To the Graduate Council:

I am submitting herewith a thesis written by Ruyih Louis Wang entitled "RULE-BASED MULTI-TEMPLATE EDGE DETECTION AND THINNING ALGORITHM." I have examined the final copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Electrical Engineering.

Dragana Brzakovic, Major Professor

We have read this thesis
and recommend its acceptance:

M. A. Abidi

Mohan M. Trivedi

Accepted for the Council:

Vice Provost
and Dean of The Graduate School

STATEMENT OF PERMISSION TO USE

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

Requests for permission for extensive quotation from or reproduction of this thesis in whole or in parts may be granted by the copyright holder.

Signature Ruijie Wang

Date May, 1989

RULE-BASED MULTI-TEMPLATE EDGE DETECTION AND THINNING ALGORITHM

A Thesis

Presented for the

Master of Science

Degree

The University of Tennessee, Knoxville

Ruyih Louis Wang

August 1989

Copyright © Ruyih Louis Wang, 1989

All rights reserved

DEDICATION

This thesis is dedicated to my parents, Mr. Ping-wen Wang and Mrs. Feng-ying Lu Wang, and to my wife, Chiung-mei Yang Wang.

ACKNOWLEDGEMENTS

I would like to reveal my appreciation to my major advisor, Dr. Dragana Brzakovic, for her support, advice, and patience throughout my graduate career and this research. I also wish to thank Dr. M. A. Abidi and Dr. M. M. Trivedi for serving on my thesis committee and reviewing this thesis.

Special thanks are extended to the University of Tennessee, Knoxville, and the Department of Electrical and Computer Engineering for providing an excellent graduate study program and research environment and equipment. I am also grateful to the Martin Marietta Corporation for supporting with a graduate research assistantship through out this research.

I am extremely appreciative to my parents whose sturdy motivation and support let me achieve this academic goal. Finally, and most importantly, I would like to thank my wife, Chiung-mei Yang Wang, for her loving support and patient patronage throughout my graduate study period.

ABSTRACT

This thesis investigates edge detection in gray-scale digital images by considering both the edge strength and the edge direction information in the neighborhood of each edge pixel in an image. The advantage of the proposed algorithm is that it is not sensitive to noise and it can maintain small edge details in a noisy environment.

The proposed edge detection algorithm is rule-based and utilizes multi-template combinations. For each image pixel, a set of templates of different sizes is used to obtain the edge information. Then, the edge information, *i.e.*, the edge strengths and the edge directions, is processed using two rule-based criteria to produce the final edge strength and final edge direction of the image pixel. One criterion defines the magnitude relationships of qualified edge pixels. The other criterion defines the direction relationships of qualified edge pixels. Qualified edge pixels which pass both criteria constitute the edges of the given image. The performance of the proposed algorithm is examined using synthetic binary images and their various noisy versions and several infrared images.

TABLE OF CONTENTS

CHAPTER PAGE

1. INTRODUCTION 1

2. EDGE DETECTION 4

2.1 Edge Classification 5

2.2 Edge-oriented Segmentation 8

2.2.1 Edge Extraction 10

2.2.2 Edge Thinning 12

2.3 Result Improvement 14

3. PERFORMANCE OF DIRECTIONAL TEMPLATES IN EDGE DETECTION 15

3.1 Model of Ideal Binary Step Edge 16

3.2 Model of n Directional Templates 19

3.3 Detection of Edge Position and Direction 21

3.4 Relationship Between Correlation Coefficient and Template Size 26

3.5 Corner Edges 28

3.6 Two-Edge Regions 32

3.7 Effects of Noise 42

CHAPTER PAGE

4. RULE-BASED MULTI-TEMPLATE EDGE DETECTION AND THINNING 45

4.1 Three-Template Edge Detection Algorithm 49

4.1.1 Magnitude Relationship 51

4.1.2 Direction Relationship 54

4.1.3 Edge Strength and Direction Assignment 58

4.2 Four-Template Edge Detection Algorithm 58

4.2.1 Magnitude Relationship 59

4.2.2 Direction Relationship 60

4.2.3 Edge Strength and Direction Assignment 65

4.3 Postprocessing 65

4.4 Thinning Algorithm 67

4.4.1 LBE Assignment 68

4.4.2 Contour Following 70

4.4.3 Edge Refinement 73

5. EXPERIMENTAL RESULTS 74

5.1 Measurement of Quantitative Performance 74

5.1.1 Quantitative Measurements 75

5.1.2 Processing of Synthetic Binary Images 78

5.1.3 Comparison 98

5.2 Performance on Infrared Images 111

5.2.1 Comparison 137

CHAPTER PAGE

6. CONCLUSIONS 140

BIBLIOGRAPHY 142

APPENDIX 153

VITA 155

LIST OF TABLES

TABLE **PAGE**

3.1 Cross-correlation coefficients in the case when the secondary edge is much weaker than the main edge, i.e., $a - b \gg a - c$. The intensities are a=200, b=100, and c=195. The symbol m is the dimension of the subarea with intensity c as shown in Figure 3.12. 39

3.2 Cross-correlation coefficients in the case when the secondary edge is as strong as the main edge, i.e., $a - b \approx a - c$. The intensities are a=200, b=190, and c=190. (Both of the two edges are weak.) 40

3.3 Cross-correlation coefficients in the case when the secondary edge is as strong as the main edge, i.e., $a-b \approx a-c$. The intensities are a=200, b=100, and c=100. (Both of the two edges are of medium strength.) 40

3.4 Cross-correlation coefficients in the case when the secondary edge is as strong as the main edge, i.e., $a - b \approx a - c$. The intensities are a=200, b=10, and c=10. (Both of the two edges are strong.) 41

3.5 Cross-correlation coefficients in the case when the secondary edge is much stronger than the main edge, i.e., $a - b \ll a - c$. The intensities are a=200, b=190, and c=100. 41

4.1 The edge confidence assignment rules in the three-template algorithm. 57

TABLE PAGE

4.2 The edge confidence assignment rules in the four-template algorithm. 63

5.1 The quantitative performance of the rule-based multi-template edge detection algorithm on the concentric circles and squares in Figure 5.1. 85

5.2 The quantitative performance of the rule-based multi-template edge detection algorithm on the corners in Figure 5.2. 86

5.3 The quantitative performance of the rule-based multi-template edge detection algorithm on the concentric circles and squares with additive Gaussian noise of zero mean and standard deviation $\sigma_{noise} = 10$, as shown in Figure 5.7(a). 91

5.4 The quantitative performance of the rule-based multi-template edge detection algorithm on the corners with additive Gaussian noise of zero mean and standard deviation $\sigma_{noise} = 10$, as shown in Figure 5.7(b). 92

5.5 The quantitative performance of the rule-based multi-template edge detection algorithm on the concentric circles and squares with additive Gaussian noise of zero mean and standard deviation $\sigma_{noise} = 20$, as shown in Figure 5.10(a). 96

5.6 The quantitative performance of the rule-based multi-template edge detection algorithm on the corners with additive Gaussian noise of zero mean and standard deviation $\sigma_{noise} = 20$, as shown in Figure 5.10(b). 97

5.7 The quantitative performance of the 3 – 5 – 7 – 9 rule-based multi-template edge detection algorithm on the circles and squares with additive Gaussian noise of zero mean and standard deviation $\sigma_{noise} = 40$, 60, and 80 as shown in Figures 5.13(a), 5.13(b), and 5.14(b), respectively. 106

5.8 The quantitative performance of the 3 – 5 – 7 – 9 rule-based multi-template edge detection algorithm on the corners with additive Gaussian noise of zero mean and standard deviation $\sigma_{noise} = 40$, 60, and 80 as shown in Figures 5.15(a), 5.15(b), and 5.16(a), respectively. 107

5.9 The quantitative performance of the edge detection algorithm using one template size on the concentric circles, squares, and corners, as shown in Figures 5.1 and 5.2. 108

5.10 The quantitative performance of the edge detection algorithm using one template size on the concentric circles, squares, and corners with additive Gaussian noise of zero mean and standard deviation $\sigma_{noise} = 20$, as shown in Figure 5.10. 109

5.11 The quantitative performance of the edge detection algorithm using one template size on the concentric circles, squares, and corners with additive Gaussian noise of zero mean and standard deviation $\sigma_{noise} = 80$, as shown in Figures 5.14(a) and 5.16(a). . 110

LIST OF FIGURES

FIGURE **PAGE**

2.1 Ideal one-dimensional edge models. (a) ideal step edge. (b) ideal roof edge. (c) ideal line edge. (d) ideal spike edge. (e) ideal ridge edge (or ideal ramp edge). 7

3.1 Positioning of a directional template on the image pixel $I(i,j)$ of an image. 16

3.2 An edge pixel $I(i,j)$ located on the edge forming angle θ_1 with the x-axis and translated by $-h$ from the center of the image $I(x,y)$. 17

3.3 The model of an $M \times N$ binary image $I(x,y)$ with intensities a and b and a step edge at angle θ_1 relative to the x-axis and displaced by $-h$ from the center. 18

3.4 The model of a $p \times q$ directional template $T(x,y)$ with edge angle θ_2 relative to the x-axis and template coefficients λ and $-\lambda$. . . . 20

3.5 Computation of the cross-correlation coefficient of an $M \times N$ image $I(x,y)$ and a $p \times q$ directional template $T(x',y')$ at a specific image pixel $I(i,j)$. 22

3.6 The directional template, edge subimage, and their overlaps with size $p \times q$ at the image pixel $I(i,j)$. (a) The directional template $T(x,y)$. (b) The edge subimage. (c)(d) The overlaps of (a) and (b) with different notation. 23

3.7 Directional template and edge subimage with vertical edge and common size $(2k\Delta p + 1) \times (2k\Delta q + 1)$. (a) Directional template with template coefficients λ and $-\lambda$. (b) Edge subimage with intensities a and b and the center pixel at $I(i, j)$. 27

3.8 The overlaps of directional template and right-angled corner edge with size $p \times q$. The angle θ_2 is the angle between the template edge and the x-axis. (a) $0 \leq \theta_2 \leq \tan^{-1}(p - 2d)/p$. (b) $\tan^{-1}(p - 2d)/p < \theta_2 < \tan^{-1} q/p$. (c) $\tan^{-1} q/p \leq \theta_2 \leq \pi/2$. 29

3.9 The best match of a strong right-angled edge and a directional template with vertical edge. 33

3.10 The strong right-angled edge (dashed line) and its rounding-off resulting edge (solid line). 33

3.11 The directional template, edge subimage, and their overlaps with size $p \times q$ at the image pixel $I(i, j)$. (a) The directional template $T(x, y)$. (b) The edge subimage contains two edges with intensities a, b, and c. (c)(d) The overlaps of (a) and (b). 34

3.12 The arrangement of a subimage containing two edges and two directional templates with sizes $(2k\Delta p + 1) \times (2k\Delta q + 1)$ and $(2(k + 1)\Delta p + 1) \times (2(k + 1)\Delta q + 1)$. 36

4.1 The edge orientation system. 47

4.2 The edge direction assignment scheme. 48

4.3 The four directional templates of template $T_5(x, y)$ for detecting the edges with directions $\mathcal{D}$ = 1, 2, 3, and 4 if the products are positive or $\mathcal{D}$ = 5, 6, 7, and 8 if negative. (a) $T_{5,1}(x, y)$. (b) $T_{5,2}(x, y)$. (c) $T_{5,3}(x, y)$. (d) $T_{5,4}(x, y)$. 50

4.4 The construction of two concentric directional templates with adjacent template sizes and common edge direction $\mathcal{D} = 1$. (a) Template sizes 3 × 3 and 5 × 5 pixels. (b) Template sizes 5 × 5 and 7 × 7 pixels. (c) Template sizes 7 × 7 and 9 × 9 pixels. 53

4.5 The assignment of $\mathcal{S}_l$, $\mathcal{S}_0$, and $\mathcal{S}_g$ to the edge pixel $I(i, j)$. 67

4.6 Examples of counters u and v measurement at the image pixel $I(i, j)$. (a) The edge pixel $I(i, j)$ centered at a 3 × 3 subimage. (b) The edge strength of the pixels in (a). (c)(d) The increment of counters u and v when the directions of the edge pixel $I(i, j)$ are $\mathcal{D}$ = 1 and 8, respectively. 69

4.7 The definition of l, a, and r of the edge pixel $I(i, j)$ under the positive orientation process with the directions $\mathcal{D}$ from 1 to 8. . . 72

5.1 The 128 × 128 artificial image with binary intensities 100 and 200 in the form of two concentric circles (radii 10 and 25 pixels) and two squares (dimensions 10 × 10 and 25 × 25 pixels). 79

5.2 The 128 × 128 artificial image with binary intensities 100 and 200 which contains corner edges with angles ranging from 15° to 165° in 15° increments. 80

5.3 The ideal edges of Figure 5.1. 81

5.4 The ideal edges of Figure 5.2. 82

5.5 The detected edge pixels of Figure 5.1 using the proposed multi-template edge detection algorithm. (a) Figure 5.1. (b) Result of the 3 – 5 – 7 templates. (c) Result of the 5 – 7 – 9 templates. (d) Result of the 3 – 5 – 7 – 9 templates. 83

5.6 The detected edge pixels of Figure 5.2 using the proposed multi-template edge detection algorithm. (a) Figure 5.2. (b) Result of the 3 – 5 – 7 templates. (c) Result of the 5 – 7 – 9 templates. (d) Result of the 3 – 5 – 7 – 9 templates. 84

5.7 (a)(b) The noisy versions of Figures 5.1 and 5.2, respectively. The injected noise is additive Gaussian noise with zero mean and standard deviation of 10 percent original intensity range ($\sigma_{noise} = 10$). 88

5.8 The detected edge pixels of Figure 5.7(a) using the proposed multi-template edge detection algorithm. (a) Figure 5.7(a). (b) Result of the 3 – 5 – 7 templates. (c) Result of the 5 – 7 – 9 templates. (d) Result of the 3 – 5 – 7 – 9 templates. 89

5.9 The detected edge pixels of Figure 5.7(b) using the proposed multi-template edge detection algorithm. (a) Figure 5.7(b). (b) Result of the 3 – 5 – 7 templates. (c) Result of the 5 – 7 – 9 templates. (d) Result of the 3 – 5 – 7 – 9 templates. 90

5.10 (a)(b) The noisy versions of Figures 5.1 and 5.2, respectively. The injected noise is additive Gaussian noise with zero mean and standard deviation of 20 percent original intensity range (σ_{noise} = 20). 93

5.11 The detected edge pixels of Figure 5.10(a) using the proposed multi-template edge detection algorithm. (a) Figure 5.10(a). (b) Result of the 3 – 5 – 7 templates. (c) Result of the 5 – 7 – 9 templates. (d) Result of the 3 – 5 – 7 – 9 templates. 94

5.12 The detected edge pixels of Figure 5.10(b) using the proposed multi-template edge detection algorithm. (a) Figure 5.10(b). (b) Result of the 3 – 5 – 7 templates. (c) Result of the 5 – 7 – 9 templates. (d) Result of the 3 – 5 – 7 – 9 templates. 95

5.13 (a)(b) The noisy versions of Figure 5.1. The injected noise are additive Gaussian noise with zero mean and standard deviation of 40 and 60 percent original intensity range (σ_{noise} = 40 and 60). (c)(d) The results of (a) and (b), respectively, using the 3 – 5 – 7 – 9 templates. 102

5.14 (a) The noisy version of Figure 5.1. The injected noise is additive Gaussian noise with zero mean and standard deviation of 80 percent original intensity range (σ_{noise} = 80). (b) The result of (a) using the 3 – 5 – 7 – 9 templates. 103

5.15 (a)(b) The noisy versions of Figure 5.2. The injected noise are additive Gaussian noise with zero mean and standard deviation of 40 and 60 percent original intensity range (σ_{noise} = 40 and 60). (c)(d) The results of (a) and (b), respectively, using the 3 – 5 – 7 – 9 templates. 104

5.16 (a) The noisy version of Figure 5.2. The injected noise is additive Gaussian noise with zero mean and standard deviation of 80 percent original intensity range (σ_{noise} = 80). (b) The result of (a) using the 3 – 5 – 7 – 9 templates. 105

5.17 An infrared image of size 256 × 256 pixels and intensities from 0 to 255. 113

5.18 The ideal segmentation of image in Figure 5.17. 114

5.19 The edge image of Figure 5.17 using the 3 – 5 – 7 algorithm. . . 115

5.20 (a) The one-pixel-wide edge of Figure 5.19. (b) The result of (a) after edge refinement procedure with threshold TSL = 20 pixels. 116

5.21 The edge image of Figure 5.17 using the 5 – 7 – 9 algorithm. . . 117

5.22 (a) The one-pixel-wide edge of Figure 5.21. (b) The result of (a) after edge refinement procedure with threshold TSL = 20 pixels. 118

5.23 The edge image of Figure 5.17 using the 3 – 5 – 7 – 9 algorithm. 119

5.24 (a) The one-pixel-wide edge of Figure 5.23. (b) The result of (a) after edge refinement procedure with threshold TSL = 20 pixels. 120

5.25 An infrared image of size 256 × 256 pixels and intensities from 0 to 255. 121

5.26 The ideal segmentation of image in Figure 5.25. 122

5.27 The edge image of Figure 5.25 using the 3 – 5 – 7 algorithm. . . 123

5.28 (a) The one-pixel-wide edge of Figure 5.27. (b) The result of (a) after edge refinement procedure with threshold TSL = 20 pixels. 124

5.29 The edge image of Figure 5.25 using the 5 – 7 – 9 algorithm. . . 125

5.30 (a) The one-pixel-wide edge of Figure 5.29. (b) The result of (a) after edge refinement procedure with threshold TSL = 20 pixels. 126

5.31 The edge image of Figure 5.25 using the 3 – 5 – 7 – 9 algorithm. 127

5.32 (a) The one-pixel-wide edge of Figure 5.31. (b) The result of (a) after edge refinement procedure with threshold TSL = 20 pixels. 128

5.33 An infrared image of size 256 × 256 pixels and intensities from 0 to 255. 129

5.34 The ideal segmentation of image in Figure 5.33. 130

5.35 The edge image of Figure 5.33 using the 3 – 5 – 7 algorithm. . . 131

5.36 (a) The one-pixel-wide edge of Figure 5.35. (b) The result of (a) after edge refinement procedure with threshold TSL = 20 pixels. 132

5.37 The edge image of Figure 5.33 using the 5 – 7 – 9 algorithm. . . 133

5.38 (a) The one-pixel-wide edge of Figure 5.37. (b) The result of (a) after edge refinement procedure with threshold TSL = 20 pixels. 134

5.39 The edge image of Figure 5.33 using the 3 – 5 – 7 – 9 algorithm. 135

5.40 (a) The one-pixel-wide edge of Figure 5.39. (b) The result of (a) after edge refinement procedure with threshold TSL = 20 pixels. 136

5.41 The comparison of the results of the 3 – 5 – 7 and the 3 – 5 – 7 – 9 combination algorithms. 138

LIST OF SYMBOLS

English Symbols

A	subarea of a directional template
a	intensity of image
b	intensity of image
C_{max}	maximum value candidate in the contour following procedure
$C_1 \sim C_5$	constants
c	intensity of image
c_0	constant
$\mathcal{D}$	direction number ($1 \sim 8$)
d	distance between a corner and the center of a directional template
E	expected value
$\mathcal{E}$	edge confidence
$G[I(x,y)]_{(i,j)}$	gradient of an image $I(x,y)$ on a specific pixel $I(i,j)$
G_x	gradient along the x-axis
G_y	gradient along the y-axis
$G_1 \sim G_5$	subareas of a directional template

h	horizontal translation of an edge from the center of a subimage
$I(x, y)$	image
(i, j)	a specific image pixel
k	constant
LBE	likelihood of being an edge ($0 \sim 1$)
l	constant
M	dimension of image along the x-axis
$MAX1 \sim MAX4$	magnitude values produced by four specific template sizes
N	dimension of image along the y-axis
$N(0; \sigma^2)$	additive Gaussian noise with zero mean and variance σ^2
n	direction number of the orientation system
p	dimension of template along the x-axis
q	dimension of template along the y-axis
T_{ec}	threshold of the edge confidence
T_{edge}	threshold of the edge magnitude increment
$T(x, y)$	directional template
TSL	threshold of segment length
U	numerator of $\gamma(i, j)$
$u(x, y)$	counter of the LBE assignment

V	denominator of $\gamma(i,j)$
$v(x,y)$	counter of the LBE assignment
$x_1 \sim x_4$	pixels on the x-axis
(x,y)	an image pixel

Greek Symbols

α	intensity of image
α_{edge}	minimum intensity difference of two pixels lie across an edge
β	intensity of image
Γ_1	function of relationship between the x and y coordinate of an image pixel
Γ_2	function of relationship between the x and y coordinate of a template pixel
γ_1	cross-correlation coefficient on a pixel using $(2\Delta p+1)\times(2\Delta q+1)$ template $(k=1)$
γ_2	cross-correlation coefficient on a pixel using $(4\Delta p+1)\times(4\Delta q+1)$ template $(k=2)$
$\gamma(i,j)$	cross-correlation coefficient on a specific image pixel (i,j)
Δp	increment between two adjacent templates along the x-axis

Δq	increment between two adjacent templates along the y-axis
$\Delta\theta$	angle between two adjacent directions in the orientation system
$\Delta\rho$	angle between an image edge and a template edge
θ_1	angle between an image edge and the x-axis
θ_2	angle between a template edge and the x-axis
λ	coefficient of directional template
$-\lambda$	coefficient of directional template
τ_1, τ_2	subareas of a directional template
ω	constant ($\omega = d/p$)

Subscripts

k_1, k_2	directional template with size $k_1 \times k_1$ pixels and edge direction $\mathcal{D} = k_2$

CHAPTER 1

INTRODUCTION

In the last few decades, computers have assisted man in his attempt to improve the quality of life by their significant contributions in the area of automation. Some examples are robot navigation, target tracking, automated manufacturing, biomedical application, character recognition, remote sensing, and satellite picture analysis. In computer vision, edge detection, *i.e.*, isolation of meaningful regions in an image, is the first and most important step in determining final image understanding. The objective of this thesis is to develop a rule-based multi-template edge detection algorithm which extracts edge pixels from noisy images to produce edge images. The implementation is carried out in the spatial domain.

The multi-template algorithm combines templates of various template sizes. Each template size has four directional templates which can be used to detect eight directions. In general, utilization of multiple templates provides for superior performance of an edge detector as reported in [82]. The large templates minimize the influence of noise and coarse textures; however, they reduce the accuracy of the edge localization. On the other hand, the small templates are well suited for edge localization but are sensitive to noise and coarse textures.

This thesis presents the theoretical foundation, rules, and performance examination of the proposed algorithm. A review of the edge detectors proposed within the last two decades and the edge classification are described in Chapter 2. The review focuses on the edge-oriented detectors. At the end of Chapter 2, a brief review of thinning algorithms, which includes binary and gray-scale thinning algorithms, is presented.

Chapter 3 defines directional templates and examines their performance. In order to confirm the detection of the positions and directions of edge pixels by directional templates, several specific edge configurations are used: single edges, double edges, and corner edges. For the purpose of examining the performance of directional templates in cases where noise is present, in Chapter 3, we also considers edges which are injected with additive Gaussian noise.

In Chapter 4, the details of the rule-based multi-template edge detection algorithm are discussed. The main body of Chapter 4 is organized into two major parts. The first deals with the three-template combinations and the second discusses the four-template combination. In each of these parts, there are two sets of rules which govern the magnitude and the direction relationships of each image pixel. Section 4.3 describes a postprocessing method, which improves the strength and the direction relationships of the detected edge pixels. Finally, Section 4.4 presents a thinning algorithm which skeletonizes the detected edges to one-pixel-wide edges.

The first section of Chapter 5 evaluates the quantitative performance of the proposed algorithm by testing this algorithm using two synthetic binary images with and without additive Gaussian noise. Also, in this section performance of the proposed algorithm is compared to performance of edge detectors that use a single size templates. In the second section, the results of processing infrared images are discussed. The last chapter, Chapter 6, summarizes the conclusions.

CHAPTER 2

EDGE DETECTION

In computer vision, objects are the elementary components used by computers to describe the contents of a digital image. Therefore, image segmentation, *i.e.*, isolation of individual objects, is an important process in image understanding. In digital image processing, two major approaches are used to achieve image segmentation. These approaches are referred to as the edge-oriented and the region-oriented approaches [56,106]. The objective of the edge-oriented approach is to detect local intensity discontinuities. In general, the edge-oriented detectors can not guarantee the continuity of resulting edges. In the region-oriented approach, the basis of segmentation is pixel similarity. In this approach, image pixels with common properties are identified and grouped to generate regions. Therefore, in general, complete segmentation can be achieved by using region-oriented detectors. Algorithms like thresholding [68,74,108,109,110], region growing [27,28,38,52,63,112], and split-and-merge [18,44,45] are well-known region-oriented methods.

The algorithm presented in this thesis belongs to the edge-oriented approach. Therefore, in this chapter, we focus our review on the edge-oriented detectors. However, before reviewing the edge-oriented detectors, this chapter discusses the classification of edges in gray-scale images.

2.1 Edge Classification

In gray-scale images, edges are defined as a change in intensity. According to the type of intensity change, edges can be classified as step edges, roof edges, line edges, spike edges, and ridge edges (or ramp edges) [14,21]. The ideal models of these edges in the one-dimensional case are:

1. **Step edge :**

$$f(x) = \begin{cases} \alpha & \text{if } x < x_1 \\ \beta & \text{if } x \geq x_1, \end{cases} \tag{2.1}$$

2. **Roof edge :**

$$f(x) = \begin{cases} \alpha \cdot \frac{x}{x_1} & \text{if } x < x_1 \\ \alpha \cdot \frac{x_2 - x}{x_2 - x_1} & \text{if } x_1 \leq x \leq x_2 \\ 0 & \text{if } x > x_2, \end{cases} \tag{2.2}$$

3. **Line edge :**

$$f(x) = \begin{cases} \alpha & \text{if } x < x_1 \\ \beta & \text{if } x_1 \leq x \leq x_2 \\ \gamma & \text{if } x > x_2, \end{cases} \tag{2.3}$$

4. **Spike edge :**

$$f(x) = \begin{cases} \alpha & \text{if } x < x_1 \\ c_0 \cdot [(x_2 - x)(x - x_1)]^{1/2} + \alpha & \text{if } x_1 \leq x \leq x_2 \\ \alpha & \text{if } x > x_2, \end{cases} \tag{2.4}$$

5. **Ridge edge :**

$$f(x) = \begin{cases} \alpha & \text{if } x < x_1 \\ (\beta - \alpha) \cdot \frac{x - x_1}{x_2 - x_1} + \alpha & \text{if } x_1 \leq x \leq x_2 \\ \beta & \text{if } x_2 < x < x_3 \\ (\gamma - \beta) \cdot \frac{x - x_3}{x_4 - x_3} + \beta & \text{if } x_3 \leq x \leq x_4 \\ \gamma & \text{if } x > x_4, \end{cases} \tag{2.5}$$

where $f(x)$ is the intensity at point x; α, β, and γ are intensities; and c_0 is a constant. These ideal edge models are shown in Figure 2.1.

In general, each of these edges requires a special purpose detector [76]. Some authors [17,76], in the last few years, developed the multi-purpose algorithms for some of these edge models. For instance, Canny [17] developed an algorithm which is able to detect step edges, roof edges, and ridge edges, and Nalwa [76] detected line edges with a step edge detector by assuming that two locally parallel straight step edges constituted a line edge. The primary interest of this thesis lies with the step edge detectors; consequently, further discussion focuses on these detectors.

