noise variance from images would allow tighter control of various parameters.

- Among the specific extensions we can cite higher order stabilizers, additional penalty term(s), and nonlinear combinations of the data sources.

- The selection of the regularization parameter is an issue that deserves further study. For a single scalar parameter, cross–validation is a potential alternative. For space–variant parameters, perhaps an adaptive learning scheme could be implemented, but it may be impractical for real-time applications.

- Finally, the issues of viewpoint invariance and the concept of differential geometry should be addressed in the surface reconstruction problem.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] M. B. Abdulghafour. *Data Fusion Through Fuzzy Reasoning Applied to Feature Extraction from Multi-Sensory Images.* PhD thesis, The University of Tennessee, Knoxville, 1992.

[2] M.A. Abidi. Sensor fusion: A new approach and its applications. In *Proc. SPIE Sensor Fusion II: Human and Machine Strategies*, Vol. 1198, pp. 235–246, Philadelphia, PA, Nov. 6–9 1989.

[3] M.A. Abidi. Fusion of multi–dimensional data using regularization. In M.A. Abidi and R.C. Gonzalez, editors, *Data Fusion in Robotics and Machine Intelligence*, Chapter 10, pp. 415–455. Academic Press, 1992.

[4] E. Al-Hujazi and A. Sood. Range image segmentation with applications to robot bin–picking using a vacuum gripper. *IEEE Trans. on Systems, Man and Cybernetics*, Vol. 20, No. 6, pp. 1313–1324, Nov/Dec 1990.

[5] M. Baccar and M.A. Abidi. Towards a reliable segmentation of range images. In *SPIE Intl. Symp. on Intelligent Robotics and Computer Vision*, 1993.

[6] M. Beckerman. A Bayes–maximum entropy method for multisensor data fusion. In *Proc. Intl. Conf. on Robotics and Automation*, pp. 1668–1674, Nice, France, May 12–14 1992.

[7] G. Bilbro and W. Snyder. Fusion of range and reflectance image data using Markov random fields. In *Proc. IEEE Int. Symp. Intell. Control*, pp. 154–158, Arlington, VA, Aug. 1989.

[8] A. Blake and A. Zisserman. *Visual Reconstruction.* The MIT Press, 1987.

[9] M. Cebrián, M. Pérez-Luque, and G. Cisneros. Edge detection alternativas for multispectral remote sensing images. In *Proc. 8th Scandinavian Conference on Image Analysis*, pp. 1047–1054, Tromso, Norway, May 25–28 1993.

[10] D. Chen. *Function Representation and Approximation by Neural Networks.* PhD thesis, The University of Michigan, 1991.

[11] P. Chou and C. Brown. Multimodal reconstruction and segmentation with MRF and HCF optimization. In *Proc. Image Understanding Workshop*, Vol. I, pp. 214–221, Cambridge, MA, April 6–8 1988. DARPA.

[12] A. Cumani, P. Grattoni, and A. Guiducci. An edge–based description of color images. *CVGIP: Graphical Models and Image Processing*, Vol. 53, No. 3, pp. 13–323, July 1991.

[13] H. Cung, P. Cohen, and P. Boulanger. Multiscale edge detection and classification in range images. In *Proc. IEEE Conf. on Robotics and Automation*, pp. 2038–2044, Cincinnati, Ohio, May 1990.

[14] C.J. Delcroix and M.A. Abidi. Fusion of edge maps in color images. In *Proc. SPIE Visual Communications and Image Processing*, Nov. 9–11 1988.

[15] H. F. Durrant-Whyte. Consistent integration and propagation of disparate sensor observations. *Intl. J. Robot. Res.*, Vol. 6, No. 3, pp. 3–24, 1987.

[16] Tin-Jun Fan. *Describing and recognizing 3-D objects using surface properties.* Springer–Verlag, New York, 1990.

[17] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Trans. Pattern Analysis and Machine Intelligence*, PAMI-6, pp. 7221–7741, 1985.

[18] B. Gil, A. Mitiche, and J. Aggarwal. Experiments in combining intensity and range edge maps. *CVGIP*, Vol. 21, pp. 395–411, 1983.

[19] G. Godin and M. Levine. Structured edge map of curved objects in a range image. In *Proc. IEEE Conf. Comp. Vis. and Patt. Recog.*, pp. 276–281, Rosemont, IL, June 1989.

[20] R.C. Gonzalez, M. Baccar, and M.A. Abidi. Segmentation of range images via data fusion and morphological watersheds. In *Proc. 8th Scandinavian Conf. on Image Analysis*, pp. 21–39, Tromso, Norway, May 25–28 1993.

[21] W.E.L. Grimson. *From Images to Surfaces: A Computational Study of the Human Early Visual System.* The MIT Press, 1981.

[22] J. K. Hackett and M. Shah. Multisensor fusion: A perspective. In *Proc. IEEE Intl. Conf. on Robotics and Automation*, pp. 1324–1330, Cincinnati, Ohio, May 13–18 1990.

[23] J. Hadamard. *Le probléme de Cauchy et les équations aux dérivées partielles linéaires hyperboliques.* Hermann, Paris, 1932.

[24] J. Harris. A new approach to surface reconstruction: The coupled depth/slope model. In *IEEE First Intl. Conf. in Computer Vision*, pp. 277–283, London, England, June 8–11 1987.

[25] T.L. Huntsberger and M.F. Descalzi. Color edge detection. *Pattern Recognition Letters*, Vol. 3, pp. 205–209, May 1985.

[26] C. Jiang and M. Ward. Shadow identification. In *Proc. IEEE on Computer Vision and Pattern Recognition*, pp. 606–612, 1992.

[27] C. Koch, J. Marroquin, and A. Yuille. Analog "neuronal" networks in early vision. *Proc. Natl. Acad. Sci. USA*, Vol. 83, pp. 4263–4267, June 1986.

[28] E. Krotkov and R. Kories. Adaptive control of cooperating sensors: Focus and stereo ranging with an agile camera system. In *Proc. IEEE Intl. Conf. on Robotics and Automation*, Philadelphia, PA, April 24–29 1988.

[29] Hsien-Che Lee. Detecting boundaries in a vector field. *IEEE Trans. on Signal Processing*, Vol. 39, No. 5, pp. 1181–1194, May 1991.

[30] R.C. Luo and M.G. Kay. Multisensor integration and fusion in intelligent systems. *IEEE Trans. on Systems, Man, and Cybernetics*, Vol. 19, No. 5, pp.9901–9931, Sept/Oct 1989.

[31] R.C. Luo and M.G. Kay. Data fusion and sensor integration: State–of–the–art 1990s. In M.A. Abidi and R.C. Gonzalez, editors, *Data Fusion in Robotics and Machine Intelligence*, Chapter 2, pp. 7–135. Academic Press, 1992.

[32] R.C. Luo and M. Lin. Robot multisensor fusion and integration: optimum estimation of fused sensor data. In *Proc. IEEE Intl. Conf. Robotics and Automation*, pp. 1076–1081, Philadelphia, PA, Apr. 1988.

[33] B. Mathur, H. Wang, S. Liu, C. Koch, and J. Luo. Pixel level data fusion: From algorithm to chip. In *SPIE Conf. Visual Information Processing: From Neurons to Chips*, Vol. 1473, pp. 153–160, Orlando, FL, Apr. 1–2 1991.

[34] S. Muraki, N. Yokoya, and K. Yamamoto. Surface reconstruction from a contour line image by regularization. *Systems and Computers in Japan*, Vol. 33, No. 2, pp. 81–90, 1992.

[35] N. Nasrabadi, S. Clifford, and Y. Liu. Integration of stereo vision and otical flow by using an energy–minimization approach. *J. Opt. Soc. Am. A*, Vol. 6, No. 6, pp. 900–907, June 1989.

[36] R. Nevatia. A color edge detector and its use in scene segmentation. *IEEE Trans. on Systems, Man, and Cybernetics*, SMC–7, No. 11, pp. 820–826, November 1977.

[37] T. Poggio, E. Gamble, and J. Little. Parallel integration of vision modules. *Science*, Vol. 242, pp. 436–440, Oct. 1988.

[38] T. Poggio and F. Girosi. Networks for approximation and learning. *Proc. of the IEEE*, Vol. 78, No. 9, pp. 1481–1496, September 1990.

[39] T. Poggio, H. Voorhees, and A. Yuille. A regularized solution to edge detection. *Journal of Complexity*, Vol. 4, pp. 106–123, 1988.