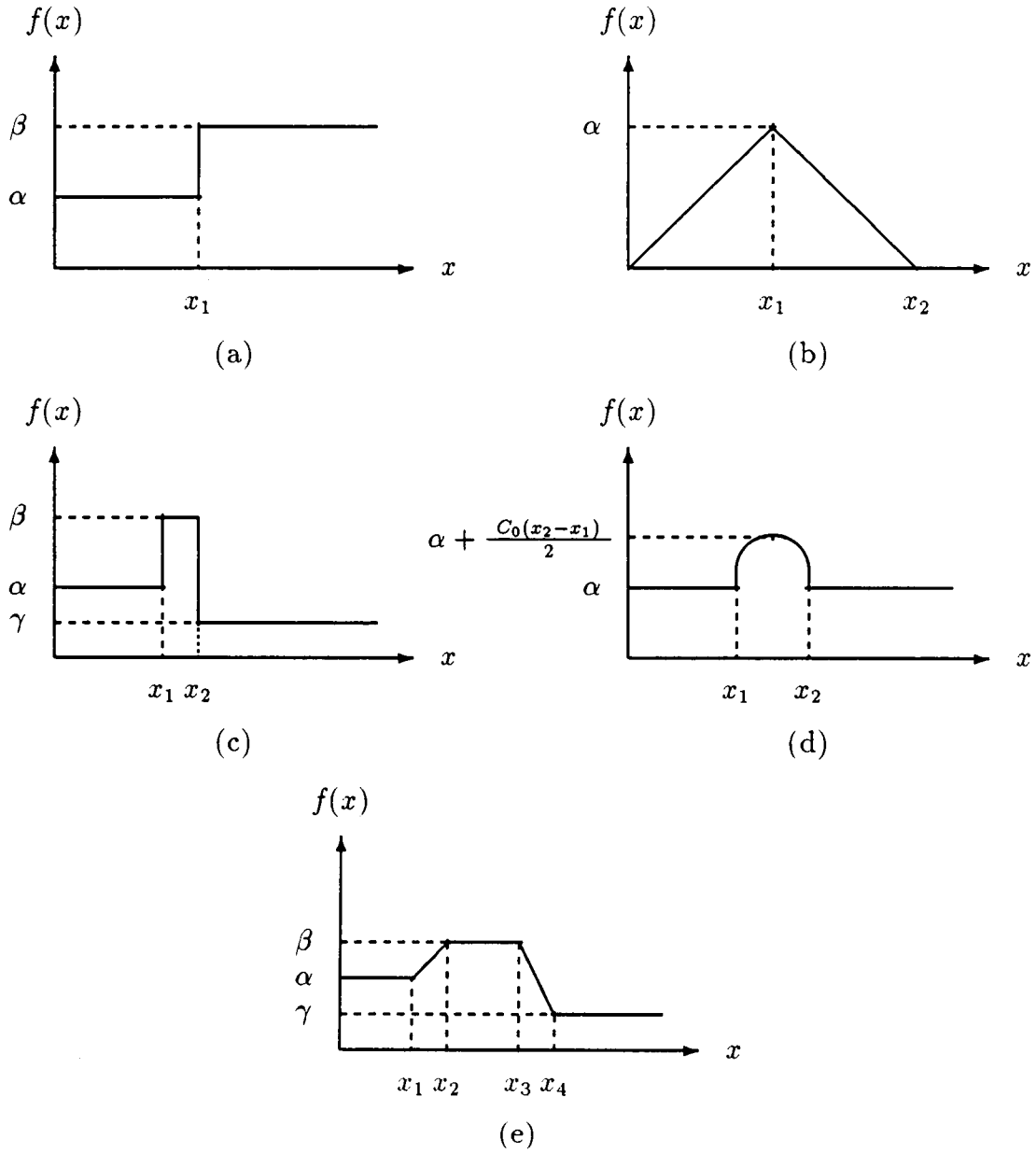


Figure 2.1: Ideal one-dimensional edge models. (a) ideal step edge. (b) ideal roof edge. (c) ideal line edge. (d) ideal spike edge. (e) ideal ridge edge (or ideal ramp edge).

2.2 Edge-oriented Segmentation

The earliest step-edge detectors were proposed by Roberts [90] and Prewitt [87]. In general, edge-oriented detection algorithms, depending on their implementation and edge orientation aspects, have several classifications. From the implementation point of view, edge-oriented detectors can be classified as either linear [71,100] or nonlinear [12,13,33,50,79,80,85,92,105] detectors and parallel [19] or sequential [9] detectors. From the edge orientation point of view, edge detectors are categorized as either directional [11,75,78] or isotropic [48,65,100].

Comprehensive surveys of the existing edge-oriented detection techniques were reported by Ballard and Brown [8], Davis [21], Kunt [56], Pavlidis [83], Rosenfeld [93], Rosenfeld and Kak [97], and Young and Fu [114]. A number of qualitative and/or quantitative measurements of some of these edge-oriented detectors were proposed by Abdou and Pratt [1], Bovik *et al.* [13], Deutsh and Fram [22], Fram and Deutsch [29], Lyvers and Mitchell [60], and Peli and Malah [84]. Comparisons of respective advantages and disadvantages of directional and isotropic detectors were presented by Brady and Horn [15] and Marr and Hildreth [65].

Edge detectors can perform over either small neighborhoods or large regions, and, hence, there are local and regional edge operators. Local edge operators detect edges over small subimages (2×2 or 3×3 pixels) by considering the

resulting magnitude and/or direction of each image pixel. Regional edge operators look for the best match between an idealized edge configuration and a large subimage (4×4 pixels or larger) [5,17,47,49,55,61,70,77,94,113]. The examination and comparison of local and regional edge operators was discussed by Shaw [101].

In general, edge-oriented image segmentation is a two-stage process. The first stage is edge extraction, and the second stage is edge thinning (or edge labeling). The purpose of the first stage is to extract the edges based on some *a priori* knowledge, *e.g.*, a magnitude threshold. The edge extraction process involves one of the following methods: gradient, surface fitting, template matching, line extraction, or artificial intelligence. The second stage, thinning, is a postprocessing stage used to locate the object contours in the resulting thick edges of the first stage. These two stages are further discussed in the following subsections.

2.2.1 Edge Extraction

1) Gradient Method

The gradient of an image $I(x, y)$ at an image pixel $I(i, j)$ is given by

$$\begin{aligned} \mathbf{G}[I(x,y)]_{(i,j)} &= \begin{bmatrix} \frac{\partial I(x,y)}{\partial x} \\ \frac{\partial I(x,y)}{\partial y} \end{bmatrix}_{(i,j)} \\ &= \begin{bmatrix} G_x(i,j) \\ G_y(i,j) \end{bmatrix} \end{aligned} \tag{2.6}$$

Various algorithms such as unweighted average methods [87,91], weighted average methods [16,30,40], and no-average methods [90] are used to approximate these two partial derivatives. One of the most well-known methods is the Sobel algorithm [36].

After evaluating gradients, each image pixel is assigned a gradient magnitude and/or a direction. According to an *a priori* threshold value, the qualified edgels, *i.e.*, short and piece-wise linear edge segments, are linked to form boundaries [16,31,48,51]. Usually the gradient method does not utilize a subimage larger than 3×3 pixels. Hence, the gradient method is an example of a local edge operator.

2) Surface Fitting Method

The surface fitting method is an example of a regional edge operator. In a

large window, this method approximates the object contours of the subimage which coincides with the window using Fourier series, least-squares planar-fit, least-squares cubic-fit, or interpolation according to the actual step edge positions and directions. Therefore, the edge of the subimage which coincides with the window can be represented by the best-fitting surface to minimize the effects of noise. Some well-known algorithms using this method were proposed by Haralick [39], Hueckel [46], Mero and Vassy [70], Nalwa and Binford [75], Nalwa [76], Prewitt [87], and Shaw [101].

3) Template Matching Method

This method uses pre-defined edge templates. For each image pixel, each edge template is convolved with the image with the center of the edge template at the image pixel. By choosing the maximum output, the edge template that is most similar to the image edge is found, and the magnitude and direction of the image pixel are determined. The number and size of the predefined edge templates decide the accuracy and resolution of this method. Usually, the template matching method uses large templates and is classified as a regional edge operator. The algorithm proposed by Nevatia and Babu [78] is a well-known algorithm that utilizes the template matching method.

4) Line Extraction Method

In a given image, this method extracts straight line edges rather than edge pixels. Depending on magnitude values and directions, edge pixels are grouped into line-support regions. By combining these line-like regions, boundaries are generated. The Hough transform [7,23,103] is a well-known algorithm using this method. Other algorithms were proposed by Burns *et al.* [16], Griffith [37], Horn [42], Mansouri *et al.* [62], and Shirai [102].

5) Artificial Intelligence Method

Artificial intelligence edge detection methods detect edges by comparing the properties of each image pixel in a given image against a rule-based decision algorithm. The multi-template algorithm is an example of this method. The multi-template algorithm can be utilized either in sequential [10] or in parallel [53,94,111] modes to provide edge detectors with fine resolution and minimal noise sensitivity.

2.2.2 Edge Thinning

Thinning is a procedure that extracts one-pixel-wide contours from thick edges by considering the magnitude and/or direction of each edge pixel. In digital image processing, thinning algorithms can be grouped into two major groups:

binary thinning algorithms [3,4,115] and gray-scale thinning algorithms [2,25,98]. In binary thinning algorithms, first, an intensity threshold is chosen to transfer an input edge image to a binary edge image. Then, edge thinning is accomplished by deleting edge pixels, which do not destroy the connectivity or junctions (T-type or Y-type) of edges, to obtain a one-pixel-wide edge image. In the gray-scale thinning algorithm, the thinning procedure is accomplished by considering the strength and/or the direction relationships of each edge pixel. Edge pixels which are not associated with the local maxima, and at the same time do not destroy the connectivity of the edges, are deleted.

In gray-scale thinning algorithms, the thinning procedure can be classified as either magnitude-based or magnitude-and-direction-based method. Magnitude-based methods are direction-blind, and typically they perform thresholding, skeletonization, and contour connection [24,88,89]. These methods have the benefits of easy implementation and connected contours; however, the positions of the edges may be shifted and the edge junctions, T-type or Y-type, may be destroyed. Magnitude-and-direction-based methods are pixel-by-pixel sequential processes which use both magnitude and direction information for each edge pixel. These methods choose a starting-point, set a threshold value and a variation angle between two adjacent edge pixels, and then perform the contour following [6,56,58,66,67]. The magnitude-and-direction-based methods result in one-pixel-wide edges which preserve junctions and exact edge positions. However, the resulting contours may be disconnected.

According to the structure of the resulting one-pixel-wide edges, there are two types of thinning algorithms: 4-connected and 8-connected thinning algorithms. The 4-connected thinning algorithms produce one-pixel-wide edges in which edge pixels are 4-neighbors of their adjacent edge pixels [46]. The resulting edge pixels of the 8-connected thinning algorithms are 8-neighbors of their adjacent edge pixels.

2.3 Result Improvement

In order to improve the result of an edge detection algorithm for a given image, preprocessing is preferred. The objective of preprocessing is to enhance the performance of an edge detection algorithm by eliminating noise, preserving edge information, and reducing degradation (*e.g.*, false contours, speckly effect, and blurring). Image enhancement [73,86,106,114,116] is used for this purpose (*e.g.*, median filter). When high resolution is desired in very noisy images (*e.g.*, infrared images), the preprocessing process is indispensable.

CHAPTER 3

PERFORMANCE OF DIRECTIONAL TEMPLATES IN EDGE DETECTION

The edge detection algorithm developed in this thesis cross-correlates several directional templates of different sizes with a neighborhood of each image pixel. Using a set of rules, the algorithm assigns edge confidence to each candidate edge pixel. By performing cross-correlation over neighborhoods of different sizes the algorithm has the unique ability to determine accurately the positions of edge pixels in noisy images. Accuracy results from processing neighborhoods of small sizes and ability to perform in the presence of noise arises from processing large neighborhoods.

For each template size, several directional templates are used to detect the edges with different orientations. A directional template is a square mask which models a straight edge in a particular direction. An example of positioning a directional template on the image pixel $I(i, j)$ is shown in Figure 3.1.

In this chapter, we define the directional templates. Then, we study the ability of these directional templates to find edge positions and directions for several kinds of edges (*i.e.*, single edge, double edges, corner edge, and noisy edge). Finally, we examine the effects of Gaussian noise on directional templates.

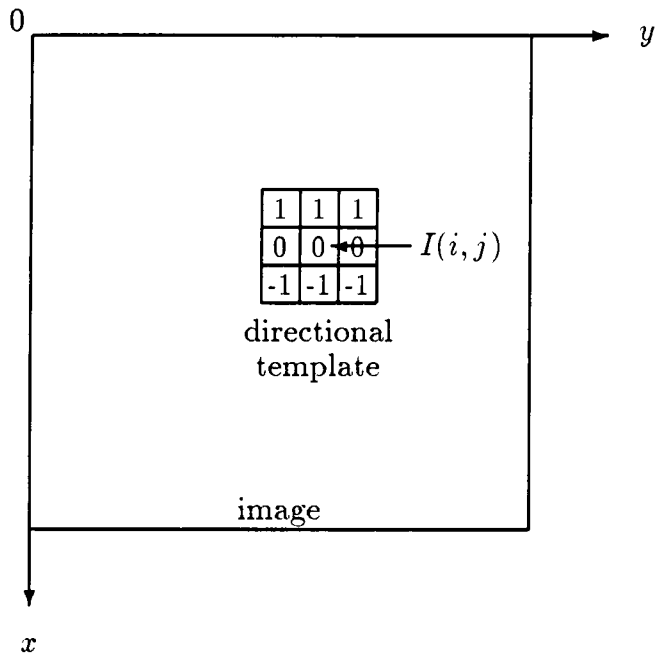


Figure 3.1: Positioning of a directional template on the image pixel $I(i,j)$ of an image.

3.1 Model of Ideal Binary Step Edge

Consider an $M \times N$ binary image $I(x,y)$, $x \in [0,\ M-1]$ and $y \in [0,\ N-1]$, with intensities a and b, $a \neq b$. An edge pixel $I(i,j)$ located on the edge and forming an angle θ_1, $-\pi/2 < \theta_1 < \pi/2$, with the x-axis (measured counterclockwise) displaced by $-h$ from the center of the image (measured along the y-axis) (shown in Figure 3.2), satisfies the equation

$$\begin{aligned} j &= \frac{(2i-M)\cdot\tan\theta_1 + N}{2} - h \\ &= \Gamma_1(i,\theta_1,h,M,N), \end{aligned} \tag{3.1}$$

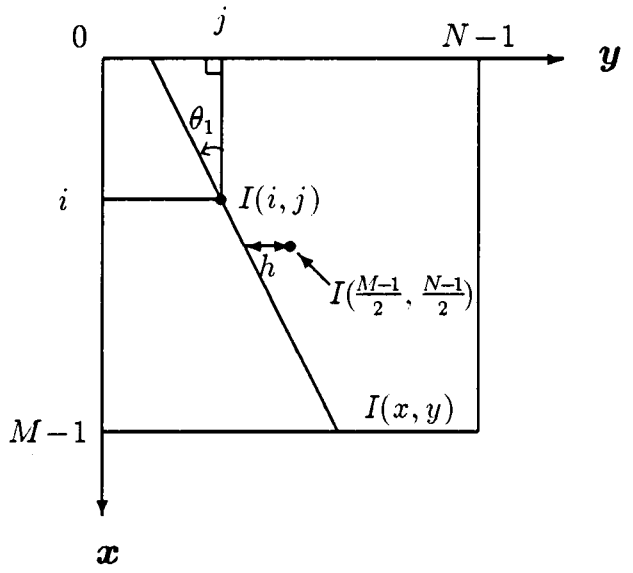


Figure 3.2: An edge pixel $I(i,j)$ located on the edge forming angle θ_1 with the x-axis and translated by $-h$ from the center of the image $I(x,y)$.

where $\Gamma_1(i, \theta_1, h, M, N)$ is a function of i, θ_1, h, M, and N.

Hence, a binary image $I(x,y)$ with size $M \times N$ and intensities a and b, containing an ideal step edge with angle θ_1 to the x-axis and translation $-h$ from the center (Figure 3.3), can be described by

$$I(x,y) = \begin{cases} a & \text{if } y \geq \Gamma_1(x, \theta_1, h, M, N) \\ b & \text{otherwise.} \end{cases} \tag{3.2}$$

The values of intensities a and b determine the strength of the edge.

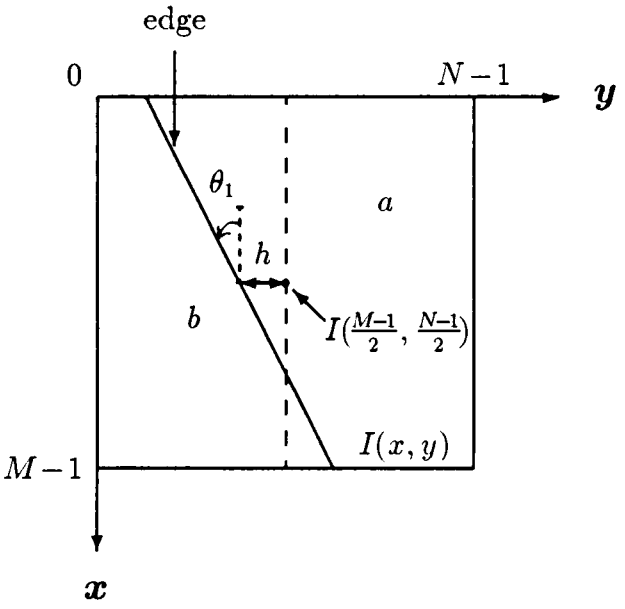


Figure 3.3: The model of an $M \times N$ binary image $I(x, y)$ with intensities a and b and a step edge at angle θ_1 relative to the x-axis and displaced by $-h$ from the center.

3.2 Model of n Directional Templates

In order to generate n directional templates, for each template size, we divide the 2π angle into n angles, each separated by an equal angle increment $\Delta\theta = 2\pi/n$. Hence, the edge angles θ_2, $-\pi \leq \theta_2 < \pi$, (measured counterclockwise relative to the x-axis) of the n directional templates, can be defined as

$$\theta_2 = l \cdot \Delta\theta, \tag{3.3}$$

where $l = n/2 - 1,\ n/2 - 2,\ ...,\ 1,\ 0,\ -1,\ ...,\ -n/2 + 1$, and $-n/2$. From Equation (3.1) it follows that for a $p \times q$ directional template, $x \in [0,\ p-1]$ and $y \in [0,\ q-1]$, a pixel $T(i,j)$ on the line which passes the center of the template and forms an angle θ_2 with the x-axis satisfies

$$\begin{aligned} j &= \frac{(2i-p)\cdot \tan\theta_2 + q}{2} \\ &= \Gamma_2(i, \theta_2, n, p, q). \end{aligned} \tag{3.4}$$

Hence, the $p \times q$ directional template with the edge forming the angle $\theta_2 = l \cdot \Delta\theta$ with the x-axis and template coefficients λ and $-\lambda$ is defined by

$$T(x,y) = \begin{cases} \lambda & \text{if } y > \Gamma_2(x, \theta_2, n, p, q) \\ 0 & \text{if } y = \Gamma_2(x, \theta_2, n, p, q) \\ -\lambda & \text{if } y < \Gamma_2(x, \theta_2, n, p, q), \end{cases} \tag{3.5}$$

where $\lambda \neq 0$. The construction of this directional template is shown in Figure 3.4.

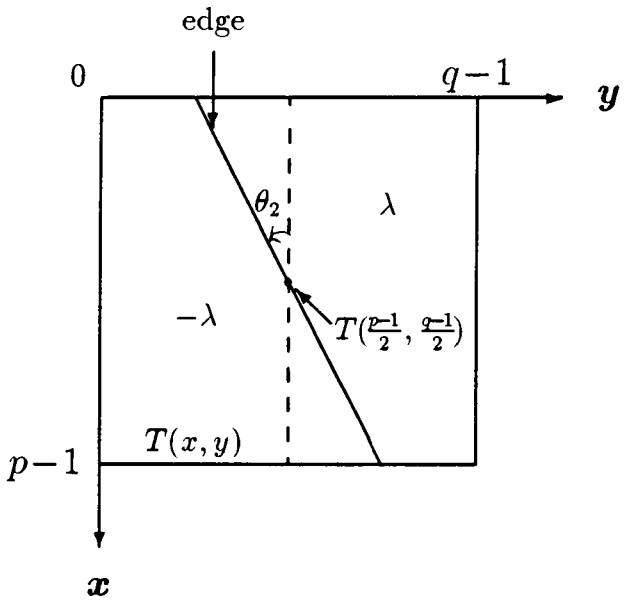


Figure 3.4: The model of a $p \times q$ directional template $T(x, y)$ with edge angle θ_2 relative to the x-axis and template coefficients λ and $-\lambda$.

3.3 Detection of Edge Position and Direction

In the previous sections, we defined the models for ideal step edge images and for directional templates. In this section, we examine the ability of the directional templates to find the correct positions and directions of edges in an image. For convenience, the continuous cases are used. The extension to the discrete cases is straightforward.

A commonly used method for edge detection using templates is *Cross-Correlation*. First, let $I(x, y)$ be the given image with size $M \times N$ and $T(x', y')$ be the directional template with size $p \times q$, $p \ll M$ and $q \ll N$. Since each $p \times q$ subimage of a given image $I(x, y)$ can be uniquely represented by its center pixel at each specific location $I(i, j)$ of the given image $I(x, y)$, the *Cross-Correlation Coefficient*, $\gamma(i, j)$, in the continuous case is

$$\gamma(i,j) = \frac{\int\int I(x,y) \cdot T(x-i, y-j)\, dx\, dy}{\{\int\int [I(x,y)]^2\, dx\, dy \cdot \int\int [T(x-i, y-j)]^2\, dx\, dy\}^{1/2}}, \qquad (3.6)$$

where $i \in [0,\ M-1]$ and $j \in [0,\ N-1]$, and the integrations are taken over the coordinates common to both $I(x, y)$ and $T(x, y)$ [36]. An example of positioning a directional template to calculate the $\gamma(i, j)$ is illustrated in Figure 3.5. In general, $\gamma(i, j)$ varies from one location to the other. The larger the value of $\gamma(i, j)$, the greater the similarity between $T(x, y)$ and the corresponding subimage.

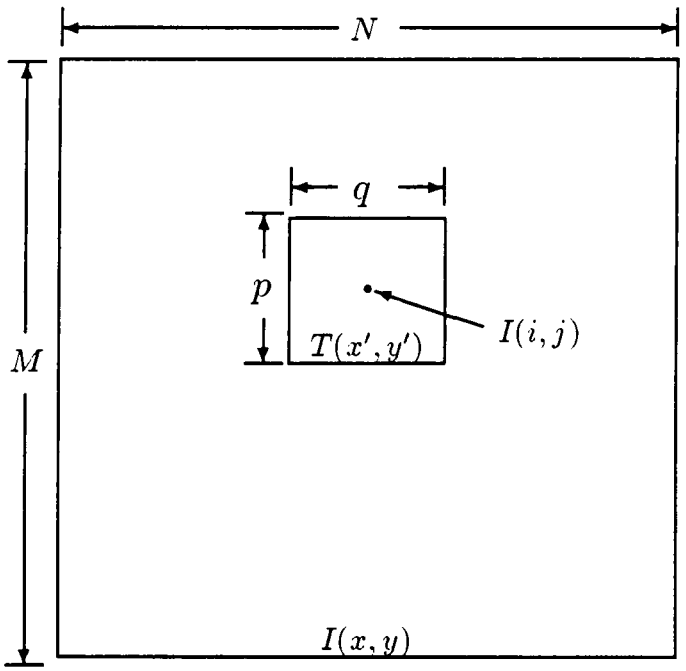


Figure 3.5: Computation of the cross-correlation coefficient of an $M \times N$ image $I(x, y)$ and a $p \times q$ directional template $T(x', y')$ at a specific image pixel $I(i, j)$.

In the following, the definitions and symbols developed in Sections 3.1 and 3.2 are used. Figure 3.6(a) contains a $p \times q$ directional template with orientation θ_2. Figure 3.6(b) contains a $p \times q$ subimage with an edge at angle θ_1 and translation $-h$ from image pixel $I(i, j)$. The overlap of the directional template and the subimage is shown in Figures 3.6(c) and 3.6(d) with different notation.

In order to simplify the notation, we assume Equation (3.6) to be of the form

$$\gamma(i, j) = \frac{U}{V}. \tag{3.7}$$

The relationships $p \leq q$, $a > b$, and $\theta_1 \leq \theta_2$ are assumed. According to Fig-

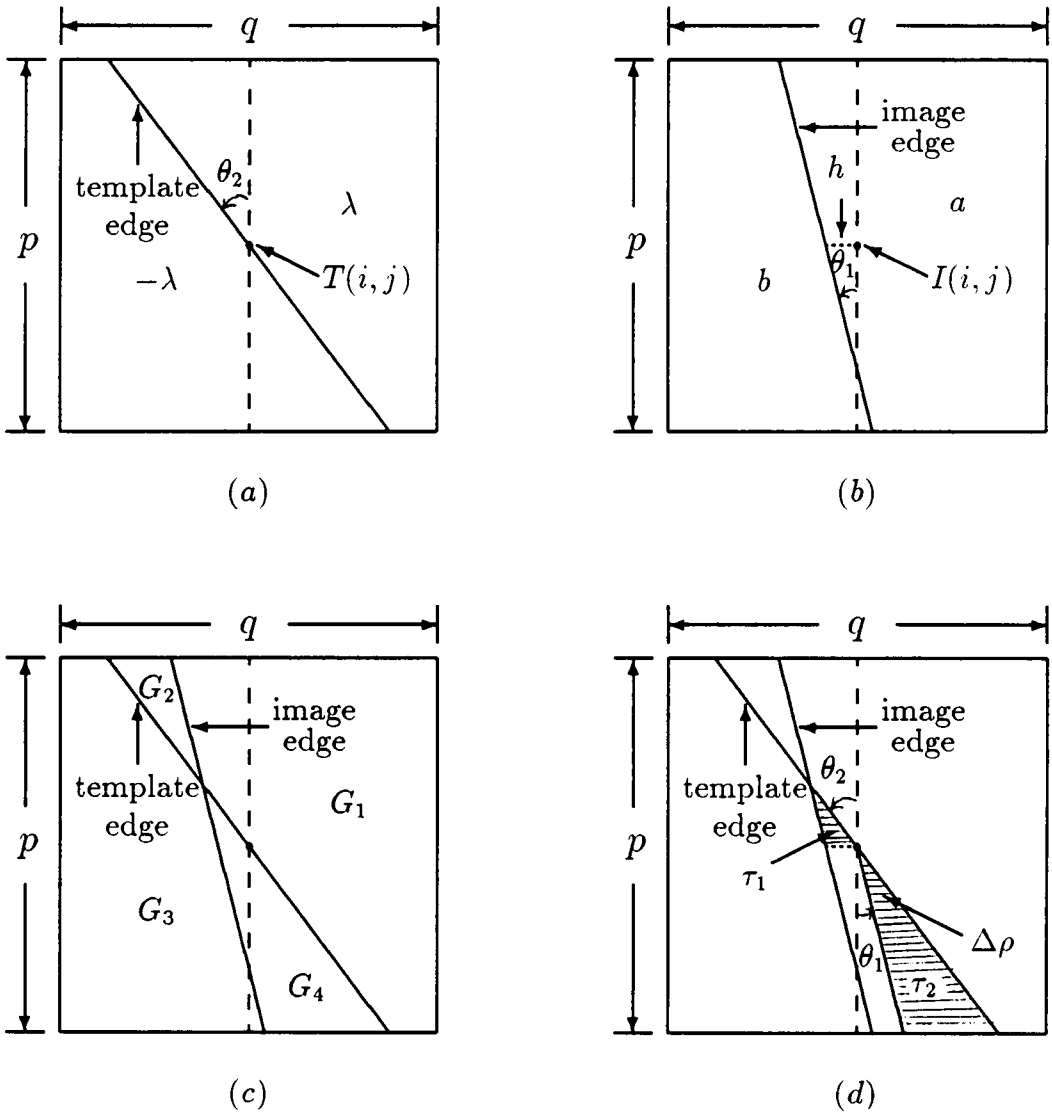


Figure 3.6: The directional template, edge subimage, and their overlaps with size $p \times q$ at the image pixel $I(i, j)$. (a) The directional template $T(x, y)$. (b) The edge subimage. (c)(d) The overlaps of (a) and (b) with different notation.

ures 3.6(a), 3.6(b), and Equation (3.6),

$$U = \lambda[a(G_1 - G_4) + b(G_2 - G_3)], \tag{3.8}$$

where G_1, G_2, G_3, and G_4 are subareas of $T(x, y)$ as shown in Figure 3.6(c). Also,

$$V = \frac{\lambda pq}{2}[2(a^2 + b^2) + \frac{4h}{q}(a^2 - b^2)]^{1/2}, \tag{3.9}$$

where h is the horizontal distance between the edge and the center pixel as shown in Figure 3.6(b).