[40] G.S. Robinson. Color edge detection. *Optical Engineering*, Vol. 16, No. 5, pp. 479–484, Sept/Oct 1977.

[41] M.C. Roggemann. *Multiple Sensor Fusion for Detecting Targets in FLIR and Range Images.* PhD thesis, Air Force Institute of Technology, Wright–Patterson AFB, OH, May 1990.

[42] S. Shaw, R. de Figueiredo, and K. Krish. Fusion of radar and optical sensors for space robotic vision. In *Proc. IEEE Intl. Conf. on Robotics and Automation*, Philadelphia, PA, April 24–29 1988.

[43] A. Shiozaki. Edge extraction using entropy operator. *CVGIP*, Vol. 36, pp. 1–9, 1986.

[44] R. Stevenson and E. Delp. Invariant reconstruction of curves and surfaces with discontinuities with applications in computer vision. Technical report, School of Electrical Engineering, Purdue University, 1990.

[45] D. Suter. *Cooperative Algorithms in Machine Vision.* PhD thesis, La Trobe University, Australia, 1990.

[46] R. Szeliski. Probabilistic modeling of surfaces. In *SPIE Conf. on Geometric Methods in Computer Vision*, 1991.

[47] D. Terzopoulos. The computation of visible–surface representations. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 10, No. 4, pp. 417–438, July 1988.

[48] S. C. A. Thomopoulos. Sensor integration and data fusion. In *Proc. SPIE Sensor Fusion II: Human and Machine Strategies*, Vol. 1198, pp. 178–191, Philadelphia, PA, Nov. 6–9 1989.

[49] A.N. Tikhonov and V.Y. Arsenin. *Solutions of Ill–Posed Problems.* Winston and Sons, Washington, DC., 1987.

[50] C. Tong, S. Rogers, J. Mills, and M. Kabrisky. Multisensor data fusion of laser radar and forward looking infrared (FLIR) for target segmentation. In *Proc. SPIE Infrared Sensors and Sensor Fusion*, Vol. 782, pp. 10–19, Orlando, FL, May 19–21 1987.

[51] P.E. Trahanias and A.N. Venetsanopoulos. Color edge detection using vector order statistics. *IEEE Trans. on Image Processing*, Vol. 2, No. 2, pp. 259–264, April 1993.

[52] S. Ullman and A. Sha'ashua. Structural saliency: The detection of globally salient structures using a locally connected network. Technical Report AI Memo 1061, Massachussetts Institute of Technology, July 1988.

[53] W.A. Wright. A Markov random field approach to data fusion and colour segmentation. *Image and Vision Computing*, Vol. 7, No. 2, pp. 144–150, May 1989.

[54] J. Yen. Generalizing the Dempster–Shafer theory to fuzzy sets. *IEEE Trans. on Systems, Man, and Cybernetics*, Vol. 20, No. 3, pp. 559–570, May/June 1990.

[55] N. Yokoya and M.D. Levine. Range image segmentation based on differential geometry: A hybrid approach. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 11, No. 6, pp. 643–649, June 1989.

VITA

Renato Alberto Salinas, was born in Temuco, Chile in 1950. He studied elementary education in the English School and secondary education at the Liceo de Hombres No. 1, both in Temuco. In 1967 he entered the State Technical University in Temuco with a major in Electrical Engineering, and in 1970 transfered to Santiago where he got his Electrical Engineer degree in 1974. The same year he accepted a position with the Electrical Engineering Department of the University of Santiago of Chile. In 1982 he was awarded a scholarship from the University of Santiago to pursue graduate studies in the University of Tennessee, Knoxville at the Electrical Engineering Department. He graduated with an M.S. in Electrical Engineering in 1985, and returned to his teaching position in Santiago. In 1988 he won a scholarship from the Organization of American States to work towards his Ph.D. in Electrical Engineering and was readmitted to the Electrical and Computer Engineering (ECE) Department of The University of Tennessee, Knoxville. In 1990 he was awarded a scholarship from the Chilean Government (Beca Presidente de la República). Mr. Salinas was a teaching assistant and later a research assistant with the ECE Department. He completed all the requirements for the degree of Doctor of Philosophy in August 1994. After graduation, Mr. Salinas shall return to Chile to pursue teaching and research in the areas of Digital Systems and Image Processing.