From Figures 3.6(c) and 3.6(d), the areas of G_1, G_2, G_3, and G_4 are

$$G_1 = \frac{1}{2}(pq + hp - 2\tau_1 - 2\tau_2), \tag{3.10}$$

$$G_2 = \frac{1}{2}(2\tau_1 + 2\tau_2 - hp), \tag{3.11}$$

$$G_3 = \frac{1}{2}(pq - hp - 2\tau_1 - 2\tau_2), \tag{3.12}$$

and

$$G_4 = \frac{1}{2}(hp + 2\tau_1 + 2\tau_2), \tag{3.13}$$

where τ_1 and τ_2 are subareas of the directional template $T(x, y)$ as shown in Figure 3.6(d) with values

$$\tau_1 = \frac{h^2}{2(\tan\theta_2 - \tan\theta_1)} \tag{3.14}$$

and

$$\tau_2 = \frac{p^2}{8}(\tan\theta_2 - \tan\theta_1). \tag{3.15}$$

Substituting Equations (3.10) through (3.13) into Equation (3.8) gives

$$U = \frac{\lambda}{2}(a-b)(pq - 4\tau_1 - 4\tau_2). \tag{3.16}$$

Furthermore, substituting Equations (3.9) and (3.16) into Equation (3.7), the cross-correlation coefficient of the particular pixel $I(i,j)$ becomes

$$\gamma(i,j) = \frac{(a-b)(pq - 4\tau_1 - 4\tau_2)}{pq[2(a^2+b^2) + \frac{4h}{q}(a^2-b^2)]^{1/2}}. \tag{3.17}$$

Hence, in the case of n directional templates, the best matched directional template is the one that yields the maximum value in Equation (3.17). It appears that if the values of p, q, a, and b are known, the maximum value of Equation (3.17) occurs when the values of τ_1, τ_2, and h are minimized. Because the minimum value of h is 0, according to Equation (3.14), minimizing h to 0 yields both $h = 0$ and $\tau_1 = 0$. By minimizing the value of τ_2, referring to Equation (3.15),

$$\begin{aligned} min\{\ \tau_2\ \} &\Rightarrow min\{\ \tan\theta_2 - \tan\theta_1\ \} \\ &\Rightarrow min\{\ \Delta\rho\ \}, \end{aligned} \tag{3.18}$$

where min denotes the minimization and $\Delta\rho = \theta_2 - \theta_1$ is the angle between the image edge and the template edge as shown in Figure 3.6(d). Using Equation (3.3) on page 19, the orientation θ_2 of the best matched directional template satisfies

$$|\theta_2 - \theta_1| = min_l\{\ |l\Delta\theta - \theta_1|\ \}, \tag{3.19}$$

where $l = n/2-1, n/2-2, ..., 1, 0, -1, ..., -n/2+1$, and $-n/2$. The ideal case is $\Delta\rho = 0$.

When the ideal match is reached, *i.e.*, $h = \tau_1 = \tau_2 = 0$, according to Equation (3.17), the maximum cross-correlation coefficient at the particular image pixel $I(i,j)$ is

$$\gamma(i,j) = \frac{a-b}{[2(a^2+b^2)]^{1/2}}. \tag{3.20}$$

Based on the discussion above, the edge of the directional template, which yields the maximum cross-correlation coefficient in the n directional templates, has the closest orientation to the edge of the corresponding subimage. The center of the directional template is located on the edge.

3.4 Relationship Between Correlation Coefficient and Template Size

Assume the ideal match in Section 3.3 is reached, *i.e.*, $\theta_1 = \theta_2$ and $h = 0$, and the edge of the directional template is vertical with angle $\theta_1 = \theta_2 = 0$. The directional template and the vertical edge subimage with size $p \times q$, $p = 2k\Delta p+1$, $q = 2k\Delta q+1$, and $k = 1, 2, 3, ...,$ are shown in Figures 3.7(a) and 3.7(b), where Δp and Δq are the unit increments of the template size.

According to Figures 3.7(a), 3.7(b), and the definition of cross-correlation in

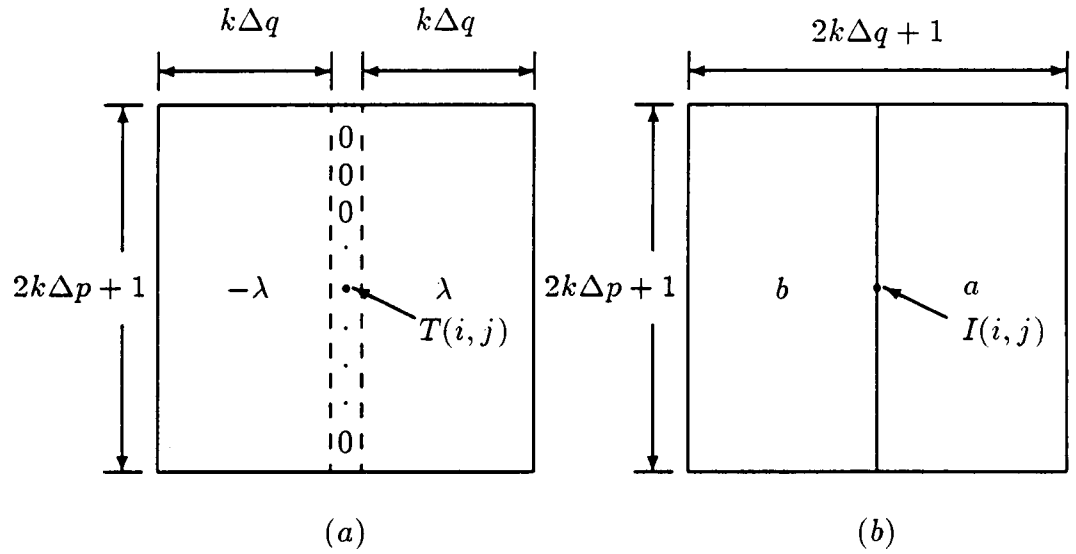


Figure 3.7: Directional template and edge subimage with vertical edge and common size $(2k\Delta p + 1) \times (2k\Delta q + 1)$. (a) Directional template with template coefficients λ and $-\lambda$. (b) Edge subimage with intensities a and b and the center pixel at $I(i, j)$.

Equation (3.6) on page 21, the numerator of Equation (3.7) becomes

$$U = \lambda(a-b)(2k\Delta p+1)(k\Delta q) \tag{3.21}$$

and the denominator of Equation (3.7) becomes

$$V = \lambda(2k\Delta p+1)(k\Delta q)\cdot[2(a^2+b^2)+\frac{2b^2}{k\Delta q}]^{1/2} \tag{3.22}$$

Substituting Equations (3.21) and (3.22) into Equation (3.7) produces

$$\gamma(i,j) = \frac{a-b}{[\,2(a^2+b^2)+\frac{2b^2}{k\Delta q}\,]^{1/2}}. \tag{3.23}$$

From Equation (3.23) it follows that when $b \neq 0$ the cross-correlation coefficient, $\gamma(i,j)$, increases when the template size increases, *i.e.*, k increases.

3.5 Corner Edges

Section 3.3 shows that the directional templates can find the positions and directions of straight edges. In this section we consider a complex situation that arises when encountering a corner edge.

For simplicity, we consider a right-angled corner edge. Using this simplification and the definitions in Sections 3.1 and 3.2, the problem can be classified into three cases : (a) $0 \leq \theta_2 \leq \tan^{-1}(p-2d)/p$, (b) $\tan^{-1}(p-2d)/p < \theta_2 < \tan^{-1} q/p$, and (c) $\tan^{-1} q/p \leq \theta_2 \leq \pi/2$, as shown in Figures 3.8(a), 3.8(b), and 3.8(c), respectively, where d is the distance between the corner and the center of the

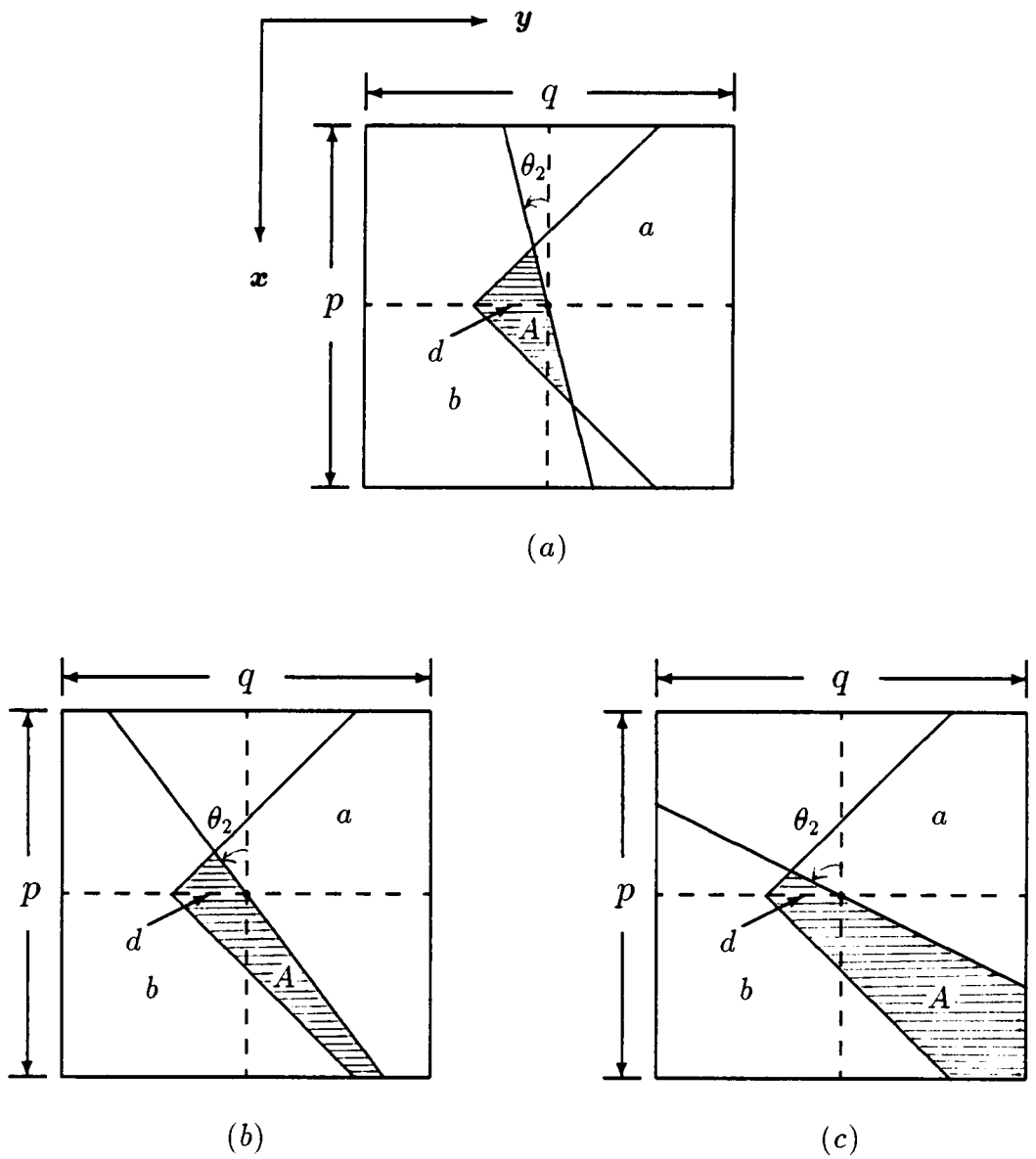


Figure 3.8: The overlaps of directional template and right-angled corner edge with size $p \times q$. The angle θ_2 is the angle between the template edge and the x-axis. (a) $0 \leq \theta_2 \leq \tan^{-1}(p-2d)/p$. (b) $\tan^{-1}(p-2d)/p < \theta_2 < \tan^{-1} q/p$. (c) $\tan^{-1} q/p \leq \theta_2 \leq \pi/2$.

directional template. The range of θ_2 is $0 \leq \theta_2 \leq \frac{\pi}{2}$. Therefore, the numerator of the cross-correlation coefficient, referring to Equation (3.6) on page 21, is

$$\begin{aligned} U &= \int_0^q \int_0^p I(x,y) \cdot T(x-i, y-j)\, dx\, dy \\ &= \lambda[(a-b)(\frac{pq}{2} - \frac{p^2}{4} + pd - 2A)], \end{aligned} \tag{3.24}$$

where A represents the dashed areas as shown in Figures 3.8(a), 3.8(b), and 3.8(c), and

$$\begin{aligned} V &= \{\int_0^q \int_0^p [I(x,y)]^2\, dx\, dy \cdot \int_0^q \int_0^p [T(x-i, y-j)]^2\, dx\, dy\}^{1/2} \\ &= \{[(a^2-b^2)(pd - \frac{p^2}{4} + \frac{pq}{2}) + b^2 pq] \cdot \lambda^2 pq\}^{1/2}. \end{aligned} \tag{3.25}$$

The value of A in Equation (3.24) is calculated in the following three cases (also refer to Figure 3.8).

Case 1 : If $0 \leq \theta_2 \leq \tan^{-1}(p-2d)/p$ as shown in Figure 3.8(a),

$$A = \frac{d^2}{1 - \tan^2 \theta_2}. \tag{3.26}$$

Case 2 : If $\tan^{-1}(p-2d)/p < \theta_2 < \tan^{-1} q/p$ as shown in Figure 3.8(b),

$$A = \frac{d^2}{2(1+\tan\theta_2)} + \frac{dp}{2} - \frac{p^2}{8}(1 - \tan\theta_2). \tag{3.27}$$

Case 3 : If $\tan^{-1} q/p \leq \theta_2 \leq \pi/2$ as shown in Figure 3.8(c),

$$A = \frac{d^2}{2(1+\tan\theta_2)} - \frac{q^2}{8\tan\theta_2} + \frac{pq}{4} + \frac{pd}{2} - \frac{p^2}{8}. \tag{3.28}$$

From Equations (3.26), (3.27), and (3.28), it follows that when the value of angle θ_2 decreases, the value of A decreases. From Equation (3.24) it follows

that when the value of A decreases, the value of U increases. Because the value of V is independent of the value of θ_2, it can be concluded that when the angle θ_2 decreases, the cross-correlation coefficient at a particular image pixel $I(i,j)$ increases. Hence, the maximum value of $\gamma(i,j)$ occurs when $\theta_2 = 0$. Then, from Equation (3.26), when $\theta_2 = 0$, the dashed area A becomes

$$A = d^2. \tag{3.29}$$

Substituting Equations (3.24), (3.25), and (3.29) into Equation (3.6) on page 21, the cross-correlation coefficient becomes

$$\gamma(i,j) = \frac{(a-b)(\frac{pq}{2} - \frac{p^2}{4} + pd - 2d^2)}{\{[(a^2-b^2)(pd - \frac{p^2}{4} + \frac{pq}{2}) + b^2pq] \cdot pq\}^{1/2}}. \tag{3.30}$$

Consider a special case of Equation (3.30) when the right-angled edge is very strong, *i.e.*, $a \gg b$. In order to further simplify the problem, we assume the templates are square, *i.e.*, $p = q$. Then, Equation (3.30) reduces to

$$\gamma(i,j) = \frac{1 + \frac{4d}{p} - \frac{8d^2}{p^2}}{2(1 + \frac{4d}{p})^{1/2}}. \tag{3.31}$$

Setting $\omega = d/p$, we obtain the form

$$\gamma(i,j) = \frac{1 + 4\omega - 8\omega^2}{2(1 + 4\omega)^{1/2}}. \tag{3.32}$$

Since $0 \leq d \leq p/2$, the range of ω is

$$0 \leq \omega \leq \frac{1}{2}. \tag{3.33}$$

By substituting the values in the interval of Equation (3.33) into Equation (3.32), the maximum $\gamma(i,j)$ occurs when $\omega = 0.14$, and the maximum value is

$\gamma(i,j) = 0.5617$ (refer to Appendix A). The construction of the best match is illustrated in Figure 3.9.

Hence, by choosing the directional template that is most similar to the edge, the direction of a strong right-angled edge is perpendicular to the bisector of the corner angle, and the distance between the real corner edge and the resulting corner edge on the bisector equals the size of the template multiplied by 0.14. In other words, the edge detection algorithm, using directional templates, performs a rounding-off effect on the corner. The strong right-angled corner and the resulting rounded edge are shown in Figure 3.10.

3.6 Two-Edge Regions

In Section 3.3, there is only one edge in the subimage which coincides with the directional templates. In this section, we consider the case of two edges in the subimage which coincides with the directional templates. The configuration of the edge subimage and the directional templates are similar to Figure 3.6 on page 23 as shown in Figure 3.11. The intensities of these three image regions, resulting from the two edges, are a, b, and c where $a > b$ and $a > c$. The edge which is closer to the center of the directional template is called the main edge, and the other is called the secondary edge.

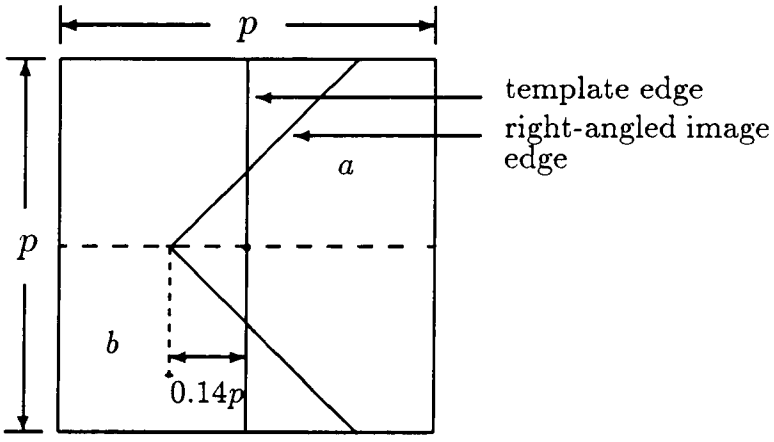


Figure 3.9: The best match of a strong right-angled edge and a directional template with vertical edge.

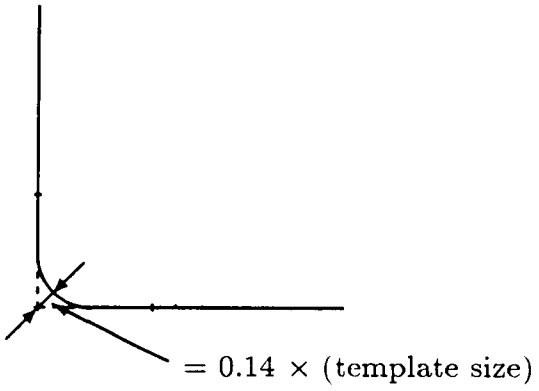


Figure 3.10: The strong right-angled corner (dashed line) and the resulting rounded edge (solid line).

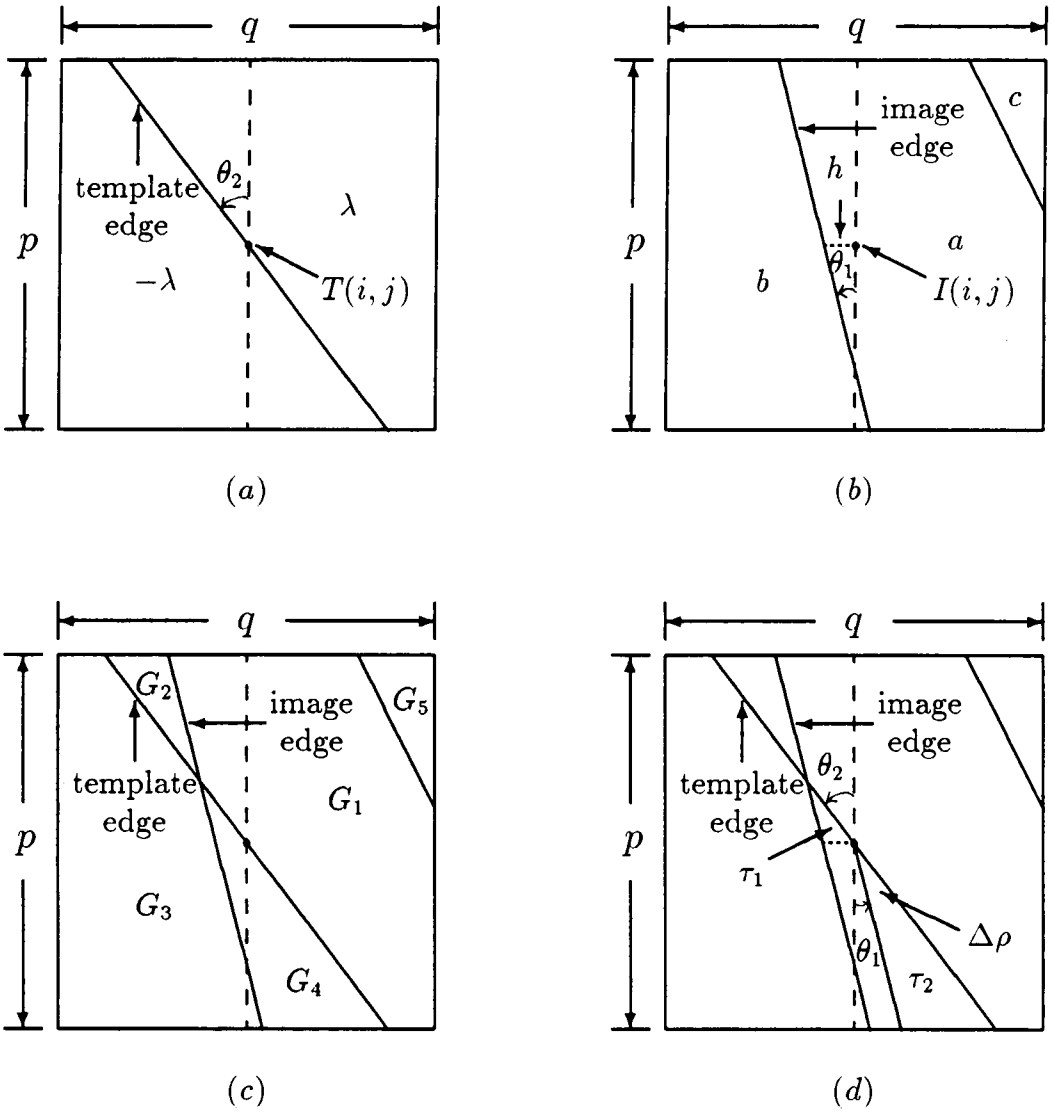


Figure 3.11: The directional template, edge subimage, and their overlaps with size $p \times q$ at the image pixel $I(i,j)$. (a) The directional template $T(x,y)$. (b) The edge subimage contains two edges with intensities a, b, and c. (c)(d) The overlaps of (a) and (b).

Referring to Figure 3.11(c), the numerator of the cross-correlation coefficient $\gamma(i,j)$ can be found by modifying the Equation (3.8) on page 24

$$U = \frac{\lambda}{2}[(a-b)(pq - 4\tau_1 - 4\tau_2) - 2G_5(a-c)], \tag{3.34}$$

where G_5 is a subarea of the directional template as shown in Figure 3.11(c) and τ_1 and τ_2 are defined in Equations (3.14) and (3.15) on page 24, respectively. By modifying the Equation (3.9) on page 24,

$$V = \frac{\lambda pq}{2}[2(a^2+b^2) + \frac{4h}{q}(a^2-b^2) - \frac{4G_5}{pq}(a^2-c^2)]^{1/2}. \tag{3.35}$$

Therefore, referring to Equations (3.34) and (3.35), Equation (3.17) on page 25 becomes

$$\gamma(i,j) = \frac{(a-b)(pq - 4\tau_1 - 4\tau_2) - 2G_5(a-c)}{pq[2(a^2+b^2) + \frac{4h}{q}(a^2-b^2) - \frac{4G_5}{pq}(a^2-c^2)]^{1/2}}. \tag{3.36}$$

If the intensity c and area G_5 are known, it follows that when τ_1, τ_2, and h are zero, the maximum value of $\gamma(i,j)$ is reached. The situation is the same as that in Section 3.3.

Hence, it appears that if there are two edges in the subimage coinciding with the directional templates, the position and the direction of the main edge which is closer to the center of the directional templates, is found by the directional templates.

Next, the relationship between the cross-correlation coefficient and the template size shown in Figure 3.11 is observed. The constructions of the two directional templates with sizes $(2k\Delta p + 1) \times (2k\Delta q + 1)$ and $(2(k+1)\Delta p + 1) \times$

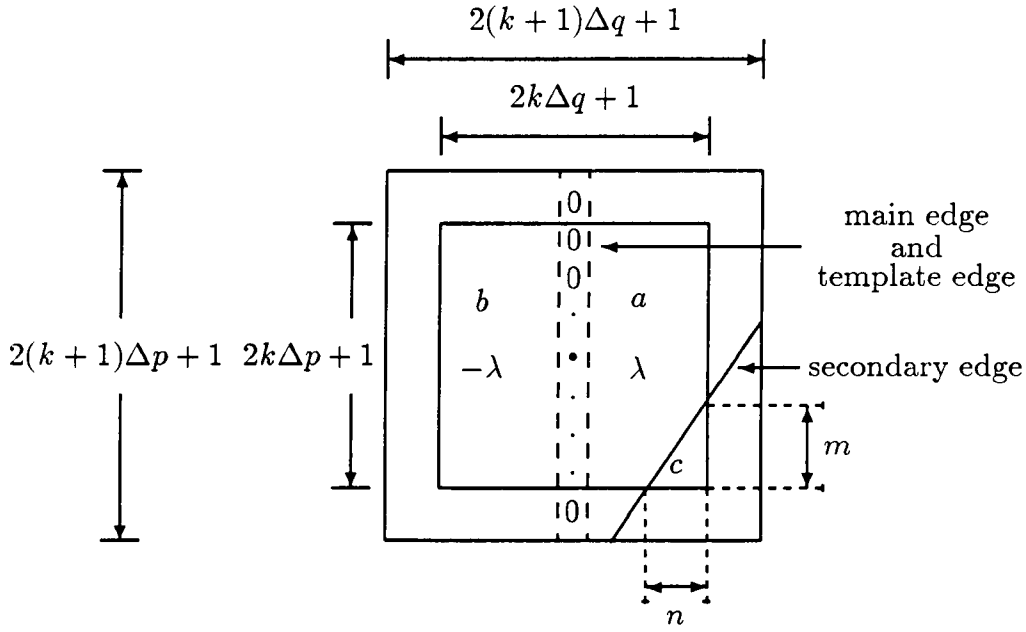


Figure 3.12: The arrangement of a subimage containing two edges and two directional templates with sizes $(2k\Delta p + 1) \times (2k\Delta q + 1)$ and $(2(k+1)\Delta p + 1) \times (2(k+1)\Delta q + 1)$.

$(2(k+1)\Delta q + 1)$ and the subimage with two edges in which an edge coincides with the edge of the directional templates are shown in Figure 3.12. As shown in Figure 3.12, the vertical and horizontal dimensions of the image region with intensity c on the inner template are m and n, respectively.

In reference to Section 3.4, the numerator of Equation (3.7) on page 22 becomes

$$U = \lambda(2k\Delta p + 1)(k\Delta q)[a - b - \frac{A_k(a-c)}{(2k\Delta p+1)(k\Delta q)}] \qquad (3.37)$$

and the denominator of Equation (3.7) becomes

$$V = \lambda(2k\Delta p+1)(k\Delta q)\cdot[2(a^2+b^2)+\frac{2b^2}{k\Delta q}-\frac{2A_k(a^2-c^2)}{(2k\Delta p+1)(k\Delta q)}]^{1/2}, \quad (3.38)$$

where A_k is the area with intensity c in the $(2k\Delta p+1)\times(2k\Delta q+1)$ directional template. Therefore, the cross-correlation coefficient equals

$$\gamma(i,j) = \frac{a-b-\frac{A_k(a-c)}{(2k\Delta p+1)(k\Delta q)}}{[2(a^2+b^2)+\frac{2b^2}{k\Delta q}-\frac{2A_k(a^2-c^2)}{(2k\Delta p+1)(k\Delta q)}]^{1/2}}. \quad (3.39)$$

The area A_k with $k=1$ equals

$$A_1 = \frac{mn}{2} \quad (3.40)$$

and when $k=2$, the area increases to

$$A_2 = \frac{1}{2}[mn+2(m+n)+2+\frac{n}{m}+\frac{m}{n}]. \quad (3.41)$$

By denoting the cross-correlation coefficients γ_1 and γ_2 for $k=1$ and $k=2$, respectively, from Equation (3.39),

$$\gamma_1 = \frac{a-b-\frac{(a-c)mn}{2(2\Delta p+1)(\Delta q)}}{[2(a^2+b^2)+\frac{2b^2}{\Delta q}-\frac{(a^2-c^2)mn}{(2\Delta p+1)(\Delta q)}]^{1/2}} \quad (3.42)$$

and

$$\gamma_2 = \frac{a-b-\frac{(a-c)[mn+2(m+n)+2+\frac{m}{n}+\frac{n}{m}]}{4(4\Delta p+1)(\Delta q)}}{[2(a^2+b^2)+\frac{b^2}{\Delta q}-\frac{(a^2-c^2)[mn+2(m+n)+2+\frac{m}{n}+\frac{n}{m}]}{2(4\Delta p+1)(\Delta q)}]^{1/2}}. \quad (3.43)$$

We are interested in the critical value of the area A_1 for which the cross-correlation coefficient does not increase when the template size increases, *i.e.*,

$$\gamma_1 \geq \gamma_2. \quad (3.44)$$

To simplify Equations (3.42) and (3.43), we make some assumptions: (i) square templates, *i.e.*, $p = q$, (ii) unit increment, *i.e.*, $\Delta p = \Delta q = 1$, and (iii) the region A_k is an equilateral right-angled triangle, *i.e.*, $m = n$. With these assumptions, we obtain

$$\gamma_1 = \frac{a - b - \frac{(a-c)m^2}{6}}{[2(a^2 + 2b^2) - \frac{(a^2-c^2)m^2}{3}]^{1/2}} \tag{3.45}$$

and

$$\gamma_2 = \frac{a - b - \frac{(a-c)(m+2)^2}{20}}{[2a^2 + 3b^2 - \frac{(a^2-c^2)(m+2)^2}{10}]^{1/2}}. \tag{3.46}$$

In order to obtain the intrinsic information, Equation (3.44) is further classified into three cases: (i) the secondary edge is much weaker than the main edge, (ii) the secondary edge is as strong as the main edge, and (iii) the secondary edge is much stronger than the main edge. In the second case, there are three conditions: both of the edges are either weak, medium, or strong. Examples shown in Tables 3.1 to 3.5 illustrate these specific cases. In these tables, symbol "+" means that when the template size increases, the cross-correlation coefficient increases, and symbol "−" means that when the template size increases, the cross-correlation coefficient decreases. Table 3.1 shows results of the case when the secondary edge is much weaker than the main edge. Tables 3.2 to 3.4 show the results of the case when the secondary edge is as strong as the main edge. The three conditions in the case when both edges are either weak, medium, or strong are shown in Tables 3.2, 3.3, and 3.4, respectively. Table 3.5 shows the result of the case when the secondary edge is much stronger than the main edge.

Table 3.1: Cross-correlation coefficients where the secondary edge is much weaker than the main edge, *i.e.*, $a - b \gg a - c$. The intensities are a=200, b=100, and c=195. The symbol m is the dimension of the subarea with intensity c as shown in Figure 3.12.

m	Template Size		
	$3 \rightarrow 5$	$5 \rightarrow 7$	$7 \rightarrow 9$
0	+0.0109	+0.0038	+0.0019
1	+0.0101	+0.0031	+0.0014
2		+0.0029	+0.0012
3			+0.0011

Table 3.1 shows, for all possible values of m, the cross-correlation coefficient always increases when the template size increases. In other words, in two-edge regions, the cross-correlation acts as if there were only one single straight edge in the area of the subimage. From Table 3.2 to Table 3.5, when the template size increases, for all possible values of m, the cross-correlation coefficient always decreases. Moreover, the stronger the secondary edge is, the more the cross-correlation coefficient decreases. Hence, when the subimage contains two edges, the cross-correlation coefficient does not increase as the template size increases, except when the secondary edge is very weak.

Table 3.2: Cross-correlation coefficients where the secondary edge is as strong as the main edge, i.e., $a - b \approx a - c$. The intensities are a=200, b=190, and c=190. (Both edges are weak.)

m	Template Size		
	$3 \rightarrow 5$	$5 \rightarrow 7$	$7 \rightarrow 9$
0	−0.0026	−0.0014	−0.0009
1	−0.0048	−0.0031	−0.0020
2		−0.0036	−0.0026
3			−0.0029

Table 3.3: Cross-correlation coefficients where the secondary edge is as strong as the main edge, i.e., $a - b \approx a - c$. The intensities are a=200, b=100, and c=100. (Both edges are of medium strength.)

m	Template Size		
	$3 \rightarrow 5$	$5 \rightarrow 7$	$7 \rightarrow 9$
0	−0.0331	−0.0168	−0.0099
1	−0.0604	−0.0339	−0.0207
2		−0.0417	−0.0279
3			−0.0315

Table 3.4: Cross-correlation coefficients where the secondary edge is as strong as the main edge, i.e., $a-b \approx a-c$. The intensities are a=200, b=10, and c=10. (Both edges are strong.)

m	Template Size		
	$3 \rightarrow 5$	$5 \rightarrow 7$	$7 \rightarrow 9$
0	−0.0708	−0.0328	−0.0189
1	−0.1152	−0.0594	−0.0354
2		−0.0725	−0.0467
3			−0.0529

Table 3.5: Cross-correlation coefficients where the secondary edge is much stronger than the main edge, i.e., $a-b \ll a-c$. The intensities are a=200, b=190, and c=100.

m	Template Size		
	$3 \rightarrow 5$	$5 \rightarrow 7$	$7 \rightarrow 9$
0	−0.0449	−0.0219	−0.0129
1	−0.0727	−0.0399	−0.0244
2		−0.0479	−0.0320
3			−0.0357

3.7 Effects of Noise

In this section, we assume that a given image contains additive Gaussian noise with zero mean and variance σ^2, *i.e.*, $N(0;\sigma^2)$. In the presence of additive Gaussian noise, the intensities of the edge subimage in Figure 3.6(b) become $a + N(0;\sigma^2)$ and $b + N(0;\sigma^2)$.

Therefore, in the noisy case, Equation (3.16) on page 25 becomes

$$\begin{aligned} U &= \frac{\lambda}{2}(a-b)(pq - 4\tau_1 - 4\tau_2) + \lambda \cdot \iint_{G_1 G_2} N(0;\sigma^2)\, dx\, dy \\ &\quad -\lambda \cdot \iint_{G_3 G_4} N(0;\sigma^2)\, dx\, dy \\ &= \frac{\lambda}{2}[(a-b)(pq - 4\tau_1 - 4\tau_2) + N(0;C_1\sigma^2)], \end{aligned} \tag{3.47}$$

where $C_1 = 4p^2q^2$. Equation (3.9) on page 24 becomes

$$\begin{aligned} V &= \frac{\lambda pq}{2}\{2[(a + N(0;\sigma^2))^2 + (b + N(0;\sigma^2))^2] \\ &\quad + \frac{4h}{q}[(a + N(0;\sigma^2))^2 - (b + N(0;\sigma^2))^2]\}^{1/2} \\ &= \frac{\lambda pq}{2}[2(a^2 + b^2) + \frac{4h}{q}(a^2 - b^2) + N(0;C_2\sigma^2) \\ &\quad + 2N_1]^{1/2}, \end{aligned} \tag{3.48}$$

where $C_2 = [4(a+b) + 8h(a-b)/q]^2$, and $N_1 = 2[N(0;\sigma^2)]^2$. Therefore, the cross-correlation coefficient becomes

$$\gamma(i,j) = \frac{(a-b)(pq - 4\tau_1 - 4\tau_2) + N(0;C_1\sigma^2)}{pq[2(a^2+b^2) + \frac{4h}{q}(a^2-b^2) + N(0;C_2\sigma^2) + 2N_1]^{1/2}}. \tag{3.49}$$

Because the expected value of a Gaussian function is its mean, and the expected value of the square of a Gaussian function is its variance [81], it follows

that $E[N(0; C_1\sigma^2)] = 0$, $E[N(0; C_2\sigma^2)] = 0$, and $E[N_1] = 2\sigma^2$, where E denotes the expected value. Hence, the expected value of Equation (3.49) is

$$E[\ \gamma(i,j)\ ] = \frac{(a-b)(pq - 4\tau_1 - 4\tau_2)}{pq[2(a^2+b^2) + \frac{4h}{q}(a^2-b^2) + 4\sigma^2]^{1/2}}. \tag{3.50}$$

A comparison of Equation (3.50) and Equation (3.17) on page 25, confirms that, in the presence of additive Gaussian noise, we can still detect the correct edge locations and directions by maximizing the expected value of the cross-correlation coefficient. Hence, the directional templates we defined earlier can find both the positions and directions of edges whether or not noise is present.

Considering the relationship between the template size and the cross-correlation coefficient in the noisy case, the Equation (3.21) on page 28 becomes

$$U = \lambda[(a-b)(2k\Delta p + 1)(k\Delta q) + N(0; C_3\sigma^2)], \tag{3.51}$$

where $C_3 = 16(2k\Delta p + 1)^2(k\Delta q)^2$, and the Equation (3.22) on page 28 becomes

$$\begin{aligned} V &= \lambda(2k\Delta p + 1)(k\Delta q) \cdot \{2(a^2+b^2) + \frac{2b^2}{k\Delta q} \\ &\quad + N(0; C_4\sigma^2) + C_5 \cdot N_1\}^{1/2}, \end{aligned} \tag{3.52}$$

where $C_4 = 2[(a+b) + b/(k\Delta q)]^2$ and $C_5 = [2 + 1/(k\Delta q)]$.

With the expected values $E[N(0; C_3\sigma^2)] = 0$, $E[N(0; C_4\sigma^2)] = 0$, and $E[N_1] = 2\sigma^2$, the expected value of the cross-correlation coefficient becomes

$$E[\ \gamma(i,j)\ ] = \frac{a-b}{[2(a^2+b^2) + \frac{2b^2}{k\Delta q} + 2C_5\sigma^2]^{1/2}}. \tag{3.53}$$

A comparison of Equation (3.53) and Equation (3.23) on page 28 shows that with the presence of additive Gaussian noise, the expected value of the cross-correlation coefficient increases as the template size is increased. Hence, with the presence of noise, the similarity between the appropriate directional template and the coinciding subimage increases when the template size increases; however, the coefficient is smaller than the coefficient in the noise-free case.

CHAPTER 4

RULE-BASED MULTI-TEMPLATE EDGE DETECTION AND THINNING

In this chapter, a rule-based multi-template edge detection algorithm is developed. In order to achieve computational efficiency, this algorithm uses the combination of either three or four template sizes (3×3, 5×5, 7×7, and 9×9 pixels) to detect the positions and directions of edge pixels. The algorithm is based on two sets of decision rules: the first governs the relationship of magnitude values obtained from different template sizes; the second handles the relationship of directions provided by different template sizes. Each set of rules employs an adjustable threshold. Based on these rules, each qualified edge pixel is assigned a final edge strength and direction.

This chapter presents the details of the two sets of decision rules for both the three-template and the four-template combinations, which are based upon the algorithm proposed by Patton and Brzakovic [82]. In addition, this chapter describes a thinning algorithm based upon Lacroix's [57] and Kunt's [56] methods to produce one-pixel-wide continuous contours. This algorithm locates the strength ridge of the edge image which resulted from the rule-based multi-template edge detection algorithm.

In order to describe the edge directions appropriately, we quantize the 2π angle into eight directions in $\pi/4$ increments. Consequently, the variable n of the orientation system described in Section 3.2 is 8. Symbol $\mathcal{D}$, numbered 1 to 8, denotes the direction that corresponds to the angles 0, $\pi/4$, $\pi/2$, $3\pi/4$, ... , and $7\pi/4$, respectively. The angles are measured counterclockwise relative to the x-axis. The edge orientation system, corresponding to the 8 direction numbers, is shown in Figure 4.1. Each edge pixel is assigned a direction $\mathcal{D}$, which is the closest direction to the direction of the edge pixel in the orientation system. For an edge pixel $I(i,j)$, the direction of the edge pixel is defined as the direction of the edge on which the edge pixel $I(i,j)$ is located. The angle from a vector pointing in the direction of decreasing intensity to the direction assigned to the edge is $\pi/2$ measured counterclockwise. The scheme of the direction assignment at the edge pixel $I(i,j)$ is shown in Figure 4.2.

We assume the directional templates defined in Section 3.2 are square and have four template sizes with the dimensions of 3 $\times$ 3, 5 $\times$ 5, 7 $\times$ 7, and 9 $\times$ 9 pixels, noted as $T_3(x,y)$, $T_5(x,y)$, $T_7(x,y)$, and $T_9(x,y)$, respectively. In this case, the variables p and q in Section 3.2 are chosen as $p = q =$ 3, 5, 7, and 9. Each template $T_i(x,y)$, $i =$ 3, 5, 7, and 9, has eight directional templates $T_{i,\mathcal{D}}(x,y)$, $\mathcal{D} =$ 1, 2, 3, ..., 8. The template coefficient λ defined in Section 3.2 is chosen as $\lambda = 1$. In order to reduce the required number of computations, the results pertaining for angles from π to 2π are obtained by

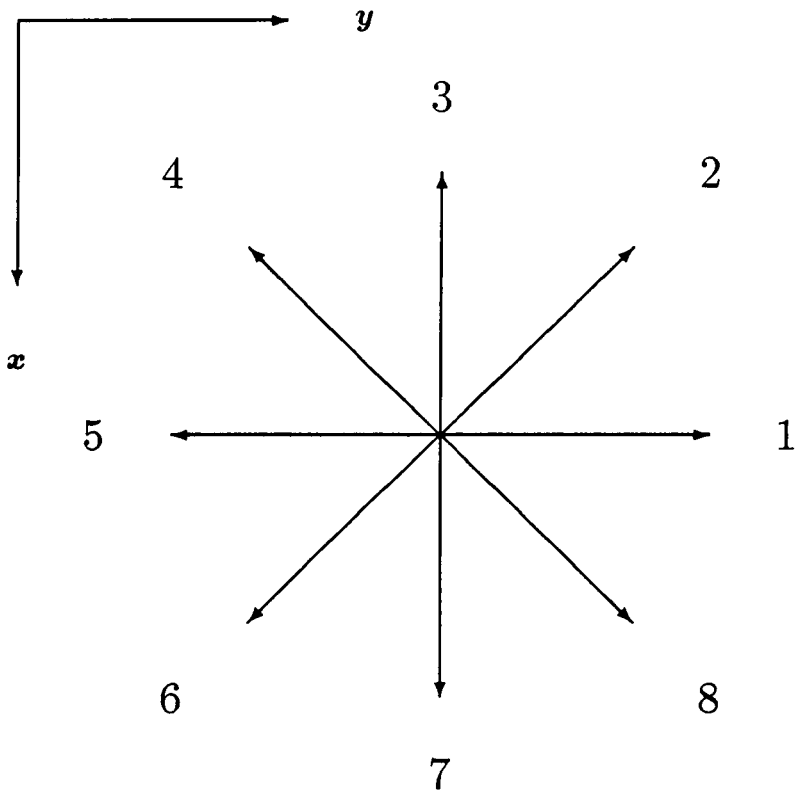


Figure 4.1: The edge orientation system.

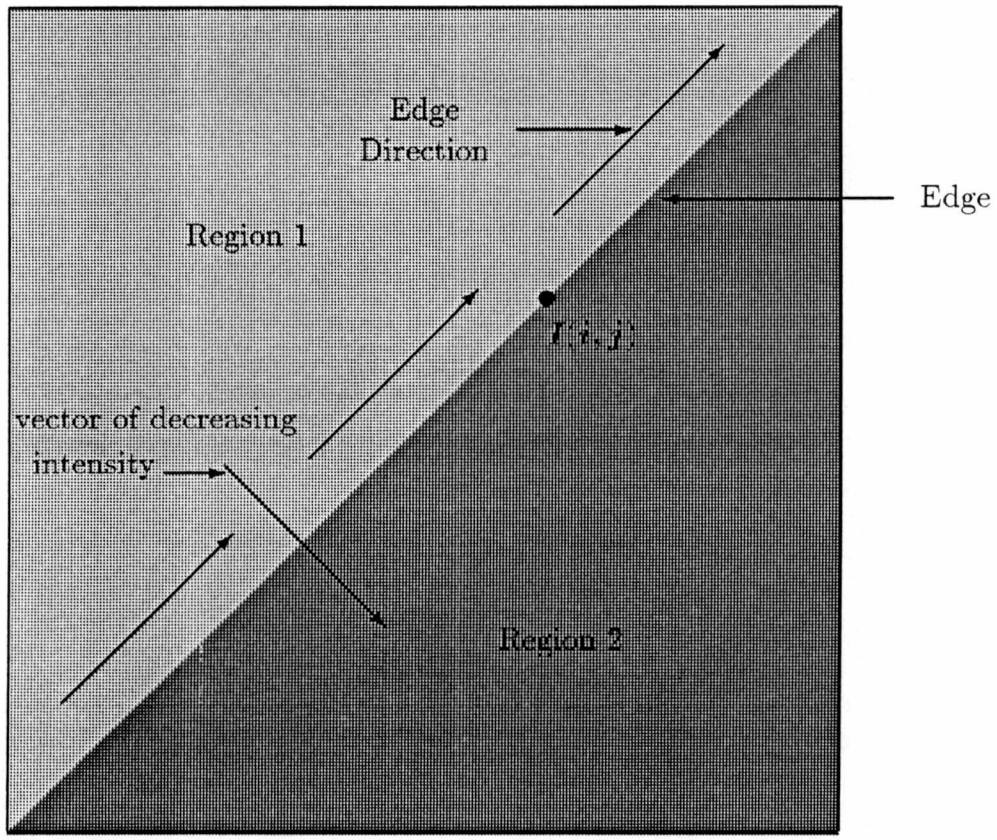


Figure 4.2: The edge direction assignment scheme.

multiplying the complementary angle results by -1. Therefore, the role of the four directional templates $T_{i,5}(x,y)$, $T_{i,6}(x,y)$, $T_{i,7}(x,y)$, and $T_{i,8}(x,y)$ can be performed by $-T_{i,1}(x,y)$, $-T_{i,2}(x,y)$, $-T_{i,3}(x,y)$, and $-T_{i,4}(x,y)$, respectively. Hence, four directional templates are enough to describe the eight directions in the orientation system. The four directional templates with size 5×5 pixels are shown in Figure 4.3.

To determine the edge strength and direction of an image pixel $I(i,j)$, several rule-based spatial measurements are taken. This is accomplished by applying multiple templates centered at the image pixel $I(i,j)$. When considering a particular template size at an image pixel $I(i,j)$, each of its four directional templates produces a magnitude output. The maximum absolute magnitude within these four magnitude values and the associated edge direction are assigned to image pixel $I(i,j)$ for the particular template size.

4.1 Three-Template Edge Detection Algorithm

If we choose three adjacent template sizes, two combinations can be formed: 3×3, 5×5, and 7×7 pixels; and 5×5, 7×7, and 9×9 pixels. The following subsections describe the decision rules pertaining to the magnitude and the direction relationships in the three-template case.

1	1	1	1	1
1	1	1	1	1
0	0	0	0	0
-1	-1	-1	-1	-1
-1	-1	-1	-1	-1

(a)

1	1	1	1	0
1	1	1	0	-1
1	1	0	-1	-1
1	0	-1	-1	-1
0	-1	-1	-1	-1

(b)

1	1	0	-1	-1
1	1	0	-1	-1
1	1	0	-1	-1
1	1	0	-1	-1
1	1	0	-1	-1

(c)

0	-1	-1	-1	-1
1	0	-1	-1	-1
1	1	0	-1	-1
1	1	1	0	-1
1	1	1	1	0

(d)

Figure 4.3: The four directional templates of template $T_5(x, y)$ for detecting the edges with directions $\mathcal{D} = 1$, 2, 3, and 4 if the products are positive or $\mathcal{D} = 5$, 6, 7, and 8 if negative. (a) $T_{5,1}(x, y)$. (b) $T_{5,2}(x, y)$. (c) $T_{5,3}(x, y)$. (d) $T_{5,4}(x, y)$.

4.1.1 Magnitude Relationship

Recalling the template operation described in the previous section for a template centered at an image pixel $I(i,j)$, we obtain a magnitude and a direction associated with this image pixel. We denote the magnitude values produced by the first, second, and third template sizes centered at an image pixel $I(i,j)$ as $MAX1$, $MAX2$, and $MAX3$, respectively. The image pixel $I(i,j)$ is considered an edge pixel if both of the following rules are satisfied.

Rule 1 : $min(|MAX1|,\ |MAX2|,\ |MAX3|) > 0.$

Rule 2 : $min(|MAX3| - |MAX2|,\ |MAX2| - |MAX1|) > T_{edge},$

where $T_{edge} > 0$ is the minimum edge magnitude increment threshold between two adjacent template sizes.

In the first rule, if any one of the three absolute magnitude values $MAX1$, $MAX2$, and $MAX3$ is zero, we assume that the image pixel $I(i,j)$ is in a smooth area, and all of the non-zero magnitude values are produced by noise. The second rule indicates that the absolute magnitude value should increase as the template size increases, and the increment between two adjacent template sizes should be larger than the threshold T_{edge}.

To estimate the value of the threshold T_{edge}, we examine the directional templates which have the same edge direction $\mathcal{D} = 1$ but different template sizes. In the definition of directional templates described in Section 3.2, the template coefficients of the template edges are zeros. When the template size increases, *i.e.*, from 3×3 to 5×5 to 7×7 pixels, we obtain 7, 11, and 15 extra pairs of image intensity differences from two sides of the common template edge, as shown in Figure 4.4.

Therefore, as a conservative assumption, we should obtain at least 7 and 11 extra pairs of image intensity differences when the template size increases in the two three-template algorithms: 3 – 5 – 7 templates and 5 – 7 – 9 templates, respectively. We assume that if an image pixel is on an edge, the difference in the intensity between its two neighbors lying across the edge should be greater than the minimum intensity difference α_{edge}; otherwise, the pixel is assumed to be in a smooth area. Hence, the threshold T_{edge} is assigned as $T_{edge} = 7 \cdot \alpha_{edge}$. We choose the value of α_{edge} in this thesis as 10, hence, $T_{edge} = 70$ for the 3 – 5 – 7 template algorithm and $T_{edge} = 110$ for the 5 – 7 – 9 template algorithm.

The threshold T_{edge} is the parameter used to adjust the sensitivity of the algorithm to noise. With a smaller T_{edge} than the one described above, we may get more details from the original image, however most of the details are produced by the noise. With a larger T_{edge} than the one we described above, we may lose small important details of the image.

2	3	4	5	6
1	1	1	1	7
0	0	0	0	0
1	-1	-1	-1	7
2	3	4	5	6

(a)

3	4	5	6	7	8	9
2	1	1	1	1	1	10
1	1	1	1	1	1	11
0	0	0	0	0	0	0
1	-1	-1	-1	-1	-1	11
2	-1	-1	-1	-1	-1	10
3	4	5	6	7	8	9

(b)

4	5	6	7	8	9	10	11	12
3	1	1	1	1	1	1	1	13
2	1	1	1	1	1	1	1	14
1	1	1	1	1	1	1	1	15
0	0	0	0	0	0	0	0	0
1	-1	-1	-1	-1	-1	-1	-1	15
2	-1	-1	-1	-1	-1	-1	-1	14
3	-1	-1	-1	-1	-1	-1	-1	13
4	5	6	7	8	9	10	11	12

(c)

Figure 4.4: The construction of two concentric directional templates with adjacent template sizes and common edge direction $\mathcal{D} = 1$. (a) Template sizes 3×3 and 5×5 pixels. (b) Template sizes 5×5 and 7×7 pixels. (c) Template sizes 7×7 and 9×9 pixels.

4.1.2 Direction Relationship

The direction of an edge pixel $I(i,j)$ (for a specific template size) is determined as discussed in Section 4.3. For the purpose of further discussion, we let the directions of the image pixel $I(i,j)$ associated with the first, second, and third template sizes be $\mathcal{D}_1$, $\mathcal{D}_2$, and $\mathcal{D}_3$, respectively. To simplify the description of the rules, we set the direction associated with the first template size to 1, *i.e.*, $\mathcal{D}_1 = 1$. When considering an image pixel $I(i,j)$, if the directions associated with the second and the third template sizes are $\mathcal{D}_2 = 2$ and $\mathcal{D}_3 = 3$, respectively, we denote this situation as 1-2-3. As shown in Figure 4.1, for notation 1-2-3, the difference in directions between the first and second templates is 45 degrees, as is the difference between the second and third templates. In addition, the angles are increasing. Each pixel is assigned an ***edge confidence***, $\mathcal{E}$, to indicate the probability that an image pixel is an edge pixel. The larger the value of $\mathcal{E}$, the more likely the image pixel is an edge pixel. Hence, a specific edge confidence threshold, T_{ec}, can be used to identify the true edge pixels.

After the directions of the three template sizes centered at the image pixel $I(i,j)$ have been assigned, the following rules assign the edge confidence value to the pixel.

1) $\mathcal{E} = 1.0$ if $\mathcal{D}_1 = \mathcal{D}_2 = \mathcal{D}_3$ (*e.g.*, 1-1-1).

2) $\mathcal{E} = 0.95$ if either

$$\mathcal{D}_1 = \mathcal{D}_2 \text{ and } |\mathcal{D}_2 - \mathcal{D}_3| = 1 \vee 7$$

or

$$\mathcal{D}_2 = \mathcal{D}_3 \text{ and } |\mathcal{D}_1 - \mathcal{D}_2| = 1 \vee 7,$$

where symbol $\vee$ means the operator "or" (*e.g.*, 1-1-2).

3) $\mathcal{E} = 0.9$ if either

$$\mathcal{D}_1 = \mathcal{D}_2 \text{ and } |\mathcal{D}_2 - \mathcal{D}_3| = 2 \vee 6$$

or

$$\mathcal{D}_2 = \mathcal{D}_3 \text{ and } |\mathcal{D}_1 - \mathcal{D}_2| = 2 \vee 6$$

(*e.g.*, 1-1-3).

4) $\mathcal{E} = 0.85$ if

$$\mathcal{D}_1 = \mathcal{D}_3 \text{ and } |\mathcal{D}_1 - \mathcal{D}_2| = 1 \vee 7$$

(*e.g.*, 1-2-1).

5) $\mathcal{E} = 0.8$ if $\mathcal{D}_1 \neq \mathcal{D}_2 \neq \mathcal{D}_3$ and either

$$\mathcal{D}_1 - \mathcal{D}_2 = \mathcal{D}_2 - \mathcal{D}_3 = -1 \vee 7$$

or

$$\mathcal{D}_1 - \mathcal{D}_2 = \mathcal{D}_2 - \mathcal{D}_3 = 1 \vee -7$$

(*e.g.*, 1-2-3).

The rules for assigning edge confidence in the three-template cases with the direction of the first template size fixed to $\mathcal{D}_1 = 1$ are shown in Table 4.1. It is possible to design additional rules for situations with edge confidence $\mathcal{E} < 0.8$. However, in practice, the edge pixels with $\mathcal{E} \geq 0.8$ are of interest, and the image pixels with $\mathcal{E} < 0.8$ are assumed to be in smooth areas.

A special case for assigning a direction to an image pixel $I(i,j)$ for a specific template size occurs when more than one directional template produces the absolute maximum magnitude. Choosing a random direction associated with one of these directional templates, which produces the absolute maximum magnitude, as the direction of the image pixel $I(i,j)$, may cause a low $\mathcal{E}$ value to be assigned to the image pixel. Therefore, in order to assign the maximum $\mathcal{E}$ value to the image pixel, all candidate directions are examined under the following conditions.

Condition 1 : $\mathcal{D}_1 = \mathcal{D}_2 = \mathcal{D}_3$.

Condition 2 : $\mathcal{D}_1 = \mathcal{D}_2$.

Condition 3 : $\mathcal{D}_2 = \mathcal{D}_3$.

Condition 4 : $\mathcal{D}_1 = \mathcal{D}_3$.

Within the candidate directions, the direction that satisfies the condition having the greatest constraint of the four conditions; *i.e.*, condition 1 has greater

Table 4.1: The edge confidence assignment rules in the three-template algorithm.

	Directions $\mathcal{D}_1$ - $\mathcal{D}_2$ - $\mathcal{D}_3$	Edge Confidence $\mathcal{E}$
1	$\underline{1}$ - 1 - 1	1.00
2	$\underline{1}$ - 1 - 2 or $\underline{1}$ - 1 - 8	0.95
	1 - $\underline{2}$ - 2 or 1 - $\underline{8}$ - 8	0.95
3	$\underline{1}$ - 1 - 3 or $\underline{1}$ - 1 - 7	0.90
	1 - $\underline{3}$ - 3 or 1 - $\underline{7}$ - 7	0.90
4	$\underline{1}$ - 2 - 1 or $\underline{1}$ - 8 - 1	0.85
5	1 - $\underline{2}$ - 3 or 1 - $\underline{8}$ - 7	0.80

constraint than condition 2, condition 2 has greater constraint than condition 3, and so forth; is appointed as the direction of the image pixel $I(i,j)$. If none of the directions satisfies any one of the above conditions, the direction associated with the first computed directional template which has the absolute maximum magnitude is assigned as the direction of the image pixel $I(i,j)$.

4.1.3 Edge Strength and Direction Assignment

When the examinations of the magnitude and the direction relationships are complete, each qualified edge pixel is assigned an edge strength and direction. The edge strength of an edge pixel is the average of its three absolute magnitude values, $|MAX1|$, $|MAX2|$, and $|MAX3|$, which are produced by the three template sizes centered at the edge pixel. The direction of an edge pixel is assigned as shown in Table 4.1.

4.2 Four-Template Edge Detection Algorithm

When the four-template case is chosen as the multi-template combination, the magnitude and the direction relationships are similar to the relationships in the three-template case. However, the relationships in this case are more complicated. As in the three-template case, the magnitude and the direction relationships in the four-template case are governed by a set of rules which are

described in the following subsections.

4.2.1 Magnitude Relationship

Using the definition similar to the one in Section 4.1.1, we denote the magnitude values produced by the 3×3, 5×5, 7×7, and 9×9 templates centered at an image pixel $I(i,j)$ as $MAX1$, $MAX2$, $MAX3$, and $MAX4$, respectively. We consider the image pixel $I(i,j)$ an edge pixel if the following rules are satisfied.

Rule 1 : $min(|MAX1|,\ |MAX2|,\ |MAX3|,\ |MAX4|) > 0.$

Rule 2 : $min(|MAX4| - |MAX3|,\ |MAX3| - |MAX2|,\ |MAX2| - |MAX1|) > T_{edge},$

where $T_{edge} > 0$ is the minimum edge magnitude increment threshold between two adjacent template sizes. In the four-template case, the threshold T_{edge} is still chosen as 70, because in this case there are 7, 11, 15, and 19 extra pairs of intensity differences, the minimum still being 7 pairs. If any one of these four absolute magnitude values $MAX1$, $MAX2$, $MAX3$, and $MAX4$ is zero, or any increment between two adjacent template sizes is smaller than the threshold T_{edge}, we assume that the image pixel $I(i,j)$ is in a smooth area, and all of the magnitude values and small increment values between two adjacent templates are produced by noise.

4.2.2 Direction Relationship

Applying a similar definition to the definition used in Section 4.1.2 for an image pixel $I(i,j)$, we let the directions associated with the four template sizes (3×3, 5×5, 7×7, and 9×9 pixels), which are centered at $I(i,j)$, be $\mathcal{D}_1$, $\mathcal{D}_2$, $\mathcal{D}_3$, and $\mathcal{D}_4$, respectively.

After all of the four directions of the image pixel $I(i,j)$ have been assigned, the following rules are considered to find the edge confidence of the image pixel.

1) $\mathcal{E} = 1.0$ if $\mathcal{D}_1 = \mathcal{D}_2 = \mathcal{D}_3 = \mathcal{D}_4$ (*e.g.*, 1-1-1-1).

2) $\mathcal{E} = 0.95$ if either

$$\mathcal{D}_2 = \mathcal{D}_3 = \mathcal{D}_4 \text{ and } |\mathcal{D}_1 - \mathcal{D}_2| = 1 \vee 7$$

or

$$\mathcal{D}_1 = \mathcal{D}_2 = \mathcal{D}_3 \text{ and } |\mathcal{D}_3 - \mathcal{D}_4| = 1 \vee 7$$

(*e.g.*, 1-1-1-2).

3) $\mathcal{E} = 0.9$ if $\mathcal{D}_1 = \mathcal{D}_2$, $\mathcal{D}_3 = \mathcal{D}_4$, and

$$|\mathcal{D}_2 - \mathcal{D}_3| = 1 \vee 7$$

(*e.g.*, 1-1-2-2).

4) $\mathcal{E} = 0.85$ if either

$$\mathcal{D}_1 = \mathcal{D}_3 = \mathcal{D}_4 \text{ and } |\mathcal{D}_1 - \mathcal{D}_2| = 1 \vee 7$$

or

$$\mathcal{D}_1 = \mathcal{D}_2 = \mathcal{D}_4 \text{ and } |\mathcal{D}_1 - \mathcal{D}_3| = 1 \vee 7$$

(*e.g.*, 1-1-2-1).

5) $\mathcal{E} = 0.8$ if

$$\mathcal{D}_2 = \mathcal{D}_3 = \mathcal{D}_4 \text{ and } |\mathcal{D}_1 - \mathcal{D}_2| = 2 \vee 6$$

(*e.g.*, 1-3-3-3).

6) $\mathcal{E} = 0.75$ if $\mathcal{D}_2 = \mathcal{D}_3$ and either

$$\mathcal{D}_1 - \mathcal{D}_2 = \mathcal{D}_3 - \mathcal{D}_4 = -1 \vee 7$$

or

$$\mathcal{D}_1 - \mathcal{D}_2 = \mathcal{D}_3 - \mathcal{D}_4 = 1 \vee -7$$

(*e.g.*, 1-2-2-3).

7) $\mathcal{E} = 0.7$ if

$$\mathcal{D}_3 = \mathcal{D}_4 \text{ and } \mathcal{D}_1 - \mathcal{D}_2 = \mathcal{D}_2 - \mathcal{D}_3 = -1 \vee 1 \vee -7 \vee 7$$

(*e.g.*, 1-2-3-3).

8) $\mathcal{E} = 0.65$ if $\mathcal{D}_1 = \mathcal{D}_3$, $\mathcal{D}_2 = \mathcal{D}_4$, and

$$|\mathcal{D}_2 - \mathcal{D}_3| = 1 \vee 7$$

(*e.g.*, 1-2-1-2).

9) $\mathcal{E} = 0.6$ if $\mathcal{D}_1 = \mathcal{D}_4$, $\mathcal{D}_2 = \mathcal{D}_3$, and

$$|\mathcal{D}_1 - \mathcal{D}_2| = 1 \vee 7$$

(*e.g.*, 1-2-2-1).

10) $\mathcal{E} = 0.55$ if $\mathcal{D}_3 = \mathcal{D}_4$ and either

$$\mathcal{D}_1 - \mathcal{D}_2 = -2 \vee 6 \text{ and } \mathcal{D}_2 - \mathcal{D}_3 = 1 \vee -7$$

or

$$\mathcal{D}_1 - \mathcal{D}_2 = -6 \vee 2 \text{ and } \mathcal{D}_2 - \mathcal{D}_3 = -1 \vee 7$$

(*e.g.*, 1-3-2-2).

11) $\mathcal{E} = 0.5$ if $\mathcal{D}_2 = \mathcal{D}_3$ and either

$$\mathcal{D}_1 - \mathcal{D}_2 = -2 \vee 6 \text{ and } \mathcal{D}_3 - \mathcal{D}_4 = 1 \vee -7$$

or

$$\mathcal{D}_1 - \mathcal{D}_2 = -6 \vee 2 \text{ and } \mathcal{D}_3 - \mathcal{D}_4 = -1 \vee 7$$

(*e.g.*, 1-3-3-2).

The rules for edge confidence assignment in the four-template case with the direction of the first template size fixed to $\mathcal{D}_1 = 1$ are shown in Table 4.2. The rules with $\mathcal{E} < 0.5$ can be formed in a similar manner. The image pixels with $\mathcal{E}$ values smaller than $T_{ec} = 0.5$ are assumed to be in smooth areas.

Table 4.2: The edge confidence assignment rules in the four-template algorithm.

	Directions $\mathcal{D}_1-\mathcal{D}_2-\mathcal{D}_3-\mathcal{D}_4$	Edge Confidence $\mathcal{E}$
1	$\underline{1}-1-1-1$	1.00
2	$1-\underline{2}-2-2$ or $1-\underline{8}-8-8$	0.95
	$\underline{1}-1-1-2$ or $\underline{1}-1-1-8$	0.95
3	$\underline{1}-1-2-2$ or $\underline{1}-1-8-8$	0.90
4	$\underline{1}-2-1-1$ or $\underline{1}-8-1-1$	0.85
	$\underline{1}-1-2-1$ or $\underline{1}-1-8-1$	0.85
5	$1-\underline{3}-3-3$	0.80
6	$1-\underline{2}-2-3$ or $1-\underline{8}-8-7$	0.75
7	$1-2-\underline{3}-3$ or $1-8-\underline{7}-7$	0.70
8	$1-\underline{2}-1-2$ or $1-\underline{8}-1-8$	0.65
9	$1-\underline{2}-2-1$ or $1-\underline{8}-8-1$	0.60
10	$1-3-\underline{2}-2$ or $1-7-\underline{8}-8$	0.55
11	$1-\underline{3}-3-2$ or $1-\underline{7}-7-8$	0.50

The rules to determine the direction of the edge at an image pixel $I(i,j)$, where more than one directional template is producing the absolute maximum magnitude, are similar to the rules in the case where three templates are used, as discussed in Section 4.1.2. The rules for the case where four templates are used are indicated by the following conditions.

Condition 1 : $\mathcal{D}_1 = \mathcal{D}_2 = \mathcal{D}_3 = \mathcal{D}_4$.

Condition 2 : $\mathcal{D}_1 = \mathcal{D}_2 = \mathcal{D}_3$.

Condition 3 : $\mathcal{D}_2 = \mathcal{D}_3 = \mathcal{D}_4$.

Condition 4 : $\mathcal{D}_1 = \mathcal{D}_2 = \mathcal{D}_4$.

Condition 5 : $\mathcal{D}_1 = \mathcal{D}_3 = \mathcal{D}_4$.

Condition 6 : $\mathcal{D}_1 = \mathcal{D}_2$.

Condition 7 : $\mathcal{D}_2 = \mathcal{D}_3$.

Condition 8 : $\mathcal{D}_3 = \mathcal{D}_4$.

Condition 9 : $\mathcal{D}_1 = \mathcal{D}_3$.

Condition 10 : $\mathcal{D}_2 = \mathcal{D}_4$.

Condition 11 : $\mathcal{D}_1 = \mathcal{D}_4$.

Within the directions associated with the absolute maximum magnitude, the direction that satisfied the condition having the greatest constraint of the eleven

conditions; *i.e.*, condition 1 has greater constraint than condition 2, condition 2 has greater constraint than condition 3, and so forth; is appointed as the direction of the image pixel $I(i,j)$. If none of the directions satisfies any one of the above conditions, the direction associated with the first computed directional template, which has the absolute maximum magnitude, is assigned as the direction of the image pixel $I(i,j)$ for that template size.

4.2.3 Edge Strength and Direction Assignment

When the examinations of the two relationships are complete, each qualified edge pixel is assigned a final edge strength and direction. The edge strength of an edge pixel is the average of its four absolute maximum magnitude values, $|MAX1|$, $|MAX2|$, $|MAX3|$, and $|MAX4|$, which are produced by the four template sizes centered at the edge pixel. The direction assignment is shown in Table 4.2.

4.3 Postprocessing

After the qualified edges of a given image are found, postprocessing is implemented for the purpose of edge enhancement, isolated holes filling, and weak edge removal. For each edge pixel, denote the edge strength of the edge pixel and its two neighbors, which lie along and against the direction of the edge pixel,

as S_0, S_l, and S_g, respectively, as shown in Figure 4.5, and their directions as $\mathcal{D}_0$, $\mathcal{D}_l$, and $\mathcal{D}_g$, respectively. Postprocessing involves the following rules:

Rule 1 : The average of S_l and S_g is assigned as the edge strength of the edge pixel $I(i,j)$ if $\mathcal{D}_0 = \mathcal{D}_l = \mathcal{D}_g$ and $S_0 < min\{S_l, S_g\}$.

Rule 2 : The average of S_l and S_g is assigned as the edge strength of the edge pixel if $S_0 < min\{S_l, S_g\}$ and either

$$\mathcal{D}_0 = \mathcal{D}_l \text{ and } |\mathcal{D}_0 - \mathcal{D}_g| = 1 \vee 7$$

or

$$\mathcal{D}_0 = \mathcal{D}_g \text{ and } |\mathcal{D}_0 - \mathcal{D}_l| = 1 \vee 7$$

(*e.g.*, 1-1-2 or 1-2-2).

Rule 3 : If the edge strength of the pixel $I(i,j)$ is zero and all of its eight neighbors are non-zero edge pixels, the average edge strength and the major direction of the eight none-zero neighbors are assigned the edge strength and direction of the pixel $I(i,j)$.

Rule 4 : If the thickness of an edge is one pixel, the edge strength of that pixel is assigned as zero.

The objective of rule 1 and rule 2 is to enhance the weak edge pixels that well agree in direction with their neighbors. Rule 3 is implemented to fill the one-pixel-wide isolated holes. The aim of rule 4 is to merge weak contours with their

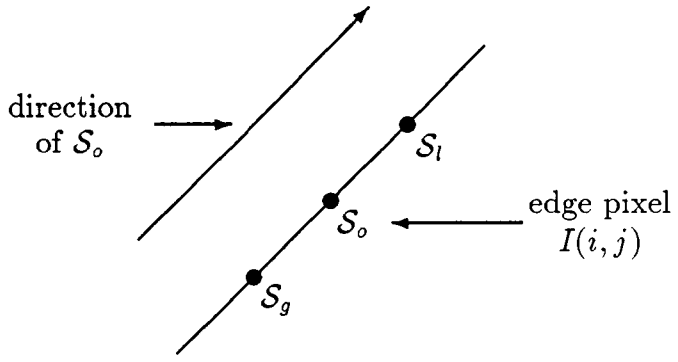


Figure 4.5: The assignment of $\mathcal{S}_l$, $\mathcal{S}_0$, and $\mathcal{S}_g$ to the edge pixel $I(i,j)$.

adjacent contours by breaking the one-pixel-wide edges between them. Because weak contours are always small, rule 4 (after thinning) results in neat edges that clearly define the object boundaries.

4.4 Thinning Algorithm

The thinning algorithm implemented here is a gray-scale, non-maximum deletion algorithm which combines both the edge strength and the direction relationships to extract 4-connected one-pixel-wide edges. The algorithm is a two-stage process. The first stage is the assignment of the likelihood of being an edge (**LBE**), which is based upon Lacroix's module [57]. The second stage is the contour following, which is based upon Kunt's method [56]. In the second stage, Kunt's algorithm is modified to generate 4-connected edges rather than

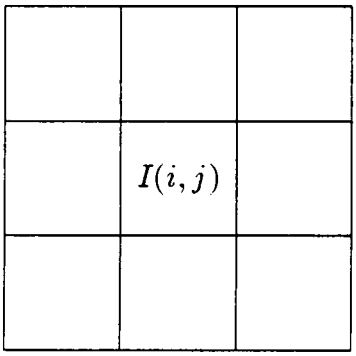
8-connected edges. At an edge pixel, each of these two stages utilizes both the edge strength and the direction relationships of the edge pixel and its neighbors. The details of the procedure are described in the following subsections.

4.4.1 LBE Assignment

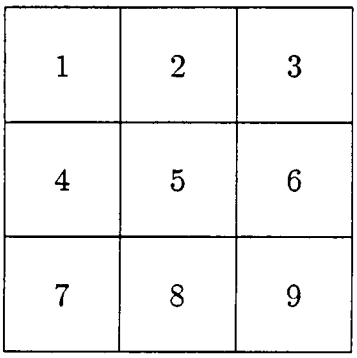
The purpose of this stage is to assign an LBE value to each edge pixel. The LBE with a value between 0 and 1 is a measurement of probability that determines whether an edge pixel is on the edge strength ridge of a given image or not. The procedure is a non-maximum deletion because the pixels with LBE $= 0$ are removed, the pixels with LBE $= 1$ are defined as final edge pixels, and the status of pixels with $0 <$ LBE < 1 is determined by further processing their local contextual information.

In this stage, each edge pixel $I(i,j)$ of the unthinned edge image is assigned two counters: $u(i,j)$ and $v(i,j)$; the counter $v(i,j)$ measures how many times the pixel $I(i,j)$ is visited, and the counter $u(i,j)$ measures how many times the pixel $I(i,j)$ is a local maximum when it is visited. When the unthinned edge image is scanned at each edge pixel, a 3×1 mask is placed so that the center of the mask coincides with the edge pixel and the other two pixels of the mask lie perpendicular to the direction of the edge pixel, as shown in Figure 4.6(c).

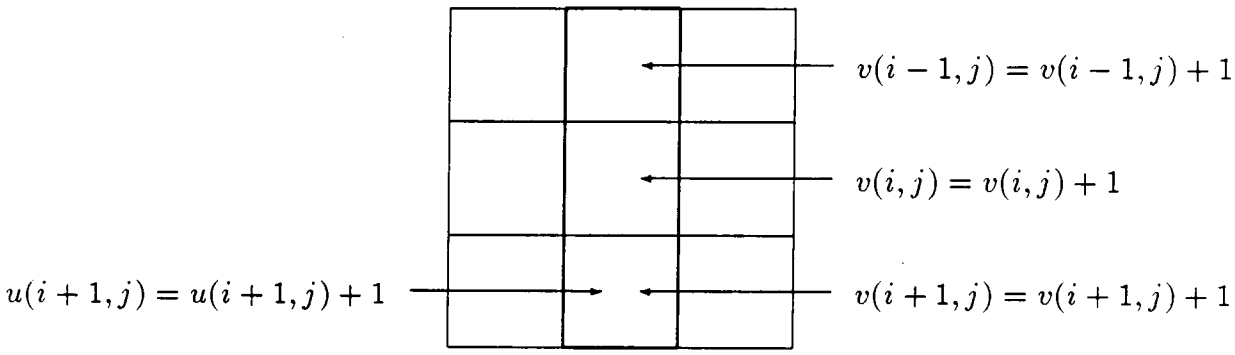
During the scanning process, all pixels in the 3×1 mask are visited, and their counters v are increased by 1. Meanwhile, only the counter u, of the pixel



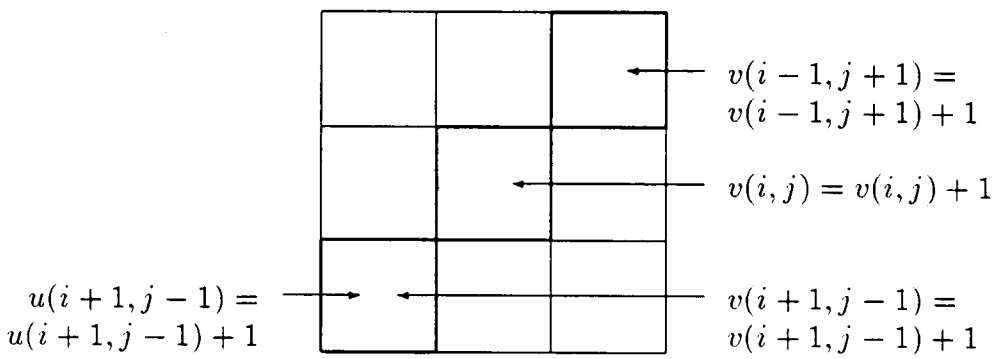
(a)



(b)



(c)



(d)

Figure 4.6: Examples of counters u and v measurement at the image pixel $I(i,j)$. (a) The edge pixel $I(i,j)$ centered at a 3×3 subimage. (b) The edge strength of the pixels in (a). (c)(d) The increment of counters u and v when the directions of the edge pixel $I(i,j)$ are $\mathcal{D} = 1$ and 8, respectively.

which has the maximum edge strength within the 3×1 mask, is increased by 1. So, at each pixel, three v counters are increased but only one u counter is increased. Two examples of using these two counters are shown in Figure 4.6. After scanning the unthinned edge image, each edge pixel has been assigned a final u value and a nonzero v value. Then, the LBE value of an edge pixel is

$$\text{LBE} = \frac{u}{v}, \tag{4.1}$$

where u and v are the u counter and v counter values associated with the edge pixel.

After the LBE assignment process, each edge pixel is assigned an LBE value between 1 and 0. The edge pixels with the maximum strength in a neighborhood have the largest LBE values, *i.e.*, LBE = 1, and the edge pixels that have the weakest strength in a neighborhood have the smallest LBE values, LBE = 0. The other non-maximum edge pixels have LBE values $0 < \text{LBE} < 1$.

4.4.2 Contour Following

The second stage, contour following, is a sequential process. Assuming that the edges are continuous and the direction changes between two adjacent edge pixels are smooth and within 45 degrees, the contour following procedure produces 4-connected one-pixel-wide edge images from the given unthinned edge images.

To avoid duplicate detection, each detected final edge pixel is marked. When

a marked edge pixel is detected, the current contour following procedure is terminated. Then, the unthinned edge image is scanned. When an unmarked edge pixel $I(i,j)$ with LBE equal to 1 is found, two contour following processes begin: the first proceeds along the direction of the edge pixel $I(i,j)$ (positive orientation) and the second proceeds in the opposite orientation (negative orientation). We denote the three candidates of the next edge pixel as l, a, and r as shown in Figure 4.7 and set the maximum value for these three candidates to C_{max}. Assuming that $C_{max} \neq 0$ and l, a, and r are unmarked, the two contour following processes are implemented according to the following rules.

Rule 1 : If n has LBE value equal to C_{max}, a becomes the next edge pixel of the contour. If any l or r has an LBE value equal to C_{max}, its LBE value is modified to 0.99.

Rule 2 : If the LBE value of a is smaller than C_{max} and only one of the l and r has an LBE value equal to C_{max}, the edge pixel with LBE value equal to C_{max} becomes the next edge pixel of the contour.

Rule 3 : If the LBE value of a is smaller than C_{max} and both l and r have LBE values equal to C_{max}, the edge pixel (l or r) whose direction is closer to the direction of the current edge pixel becomes the next edge pixel of the contour, and the LBE value of the other edge pixel is modified to 1.

Rule 4 : If the next edge pixel is not a 4-neighbor of the current edge pixel, the edge pixel of the two common 4-neighbors of these two edge pixels

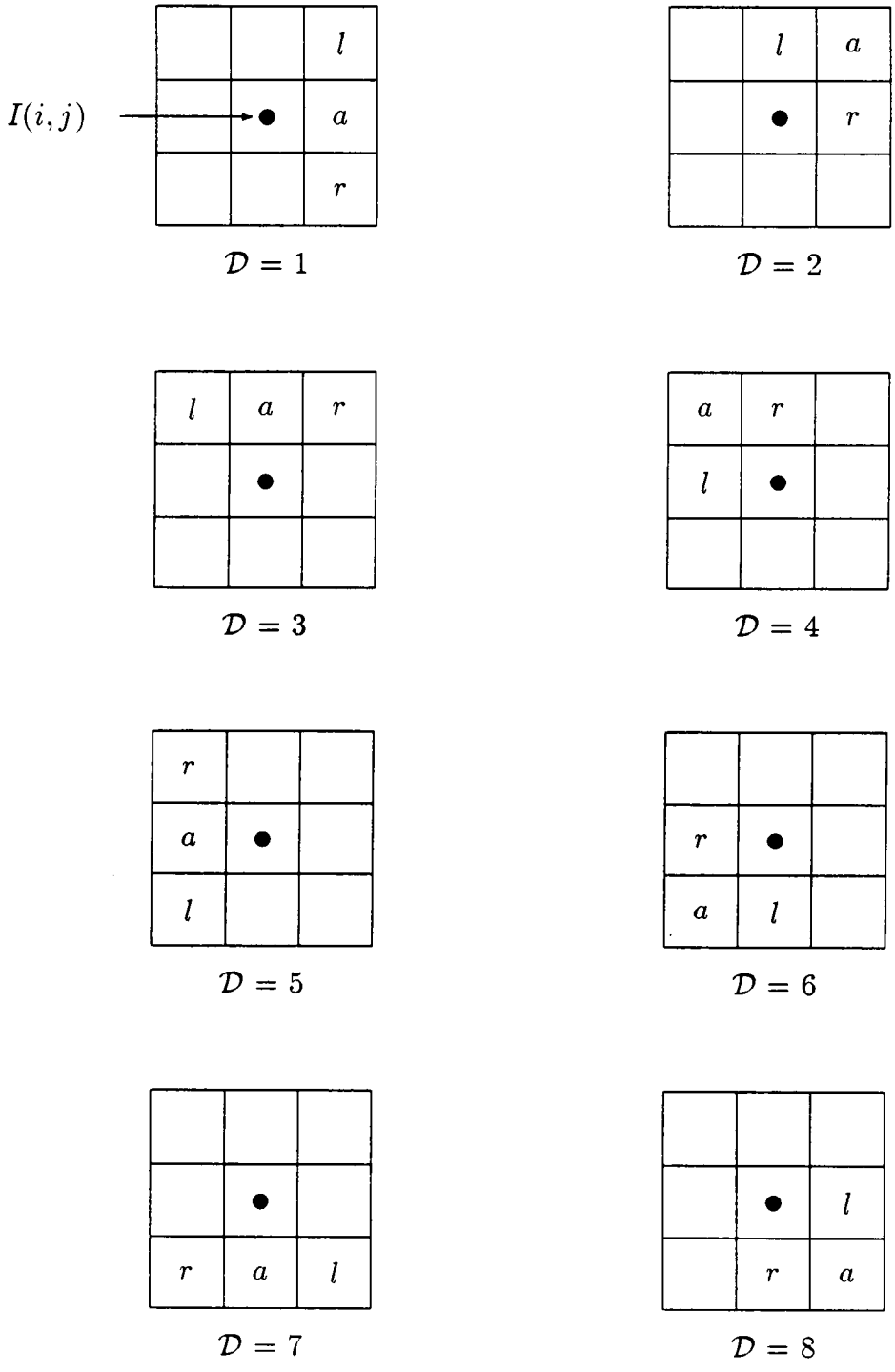


Figure 4.7: The definition of l, a, and r of the edge pixel $I(i,j)$ under the positive orientation process with the directions $\mathcal{D}$ from 1 to 8.

with the larger edge strength value is included in the contour.

Rule 5 : Other situations cause the termination of this process.

In rule 1, we modify the LBE value of the non-contour edge pixel to 0.99 to avoid duplication of detection. In rule 3, we modify the LBE value to 1 to force a new contour following procedure here; therefore, a T-type or Y-type junction can be formed. The purpose of rule 4 is to preserve the 4-connectivity of the resulting edges. The procedure is repeated until both of the contour following processes terminate.

4.4.3 Edge Refinement

After a one-pixel-wide edge image is created, it is also essential, in most cases, to refine the edges and remove the small and meaningless segments from the edge image. This process is accomplished by choosing a threshold of segment length (**TSL**) and then comparing all segment lengths of the edge image with the TSL. Any segment which has a length smaller than the TSL is removed.

The value of the TSL is critical to the final result. When a large TSL value is used, the objects with long or closed contours can be observed clearly, however details may be lost. On the contrary, with a small TSL value, the details are preserved but the objects may become ambiguous because of the numerous small segments. The value of TSL is chosen as TSL = 20 pixels in this thesis.

CHAPTER 5

EXPERIMENTAL RESULTS

In Chapter 4 of this thesis, a rule-based multi-template edge detection algorithm is proposed. In the first part of this chapter we examine the performance of the three-template and the four-template algorithms on specially designed images. Then, in the second part of this chapter, we evaluate the performance of these algorithms on complex infrared images.

5.1 Measurement of Quantitative Performance

Detection of different kinds of edges requires a flexible edge detection algorithm. Usually, such an edge detection algorithm employs adjustable parameters to provide flexibility to the algorithm. Depending on the edge complexity, intensity contrast, and noise density of a given image, these parameters can be modified to extract the largest percentage of edge pixels of interest in the image. The proposed rule-based multi-template edge detection algorithm has two adjustable parameters: the minimum increment threshold T_{edge} and the edge confidence threshold T_{ec}. In this section we examine the performance of the proposed algorithm when it is applied to synthetic images, with or without the presence of noise, by examining its quantitative performance relative to the cri-

teria discussed by Peli and Malah [84] and Abdou and Pratt [1].

5.1.1 Quantitative Measurements

The performance of an edge detector can be evaluated in terms of quantitative performance or qualitative performance. The focus of our observation is on the quantitative performance. The quantitative performance evaluation is based on the measurements proposed by Peli and Malah [84] and Adbou and Pratt [1]. The measurements of the quantitative performance take the following into consideration.

a) R_a : Percentage of edge pixels correctly detected on the ideal edge.

b) R_b : Number of detected edge pixels which do not coincide with the ideal edge (normalized by the number of pixels on the edge).

c) R_c : Noise-to-signal ratio — defined as the ratio of the number of detected edge pixels, which do not coincide with the ideal edge, to the number of detected edge pixels which coincide with the ideal edge.

d) R_d : Mean width of a detected edge — defined as the ratio of the total number of detected edge pixels to the number of ideal edge pixels.

e) R_1 : Weighted and normalized deviation of a detected edge pixel from the ideal edge

$$R_1 = \frac{1}{I_N} \sum_{i=1}^{I_A} \frac{1}{1 + \delta e^2(i)},$$

where

I_A — the actual number of detected edge pixels,

I_I — the number of edge pixels on the ideal edge,

I_N — $\max\{I_I, I_A\}$,

δ — scaling constant (taken as 1/9 in Pratt's work), and

$e(i)$ — normal distance between the ith edge pixel and the ideal edge.

f) R_2 : Average squared deviation of a detected edge pixel from the ideal edge

$$R_2 = \frac{1}{I_A} \sum_{i=1}^{I_A} e^2(i).$$

g) R_3 : Mean absolute value of the deviation

$$R_3 = \frac{1}{I_A} \sum_{i=1}^{I_A} |e(i)|.$$

By denoting the number of detected edge pixels which coincide with the ideal edge as N_a, the quantitative measurements can be represented as

a) :

$$R_a = \frac{N_a}{I_I}$$

b) :

$$R_b = \frac{I_A - N_a}{I_I}$$

c) :

$$R_c = \frac{I_A - N_a}{N_a}$$

d) :

$$R_d = \frac{I_A}{I_I}$$

e) :

$$R_1 = \frac{1}{I_N} \sum_{i=1}^{I_A} \frac{1}{1 + \delta e^2(i)}$$

f) :

$$R_2 = \frac{1}{I_A} \sum_{i=1}^{I_A} e^2(i).$$

g) :

$$R_3 = \frac{1}{I_A} \sum_{i=1}^{I_A} |e(i)|.$$

In this quantitative performance measurement, the ideal case occurs when all of the detected edge pixels are ideal edge pixels and all of the ideal edge pixels are detected. The best possible values of the ideal case are $R_a = R_d = R_1 = 1$ and $R_b = R_c = R_2 = R_3 = 0$. If the value of R_1 is smaller than 0.5, the edge detection algorithm is considered to fail.

5.1.2 Processing of Synthetic Binary Images

For the purpose of examining the quantitative performance of the proposed edge detection algorithm, two synthetic binary images with intensities 100 and 200 and size 128 × 128 pixels are created, as shown in Figures 5.1 and 5.2. Figure 5.1 contains two concentric circles with radii 10 and 25 pixels and two squares with dimensions 10 × 10 and 25 × 25 pixels. Figure 5.2 contains corners with angles ranging from 15° to 165° in 15° increments. Figures 5.3 and 5.4 show the ideal edges of the synthetic images in Figures 5.1 and 5.2, respectively. Using the threshold values defined in Sections 4.1.1, 4.1.2, 4.2.1, and 4.2.2, the detected edge pixels of Figure 5.1 are as follows: Figure 5.5(b) is generated using the 3 - 5 - 7 templates with threshold $T_{edge} = 70$ and $T_{ec} = 0.8$; Figure 5.5(c) is generated using the 5 - 7 - 9 templates with threshold $T_{edge} = 110$ and $T_{ec} = 0.8$; and Figure 5.5(d) is generated using the 3 - 5 - 7 - 9 templates with threshold $T_{edge} = 70$ and $T_{ec} = 0.5$. Figures 5.6(b), 5.6(c), and 5.6(d) show the detected edge pixels of Figure 5.2 using the proposed algorithm. For the ideal edge given by Figures 5.3 and 5.4 and the detected edge pixels given by Figures 5.5 and 5.6, the observations of the quantitative performance on the circles, squares, and corners are listed in Tables 5.1 and 5.2. It should be noted that the ideal edges, considered in this thesis, are of thickness two pixels in order to avoid introducing possible biases in edge allocation.

To consider the case with the presence of noise, the two synthetic images in

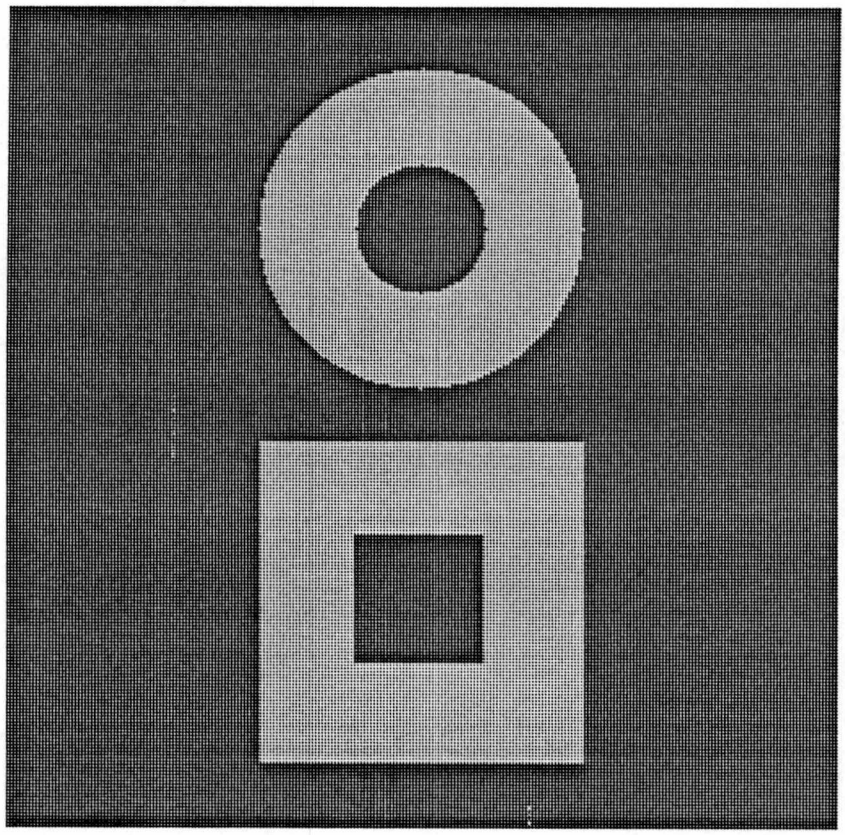


Figure 5.1: The 128 × 128 synthetic image, with binary intensities 100 and 200, in the form of two concentric circles (radii 10 and 25 pixels) and two squares (dimensions 10 × 10 and 25 × 25 pixels).

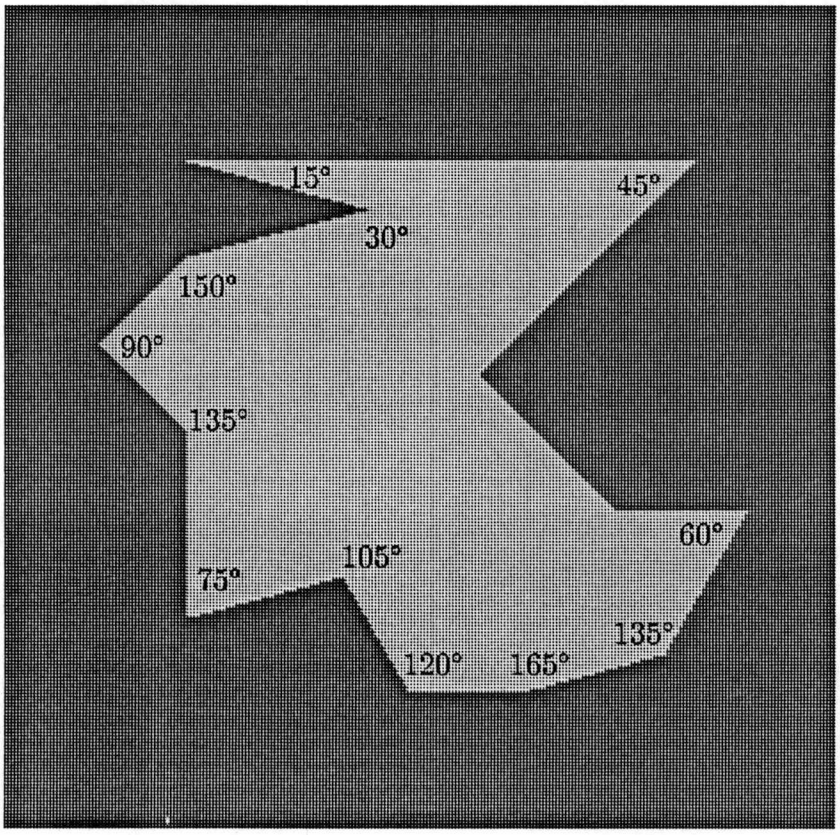


Figure 5.2: The 128 × 128 synthetic image, with binary intensities 100 and 200, which contains corner edges with angles ranging from 15° to 165° in 15° increments.

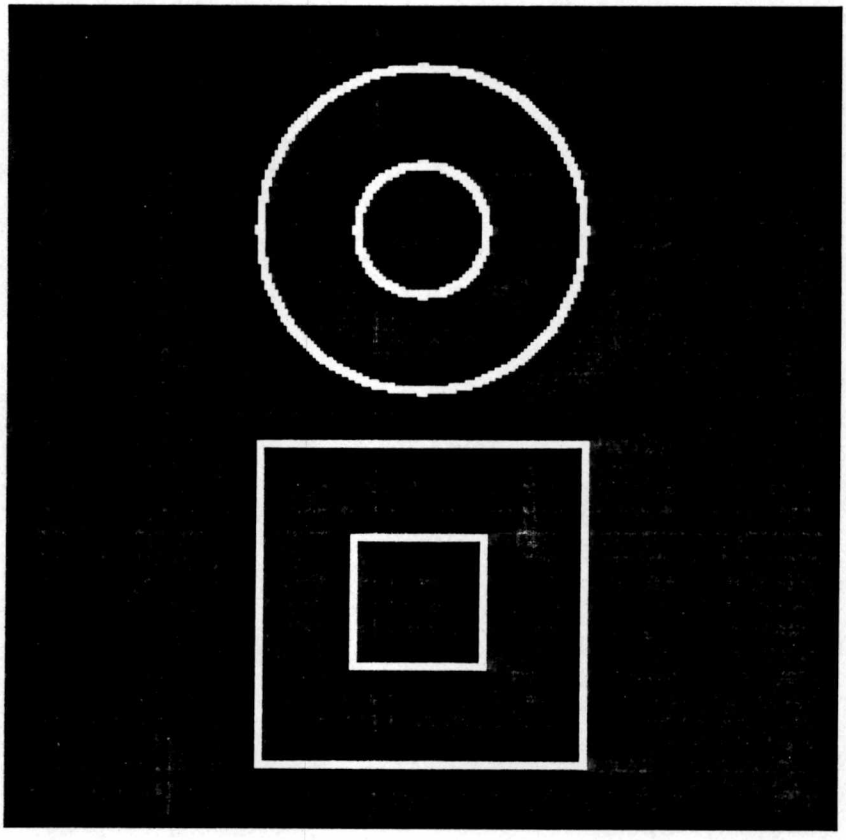


Figure 5.3: The ideal edges of Figure 5.1.

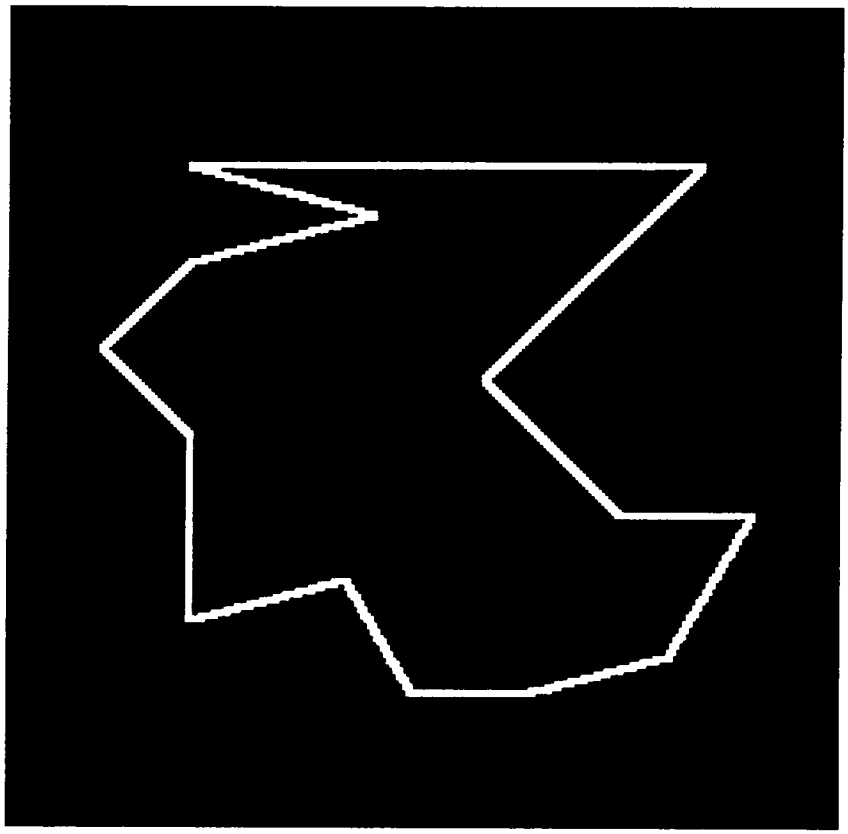
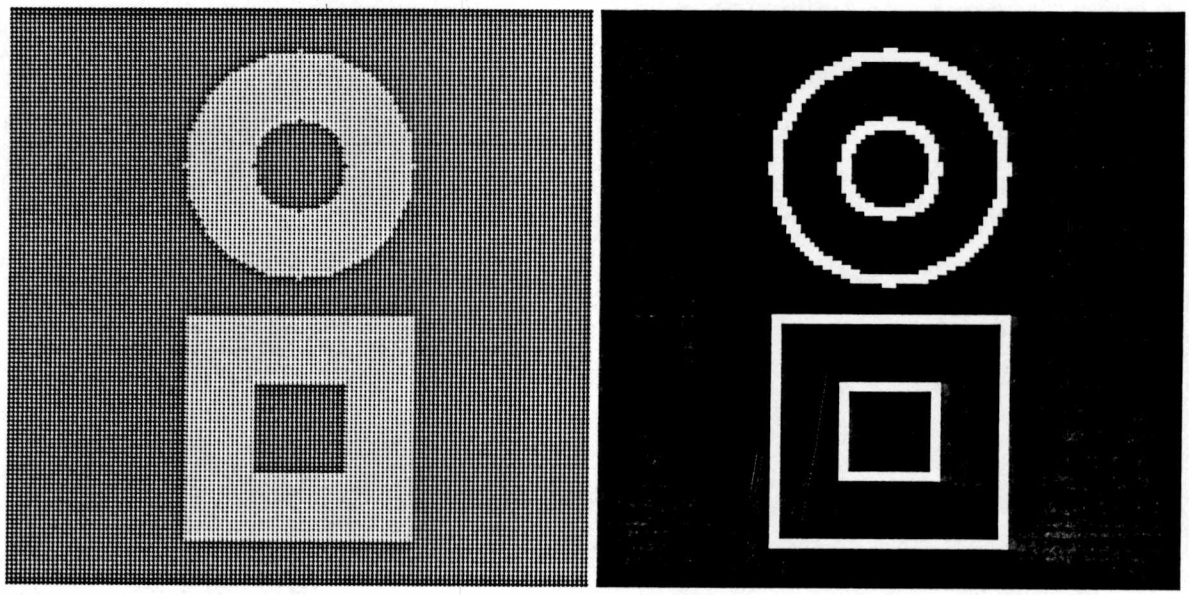
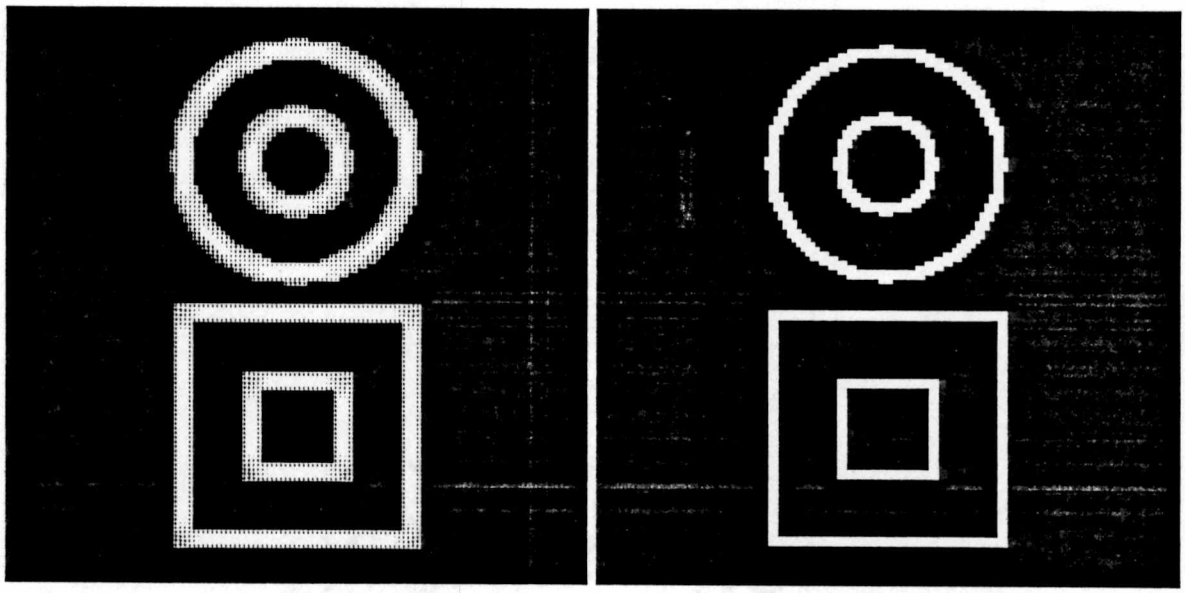


Figure 5.4: The ideal edges of Figure 5.2.



(a) (b)



(c) (d)

Figure 5.5: The detected edge pixels of Figure 5.1 using the proposed multi-template edge detection algorithm. (a) Figure 5.1. (b) Result of the 3 – 5 – 7 templates. (c) Result of the 5 – 7 – 9 templates. (d) Result of the 3 – 5 – 7 – 9 templates.

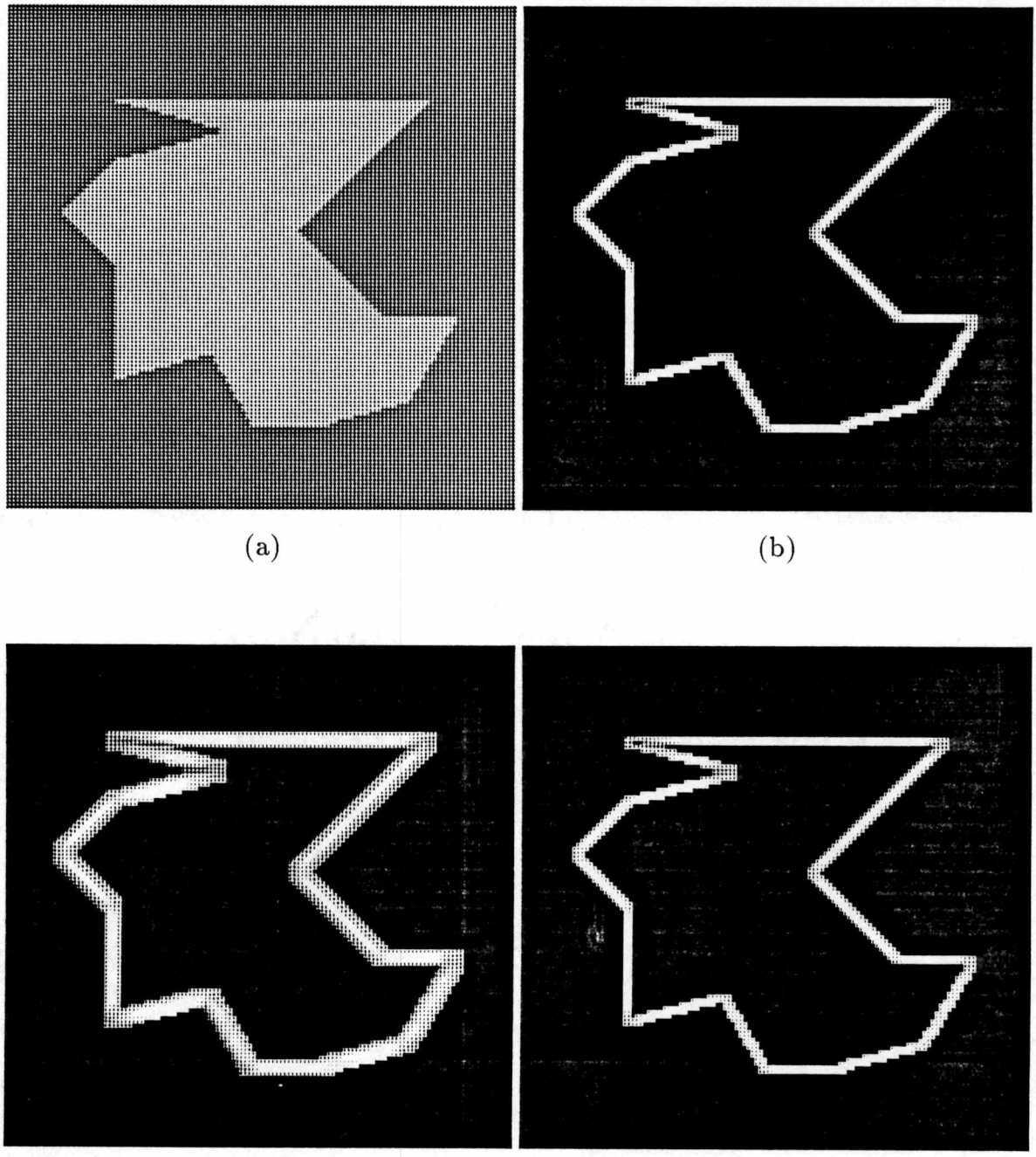


Figure 5.6: The detected edge pixels of Figure 5.2 using the proposed multi-template edge detection algorithm. (a) Figure 5.2. (b) Result of the 3 – 5 – 7 templates. (c) Result of the 5 – 7 – 9 templates. (d) Result of the 3 – 5 – 7 – 9 templates.

Table 5.1: The quantitative performance of the rule-based multi-template edge detection algorithm on the concentric circles and squares in Figure 5.1.

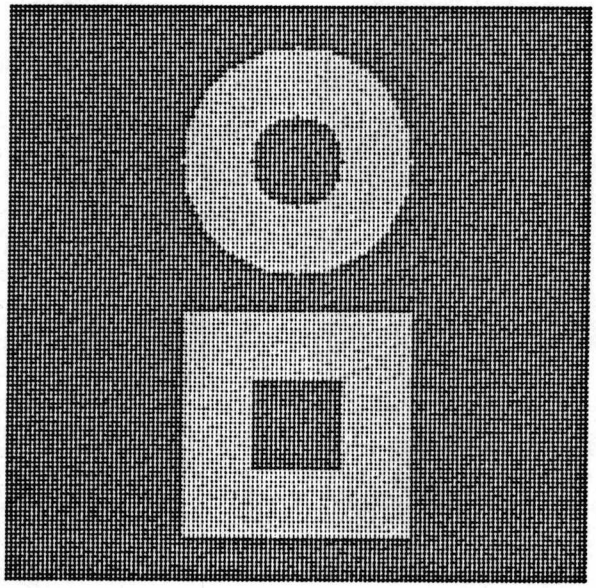
	Circles			**Squares**		
	$I_I = 568$			$I_I = 560$		$\sigma_{noise} = 0$
	3 – 5 – 7	5 – 7 – 9	3 – 5 – 7 – 9	3 – 5 – 7	5 – 7 – 9	3 – 5 – 7 – 9
R_a	1.000	1.000	1.000	1.000	1.000	1.000
R_b	0.000	0.986	0.000	0.000	1.000	0.000
R_c	0.000	0.986	0.000	0.000	1.000	0.000
R_d	1.000	1.986	1.000	1.000	2.000	1.000
R_1	1.000	0.939	1.000	1.000	0.949	1.000
R_2	0.000	0.638	0.000	0.000	0.507	0.000
R_3	0.000	0.555	0.000	0.000	0.503	0.000
I_A	568	1128	568	560	1120	560
N_a	568	568	568	560	560	560

Table 5.2: The quantitative performance of the rule-based multi-template edge detection algorithm on the corners in Figure 5.2.

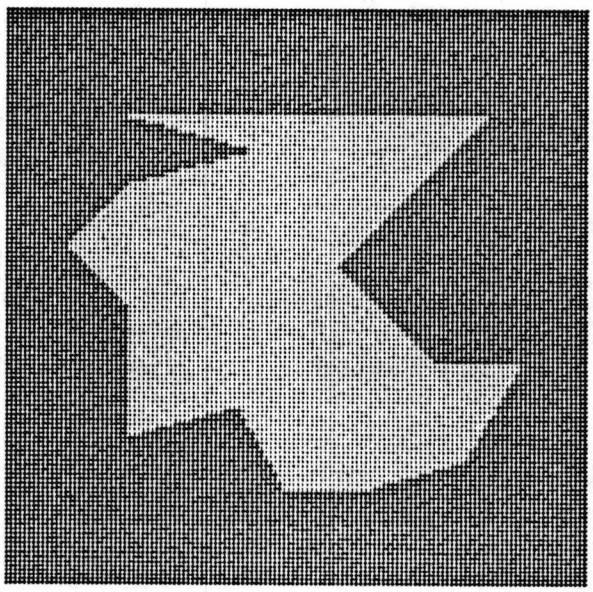
	Corners		$I_I = 982$ $\sigma_{noise} = 0$
	3 – 5 – 7	5 – 7 – 9	3 – 5 – 7 – 9
R_a	0.998	1.000	0.998
R_b	0.000	0.965	0.000
R_c	0.000	0.956	0.000
R_d	0.998	1.965	0.998
R_1	0.998	0.940	0.998
R_2	0.000	0.620	0.000
R_3	0.000	0.544	0.000
I_A	980	1930	980
N_a	980	982	980

Figures 5.1 and 5.2 are injected with additive Gaussian noise with zero mean and variance σ^2_{noise}. Figures 5.7(a) and 5.7(b) are noisy versions of Figures 5.1 and 5.2 with the standard deviation, σ_{noise}, of 10 percent of the intensity range of the original images (*i.e.*, $\sigma_{noise} = 10$). The detected edge pixels of Figures 5.7(a) and 5.7(b), using the rule-based multi-template algorithm, are shown in Figures 5.8 and 5.9, respectively. The observations of quantitative performance of the edge detection algorithm on circles, squares, and corners in the noisy images are listed in Tables 5.3 and 5.4.

Moreover, to evaluate the performance of the proposed algorithm in very noisy images, the standard deviation, σ_{noise}, of the additive Gaussian noise is raised to 20 percent of the intensity range of the original images (*i.e.*, $\sigma_{noise} = 20$). Figures 5.10(a) and 5.10(b) show the more noisy versions of Figures 5.1 and 5.2, respectively. Figures 5.11 and 5.12 present the detected edge pixels in noisy images of Figures 5.10(a) and 5.10(b), respectively. In Tables 5.5 and 5.6, the quantitative performances of the edge detection algorithm on the circles, squares, and corners in the noisy images are given.

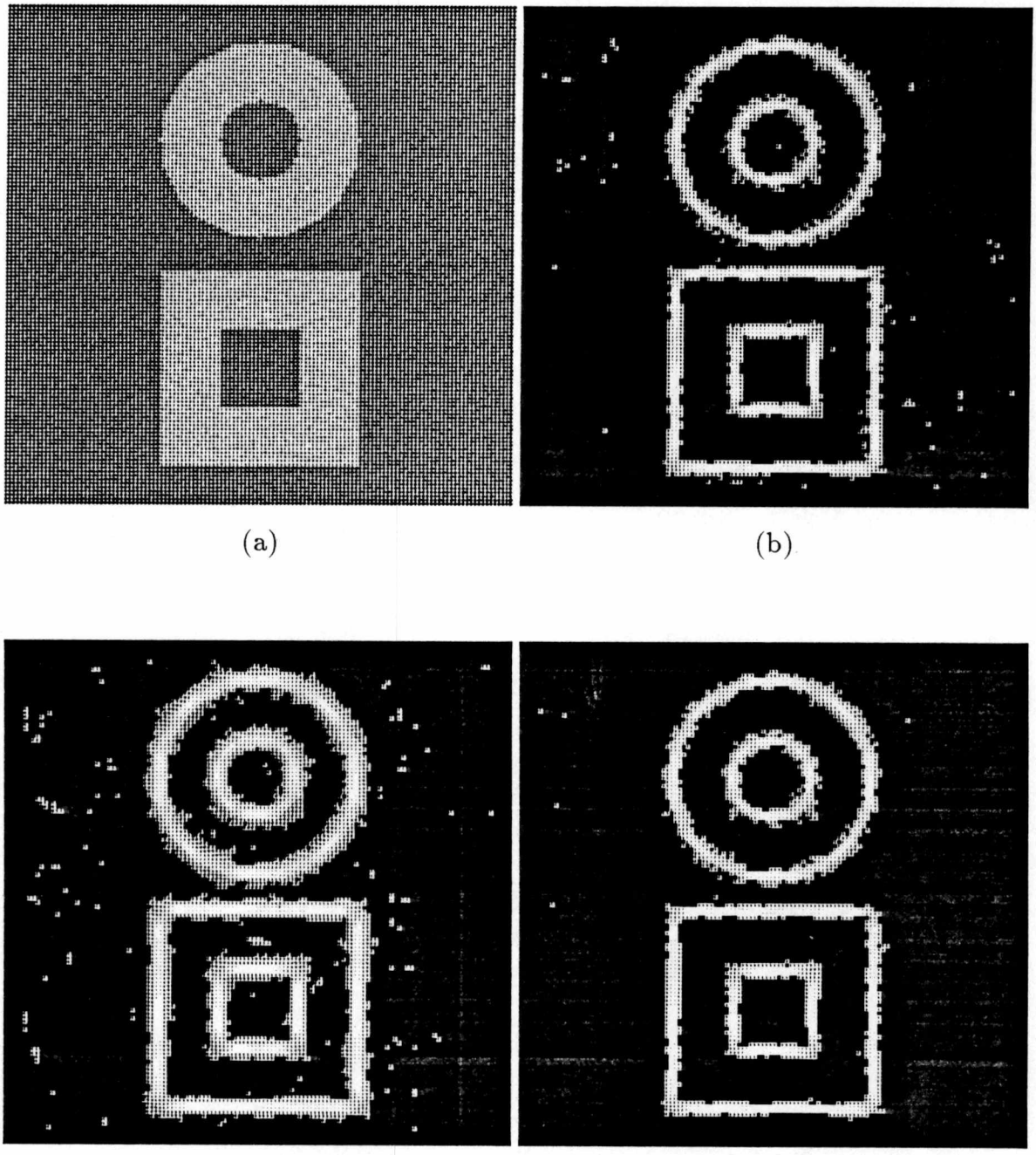


(a)



(b)

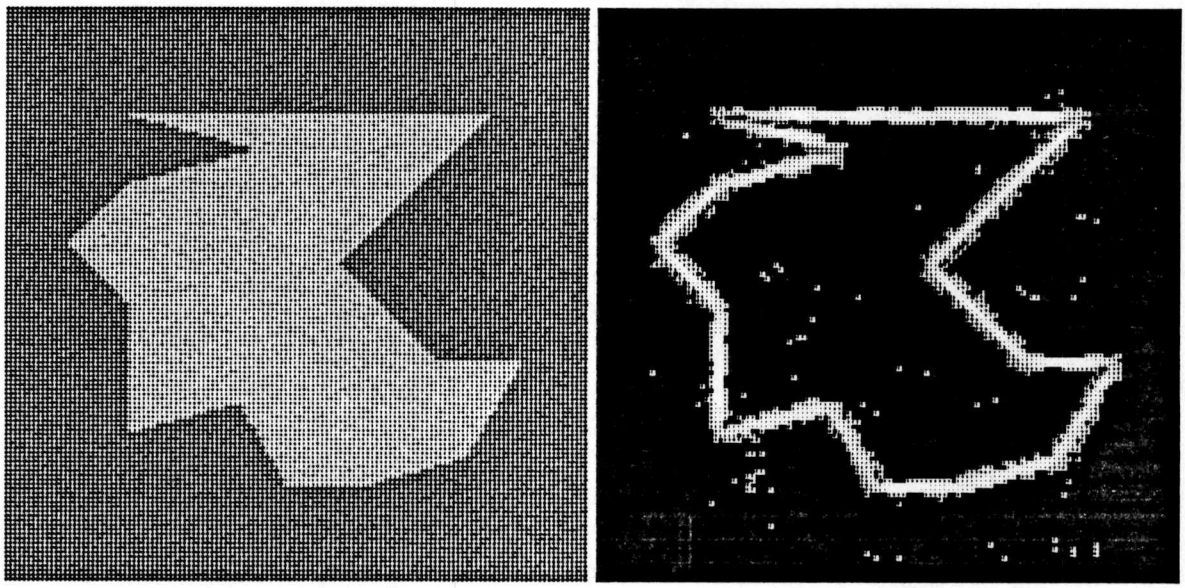
Figure 5.7: (a)(b) The noisy versions of Figures 5.1 and 5.2, respectively. The injected noise is additive Gaussian noise with zero mean and standard deviation of 10 percent of original intensity range ($\sigma_{noise} = 10$).



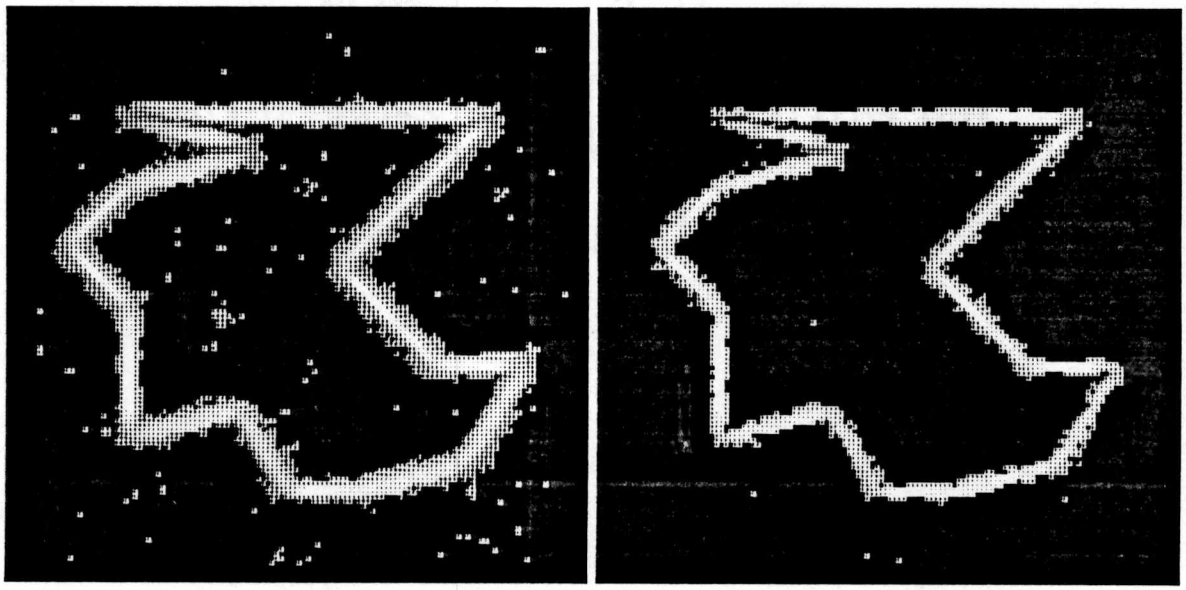
(a) (b)

(c) (d)

Figure 5.8: The detected edge pixels of Figure 5.7(a) using the proposed multi-template edge detection algorithm. (a) Figure 5.7(a). (b) Result of the 3 – 5 – 7 templates. (c) Result of the 5 – 7 – 9 templates. (d) Result of the 3 – 5 – 7 – 9 templates.



(a) (b)



(c) (d)

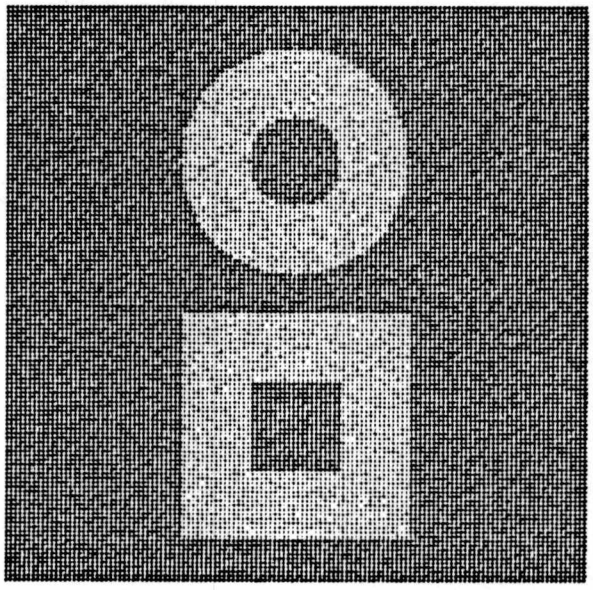
Figure 5.9: The detected edge pixels of Figure 5.7(b) using the proposed multi-template edge detection algorithm. (a) Figure 5.7(b). (b) Result of the 3 – 5 – 7 templates. (c) Result of the 5 – 7 – 9 templates. (d) Result of the 3 – 5 – 7 – 9 templates.

Table 5.3: The quantitative performance of the rule-based multi-template edge detection algorithm on the concentric circles and squares with additive Gaussian noise of zero mean and standard deviation $\sigma_{noise} = 10$, as shown in Figure 5.7(a).

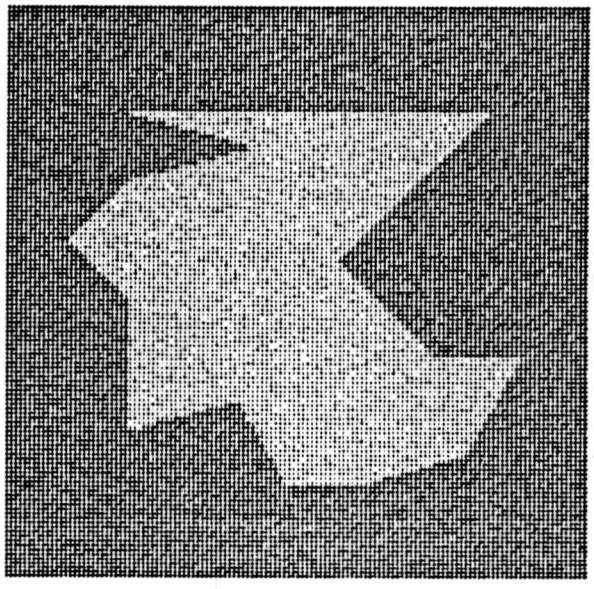
	Circles			**Squares**		
	$I_I = 568$			$I_I = 560$		$\sigma_{noise} = 10$
	3 – 5 – 7	5 – 7 – 9	3 – 5 – 7 – 9	3 – 5 – 7	5 – 7 – 9	3 – 5 – 7 – 9
R_a	0.996	1.000	1.000	1.000	1.000	0.998
R_b	0.569	1.585	0.533	0.713	1.691	0.664
R_c	0.571	1.585	0.533	0.713	1.691	0.665
R_d	1.565	2.585	1.533	1.713	2.691	1.663
R_1	0.946	0.855	0.952	0.946	0.865	0.957
R_2	0.956	3.546	0.719	0.933	2.533	0.455
R_3	0.470	1.105	0.425	0.498	1.027	0.416
I_A	889	1468	871	959	1507	931
N_a	566	568	568	560	560	559

Table 5.4: The quantitative performance of the rule-based multi-template edge detection algorithm on the corners with additive Gaussian noise of zero mean and standard deviation $\sigma_{noise} = 10$, as shown in Figure 5.7(b).

	Corners		$I_I = 982$ $\sigma_{noise} = 10$
	3 – 5 – 7	5 – 7 – 9	3 – 5 – 7 – 9
R_a	0.991	1.000	0.988
R_b	0.655	1.643	0.539
R_c	0.661	1.643	0.545
R_d	1.646	2.643	1.526
R_1	0.915	0.833	0.951
R_2	7.543	11.111	1.190
R_3	0.913	1.522	0.450
I_A	1616	2595	1499
N_a	973	982	970

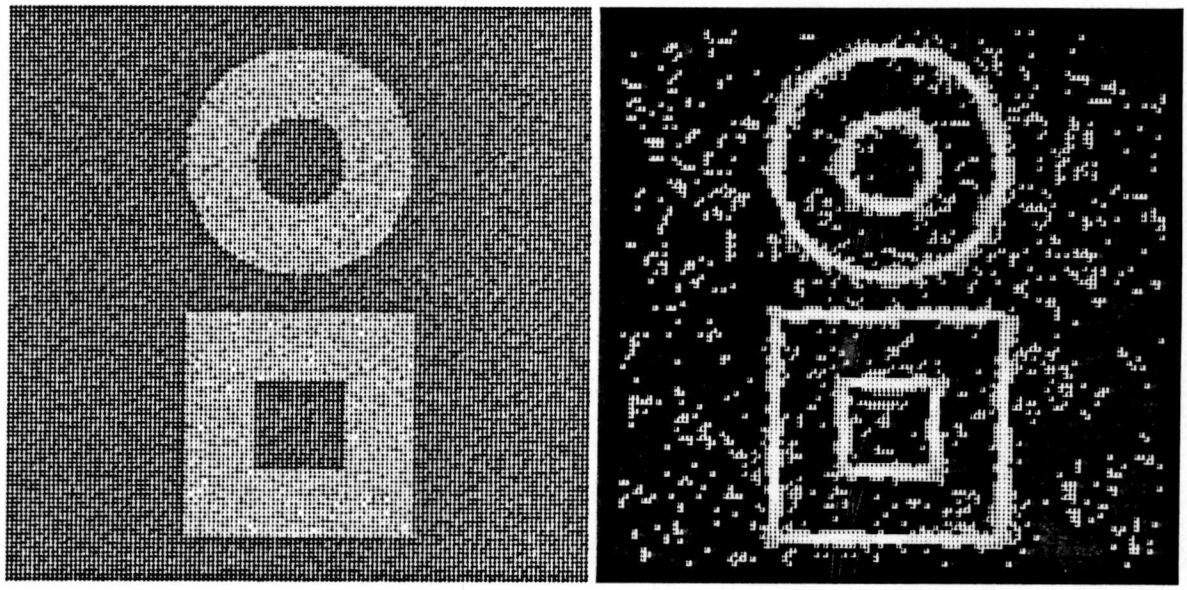


(a)

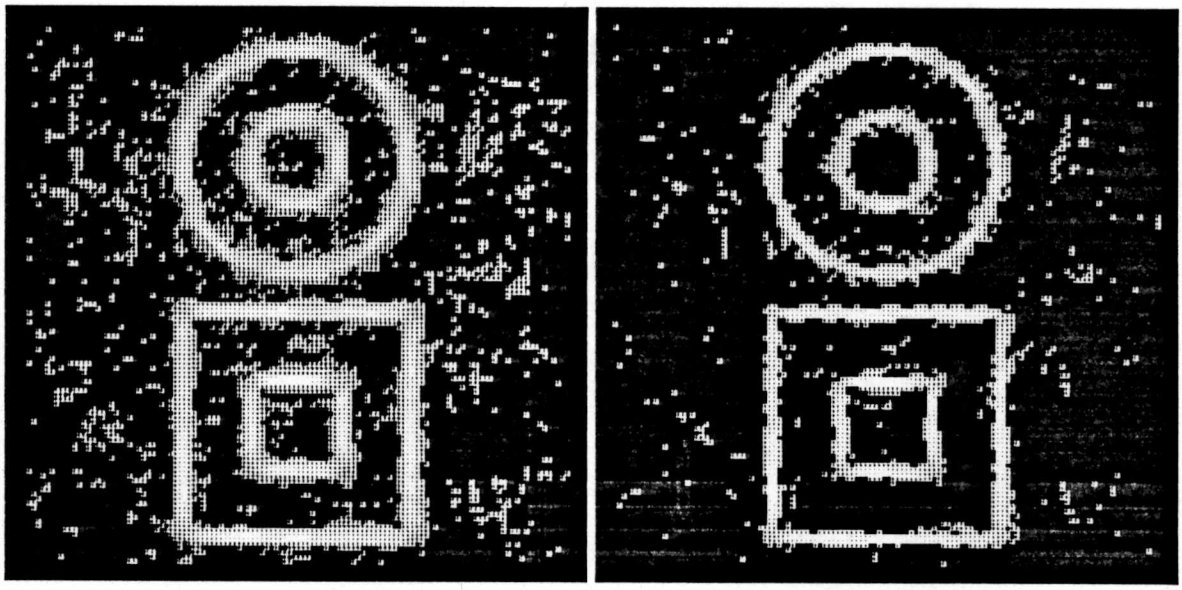


(b)

Figure 5.10: (a)(b) The noisy versions of Figures 5.1 and 5.2, respectively. The injected noise is additive Gaussian noise with zero mean and standard deviation of 20 percent of original intensity range ($\sigma_{noise} = 20$).



(a) (b)



(c) (d)

Figure 5.11: The detected edge pixels of Figure 5.10(a) using the proposed multi-template edge detection algorithm. (a) Figure 5.10(a). (b) Result of the 3 – 5 – 7 templates. (c) Result of the 5 – 7 – 9 templates. (d) Result of the 3 – 5 – 7 – 9 templates.

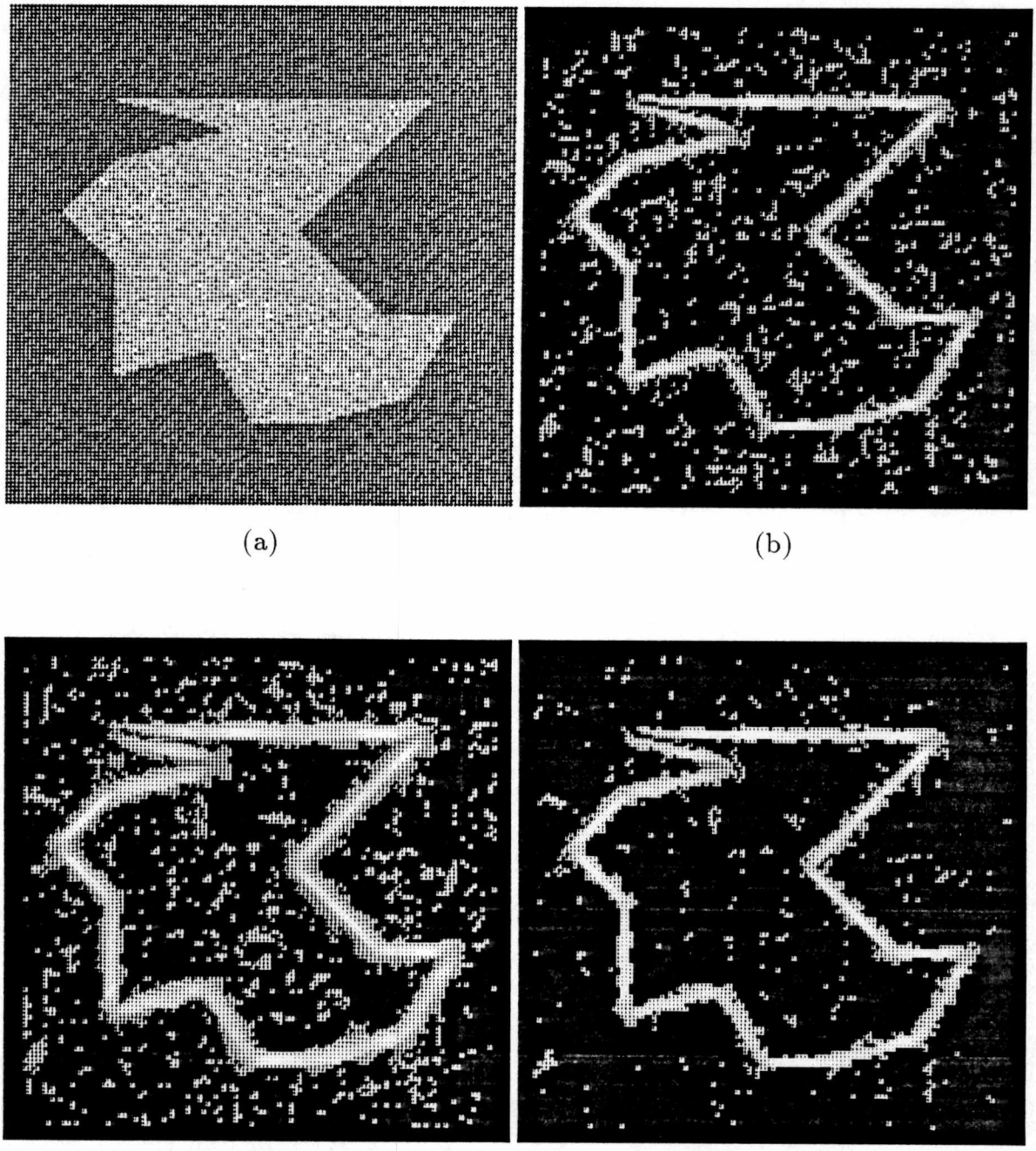


Figure 5.12: The detected edge pixels of Figure 5.10(b) using the proposed multi-template edge detection algorithm. (a) Figure 5.10(b). (b) Result of the 3 – 5 – 7 templates. (c) Result of the 5 – 7 – 9 templates. (d) Result of the 3 – 5 – 7 – 9 templates.

Table 5.5: The quantitative performance of the rule-based multi-template edge detection algorithm on the concentric circles and squares with additive Gaussian noise of zero mean and standard deviation $\sigma_{noise} = 20$, as shown in Figure 5.10(a).

	Circles			**Squares**		
	$I_I = 568$			$I_I = 560$		$\sigma_{noise} = 20$
	3 – 5 – 7	5 – 7 – 9	3 – 5 – 7 – 9	3 – 5 – 7	5 – 7 – 9	3 – 5 – 7 – 9
R_a	0.993	1.000	0.995	0.998	1.000	0.996
R_b	1.114	2.016	0.717	1.221	2.134	0.816
R_c	1.122	2.016	0.720	1.224	2.134	0.819
R_d	2.107	3.016	1.711	2.220	3.134	1.813
R_1	0.788	0.768	0.893	0.802	0.785	0.902
R_2	12.201	9.751	4.512	7.084	6.476	2.690
R_3	1.802	1.818	0.889	1.513	1.613	0.796
I_A	1197	1713	972	1243	1755	1015
N_a	564	568	565	559	560	558

Table 5.6: The quantitative performance of the rule-based multi-template edge detection algorithm on the corners with additive Gaussian noise of zero mean and standard deviation $\sigma_{noise} = 20$, as shown in Figure 5.10(b).

	Corners		$I_I = 982$ $\sigma_{noise} = 20$
	3 – 5 – 7	5 – 7 – 9	3 – 5 – 7 – 9
R_a	0.985	0.992	0.984
R_b	1.799	2.814	0.895
R_c	1.827	2.837	0.910
R_d	2.784	3.805	1.879
R_1	0.603	0.611	0.807
R_2	71.073	61.761	29.747
R_3	5.114	4.700	2.354
I_A	2734	3737	1845
N_a	967	974	966

5.1.3 Comparison

In Table 5.1, the measurement indicates that in the noiseless case, all three combinations, the 3 – 5 – 7, the 5 – 7 – 9, and the 3 – 5 – 7 – 9 templates, can detect all of the edge pixels of the circles (568 pixels) and the squares (560 pixels). Both the 3 – 5 – 7 and the 3 – 5 – 7 – 9 combinations have the best possible values of $R_a = R_d = R_1 = 1$ and $R_b = R_c = R_2 = R_3 = 0$. Because of the large template sizes, the number of detected edge pixels for the 5 – 7 – 9 combination is approximately twice the number of the real edge pixels. For the measurement at the corners in Table 5.2, both the 3 – 5 – 7 and the 3 – 5 – 7 – 9 combinations have the same performance — two missing edge pixels. These two missing edge pixels, one pixel at the 15° corner and one pixel at the 30° corner, do not break the connectivity of the contour but generate the rounding-off effect. With large template sizes, the 5 – 7 – 9 combination can detect all of the edge pixels on these corners, however a high noise-to-signal ratio ($R_c = 0.956$) results.

In the case with the presence of additive Gaussian noise with standard deviation $\sigma_{noise} = 10$, Table 5.3 shows that the proposed combinations can still detect almost all of the edge pixels on the edges of the circles and the squares and obtain high ideal edge pixel detecting ratios R_a (0.996, 1, and 1 for 3 – 5 – 7, 5 – 7 – 9 and 3 – 5 – 7 – 9 combinations in circles). For the corners, the missing edge pixels of the 3 – 5 – 7, the 5 – 7 – 9, and the 3 – 5 – 7 – 9 combinations are 9, 0, and 10 pixels, respectively, as shown in Table 5.4. In this noisy case, the 3

– 5 – 7 – 9 combination still remains the lowest noise-to-signal ratio $R_c = 0.533$ within these three combinations. The 5 – 7 – 9 combination not only has the highest ideal edge pixel detecting ratio R_a but also has the largest noise-to-signal ratio R_c of each of the three kinds of edges (circles, squares, and corners). In the noisy case, the measure that has the largest increment is the average squared deviation R_2. In each of the three kinds of edges, both the 3 – 5 – 7 and the 5 – 7 – 9 combinations have a large rise in R_2. Although the 3 – 5 – 7 – 9 combination has a slightly smaller ideal edge pixel detecting ratio, R_a, than the 3 – 5 – 7 and the 5 – 7 – 9 combinations, its measure R_2 (1.190 for corners) has a much better performance than the measures in the other two combinations (7.543 and 11.111 for corners). This indicates that the four-template algorithm is less sensitive to noise than the three-template algorithm.

With the presence of a large amount of noise, the multi-template combinations can still detect most of the edge pixels of the synthetic images, however they have lower measures of R_1 than the noiseless and the medium noise cases. Because of noise, all three combinations obtain a large amount of noisy pixels in their detected edge pixels. This effect causes a large noise-to-signal ratio R_c and a large average squared deviation R_2 to be assigned. Within these three combinations, the four-template combination still has the most stable performance.

In Peli and Malah's report, it was pointed out that even for an ideal edge most of the edge detection algorithms had problems in finding all of the edge

pixels in a given image, especially for a circle. Observing the measures R_a in both the noiseless and the noisy cases, it appears that all three proposed combinations have measures of R_1 much greater than the failing threshold 0.5 in each of the three kinds of edges. Except for the measures of the average squared deviation R_2, each of the measures in the three combinations performed well. Because most of the detected noise pixels, which caused large measures of R_2 to be assigned, are weak and discontinued, they can be easily removed by an edge thinning algorithm. Thus, the proposed rule-based multi-template edge detection algorithm has excellent performance based on the quantitative measurement presented by Peli and Malah.

Based on the observed quantitative performance, we can conclude the following: both the 3 - 5 - 7 and the 3 - 5 - 7 - 9 combinations are excellent for finding edges in noiseless images. Because the four-template algorithm is less sensitive to noise than the three-template algorithm, the 3 - 5 - 7 - 9 combination is better than the 3 - 5 - 7 combination. When the amount of noise becomes large, the superiority of the 3 - 5 - 7 - 9 combination over the 3 - 5 - 7 combination becomes obvious. Thus, the four-template algorithm is more effective than the three-template algorithm for edge detection in noisy images. Of the three combinations, the 5 - 7 - 9 combination has the least merit in either the noiseless or the noisy case.

To observe the performance of the algorithm in very noisy images, the stan-

dard deviation of the additive Gaussian noise in the two binary synthetic images is increased to σ_{noise} = 40, 60, and 80, as shown in Figures 5.13(a), 5.13(b), 5.14(a), 5.15(a), 5.15(b), and 5.16(a), respectively. To get the meaning, only the premium template combination, the four-template combination, is implemented. The resulting edge images are shown in Figures 5.13(c), 5.13(d), 5.14(b), 5.15(c), 5.15(d), and 5.16(b). The quantitative performance of the edge detection algorithm are listed in Tables 5.7 and 5.8. From the information given in Tables 5.7 and 5.8, it appears that the algorithm continues to perform well. With the presence of huge amount of noise, the four-template combination can still detect more than 78 percent of the ideal edge pixels (R_a = 0.785 on corners with σ_{noise} = 80) and keep the noise-to-signal ratios R_c smaller than 2.8 (R_c = 2.722 on corners with σ_{noise} = 80).

In order to compare performance of the proposed edge detection algorithm with edge detectors that employ only a single size template, we consider performance of edge detectors that use only 3 × 3 or 9 × 9 directional templates. The results obtained in this cases are summarized in Tables 5.9, 5.10, and 5.11. In general, these edge detectors perform very well when no noise is present (Table 5.9). However, when the noise increases the performance of these edge detectors deteriorates since most pixels are considered to be edges (Tables 5.10 and 5.11). Comparison between 3 - 5 - 7 combination and the Sobel edge detector was carried out in [82].

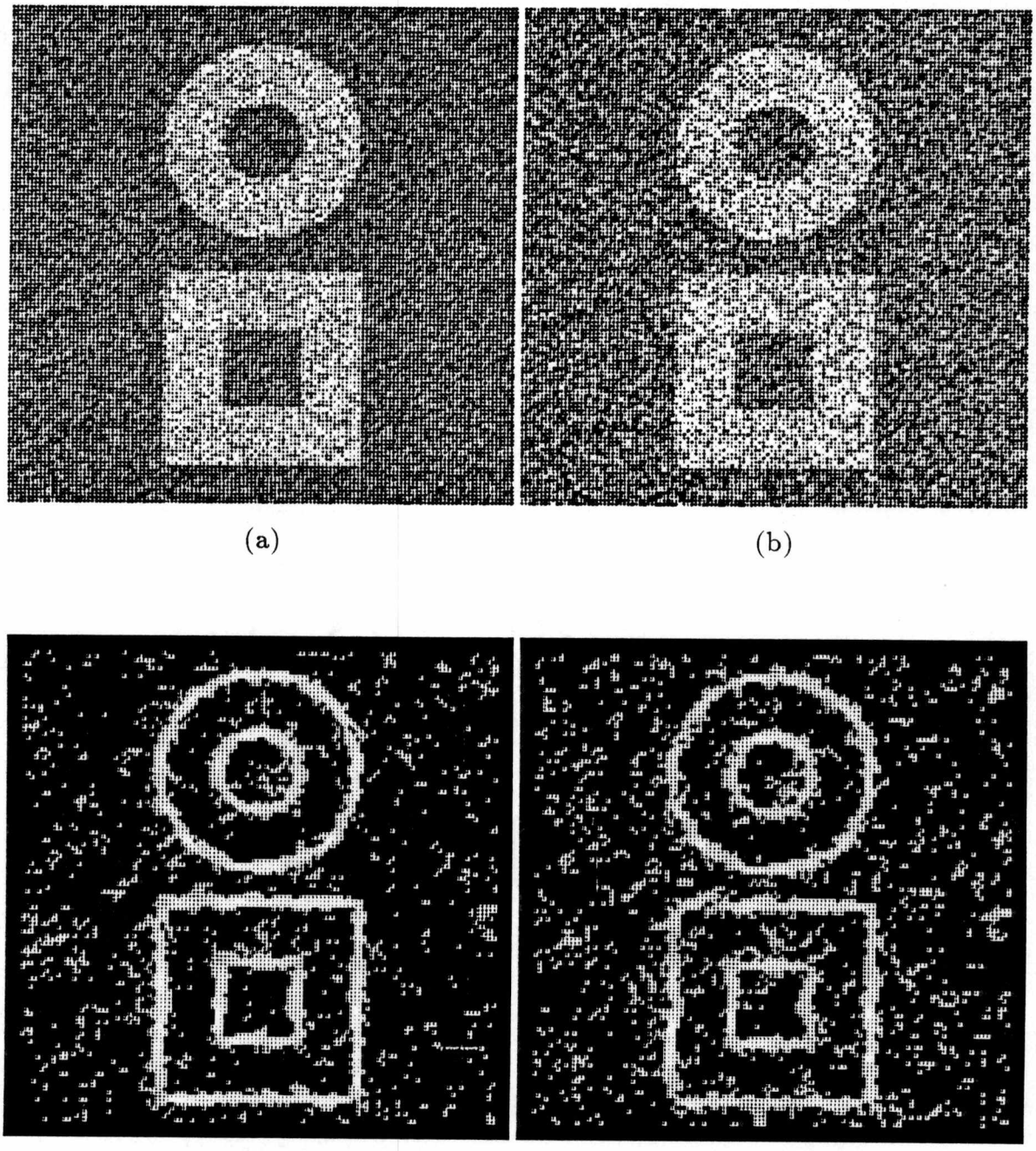
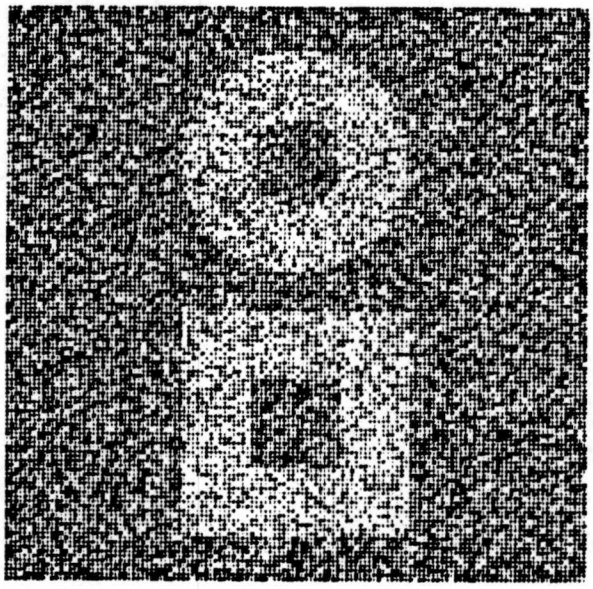
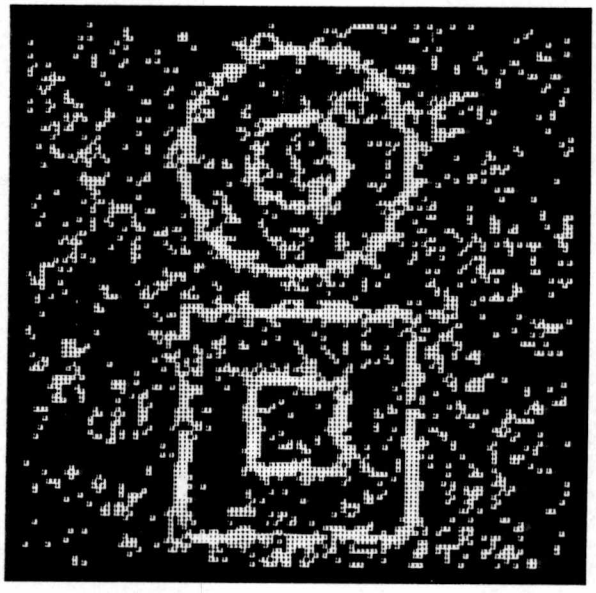


Figure 5.13: (a)(b) The noisy versions of Figure 5.1. The injected noise is additive Gaussian noise with zero mean and standard deviation of 40 and 60 percent of original intensity range ($\sigma_{noise} = 40$ and 60). (c)(d) The results of (a) and (b), respectively, using the 3 – 5 – 7 – 9 templates.



(a)



(b)

Figure 5.14: (a) The noisy version of Figure 5.1. The injected noise is additive Gaussian noise with zero mean and standard deviation of 80 percent of original intensity range ($\sigma_{noise} = 80$). (b) The result of (a) using the 3 – 5 – 7 – 9 templates.

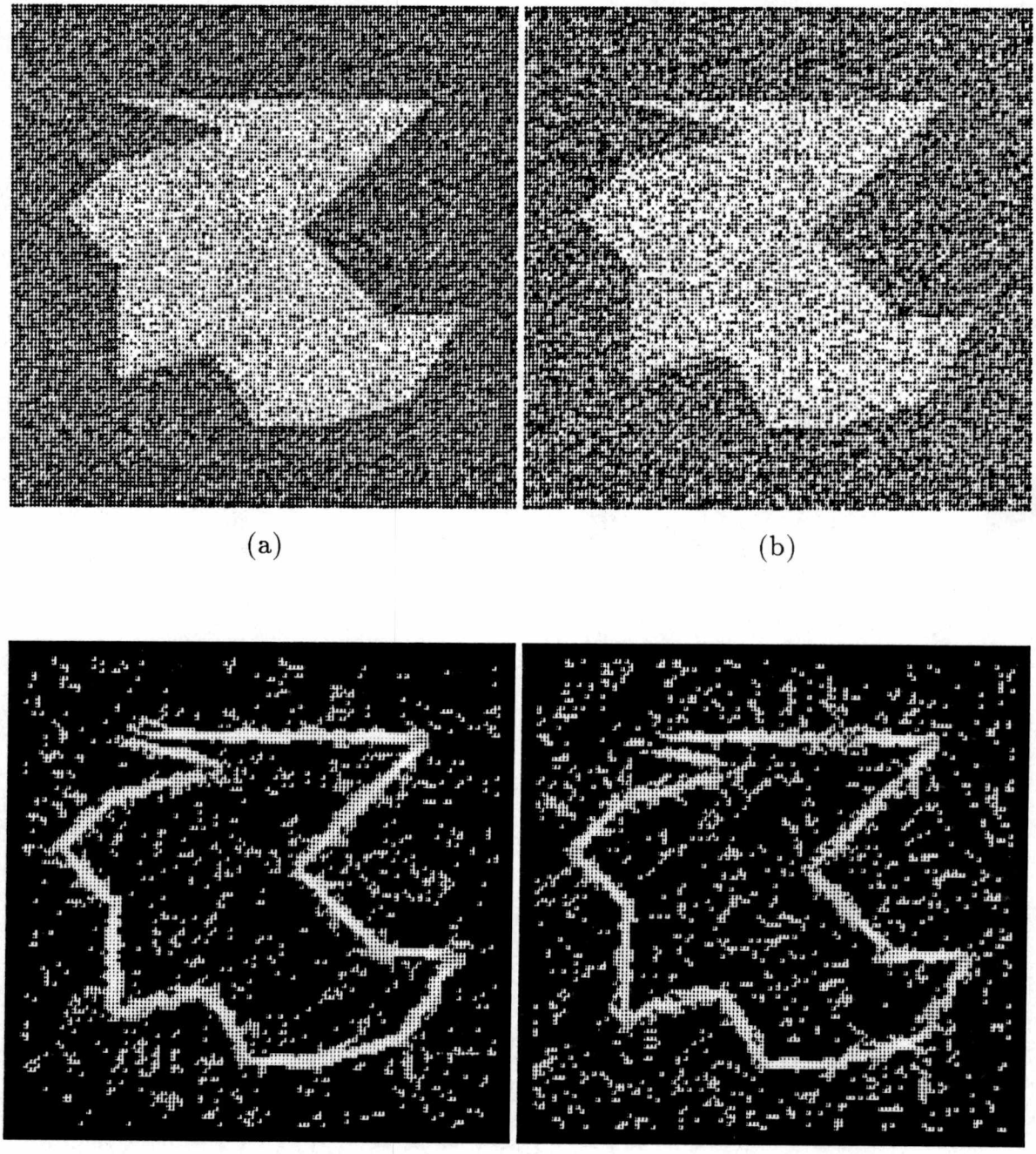
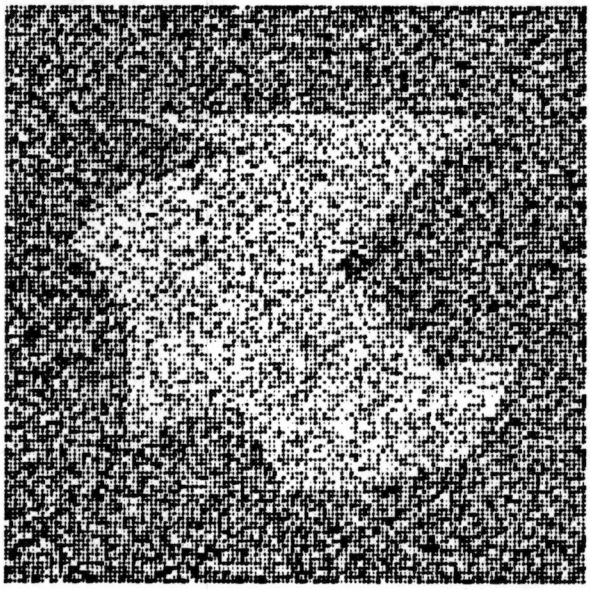
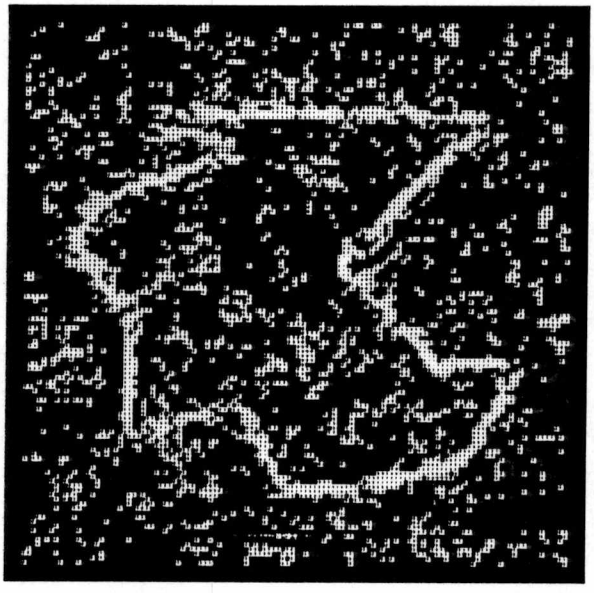


Figure 5.15: (a)(b) The noisy versions of Figure 5.2. The injected noise is additive Gaussian noise with zero mean and standard deviation of 40 and 60 percent of original intensity range ($\sigma_{noise} = 40$ and 60). (c)(d) The results of (a) and (b), respectively, using the 3 – 5 – 7 – 9 templates.



(a)



(b)

Figure 5.16: (a) The noisy version of Figure 5.2. The injected noise is additive Gaussian noise with zero mean and standard deviation of 80 percent of original intensity range ($\sigma_{noise} = 80$). (b) The result of (a) using the 3 – 5 – 7 – 9 templates.

Table 5.7: The quantitative performance of the 3 – 5 – 7 – 9 rule-based multi-template edge detection algorithm on the circles and squares with additive Gaussian noise of zero mean and standard deviation $\sigma_{noise} = 40$, 60, and 80 as shown in Figures 5.13(a), 5.13(b), and 5.14(b), respectively.

	Circles			**Squares**		
	$I_I = 568$			$I_I = 560$		
	$\sigma_{noise} = 40$	$\sigma_{noise} = 60$	$\sigma_{noise} = 80$	$\sigma_{noise} = 40$	$\sigma_{noise} = 60$	$\sigma_{noise} = 80$
R_a	0.954	0.924	0.840	0.995	0.964	0.886
R_b	1.021	1.121	1.174	1.188	1.446	1.393
R_c	1.070	1.213	1.398	1.194	1.500	1.573
R_d	1.975	2.046	2.014	2.182	2.411	2.279
R_1	0.769	0.735	0.704	0.815	0.760	0.733
R_2	16.848	22.446	21.697	5.998	8.784	10.652
R_3	2.087	2.531	2.656	1.395	1.811	2.038
I_A	1122	1162	1144	1222	1350	1276
N_a	542	525	477	557	540	496

Table 5.8: The quantitative performance of the 3 - 5 - 7 - 9 rule-based multi-template edge detection algorithm on the corners with additive Gaussian noise of zero mean and standard deviation σ_{noise} = 40, 60, and 80 as shown in Figures 5.15(a), 5.15(b), and 5.16(a), respectively.

Corners			$I_I = 982$
	$\sigma_{noise} = 40$	$\sigma_{noise} = 60$	$\sigma_{noise} = 80$
R_a	0.953	0.889	0.785
R_b	1.725	1.871	2.137
R_c	1.810	2.104	2.722
R_d	2.678	2.760	2.923
R_1	0.609	0.547	0.477
R_2	65.495	90.497	103.347
R_3	4.854	6.097	7.024
I_A	2630	2710	2870
N_a	936	873	771

Table 5.9: The quantitative performance of the edge detection algorithm using one template size on the concentric circles, squares, and corners, as shown in Figures 5.1 and 5.2.

	Circles		**Squares**		**Corners**	
	$I_I = 568$		$I_I = 560$		$I_I = 982$	
	$\sigma_{noise} = 0$					
	3×3	9×9	3×3	9×9	3×3	9×9
R_a	1.000	1.000	1.000	1.000	0.998	1.000
R_b	0.000	3.060	0.000	3.000	0.000	2.790
R_c	0.000	3.060	0.000	3.000	0.000	2.790
R_d	1.000	4.060	1.000	4.000	0.998	3.790
R_1	1.000	0.720	1.000	0.770	0.998	0.746
R_2	0.000	6.662	0.000	3.593	0.000	4.533
R_3	0.000	1.893	0.000	1.516	0.000	1.662
I_A	568	2306	560	2240	980	3722
N_a	568	568	560	560	980	982

Table 5.10: The quantitative performance of the edge detection algorithm using one template size on the concentric circles, squares, and corners with additive Gaussian noise of zero mean and standard deviation $\sigma_{noise} = 20$, as shown in Figure 5.10.

	Circles		**Squares**		**Corners**	
	$I_I = 568$		$I_I = 560$		$I_I = 982$	
	$\sigma_{noise} = 20$					
	3×3	9×9	3×3	9×9	3×3	9×9
R_a	1.000	1.000	1.000	1.000	0.999	1.000
R_b	5.741	6.405	5.755	6.420	11.944	13.079
R_c	5.741	6.405	5.755	6.420	11.956	13.079
R_d	6.741	7.405	6.755	7.420	12.943	14.079
R_1	0.471	0.470	0.523	0.521	0.269	0.269
R_2	40.017	39.306	19.300	19.315	143.416	144.061
R_3	4.734	4.701	3.529	3.544	9.700	9.720
I_A	3829	4206	3783	4155	12710	13826
N_a	568	568	560	560	981	982

Table 5.11: The quantitative performance of the edge detection algorithm using one template size on the concentric circles, squares, and corners with additive Gaussian noise of zero mean and standard deviation $\sigma_{noise} = 80$, as shown in Figures 5.14(a) and 5.16(a).

	Circles		**Squares**		**Corners**	
	$I_I = 568$		$I_I = 560$		$I_I = 982$	
	$\sigma_{noise} = 80$					
	3 × 3	9 × 9	3 × 3	9 × 9	3 × 3	9 × 9
R_a	0.995	1.000	1.000	1.000	0.997	1.000
R_b	6.602	6.623	6.584	6.602	13.624	13.658
R_c	6.637	6.623	6.584	6.602	13.666	13.658
R_d	7.597	7.623	7.584	7.602	14.621	14.658
R_1	0.461	0.461	0.514	0.514	0.262	0.262
R_2	40.736	40.720	19.742	19.735	145.868	145.890
R_3	4.812	4.810	3.600	3.599	9.826	9.828
I_A	4315	4330	4247	4257	14358	14394
N_a	565	568	560	560	979	982

5.2 Performance on Infrared Images

In the previous section, the quantitative performance of the proposed rule-based multi-template edge detection algorithm on two binary synthetic images was affirmed. In this section, the proposed rule-based multi-template edge detection algorithm is applied to extract the edges and one-pixel-wide contours of the noisy and low-contrast infrared images.

Typical examples of infrared images are shown in Figures 5.17, 5.25, and 5.33. The size of these images is 256×256 pixels, and their intensities vary from 0 to 255. Each of these infrared images has an ideal segmentation image, as shown in Figures 5.18, 5.26, and 5.34. The proposed algorithm, with threshold values $T_{edge} = 70$ and $T_{ec} = 0.8$ for the 3 - 5 - 7 combination, $T_{edge} = 110$ and $T_{ec} = 0.8$ for the 5 - 7 - 9 combination, and $T_{edge} = 70$ and $T_{ec} = 0.5$ for the 3 - 5 - 7 - 9 combination, was tested on each of these three infrared images to find their edges. Before the processing, in order to improve the results, a 5×5 median filter was used to achieve image enhancement.

After all of the qualified edge pixels were detected, the thinning algorithm described in Section 4.4, with the threshold of segment length TSL = 20 pixels, was applied to extract the one-pixel-wide continuous contours on the ridge of the detected edges. Using the proposed algorithm, the edge and contour images of

these three infrared images and their synthetic segmentation images are shown in the Figures 5.19 to 5.40.

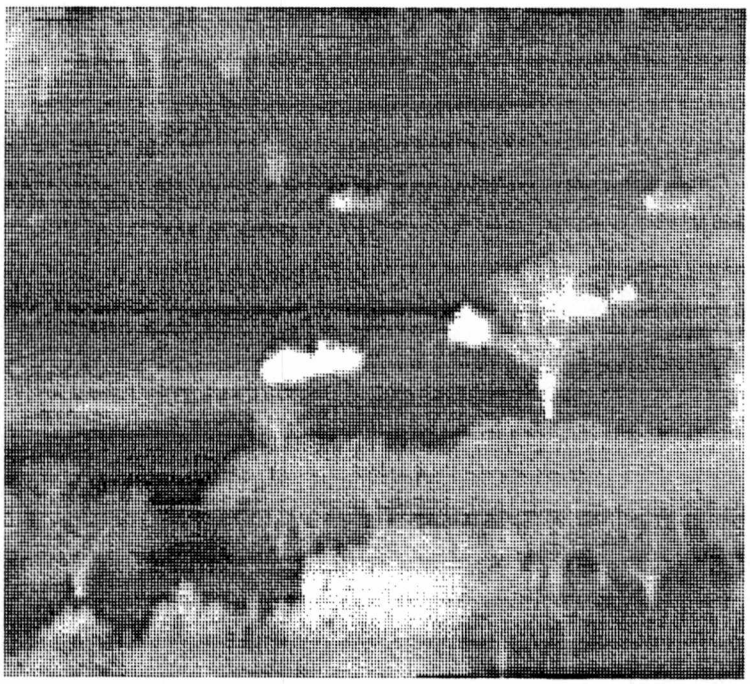


Figure 5.17: An infrared image of size 256 $\times$ 256 pixels and intensities from 0 to 255.

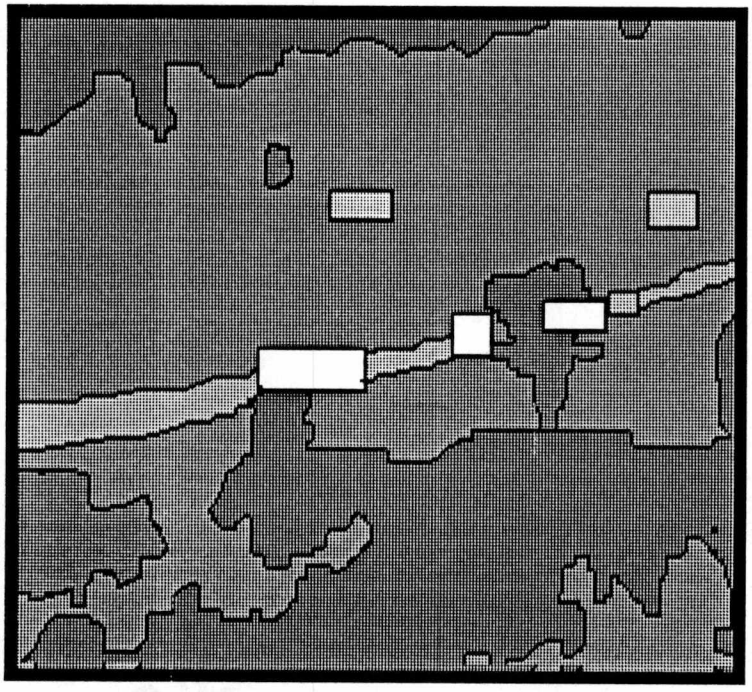


Figure 5.18: The ideal segmentation of image in Figure 5.17.

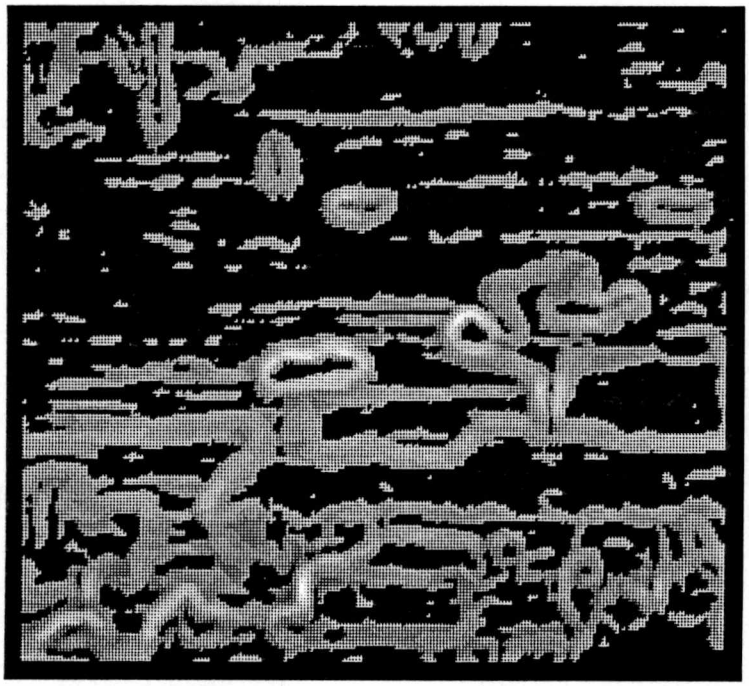
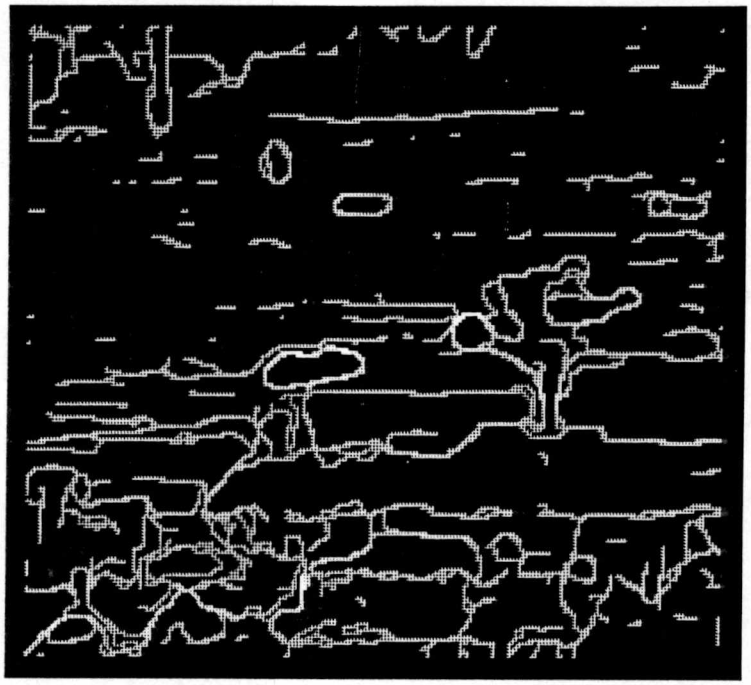
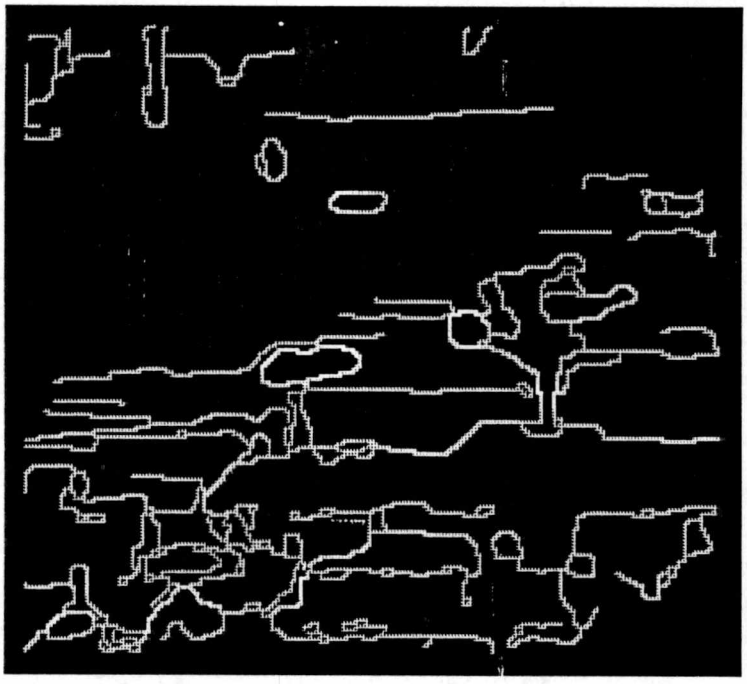


Figure 5.19: The edge image of Figure 5.17 using the $3-5-7$ algorithm.



(a)



(b)

Figure 5.20: (a) The one-pixel-wide edge of Figure 5.19. (b) The result of (a) after edge refinement procedure with threshold TSL = 20 pixels.

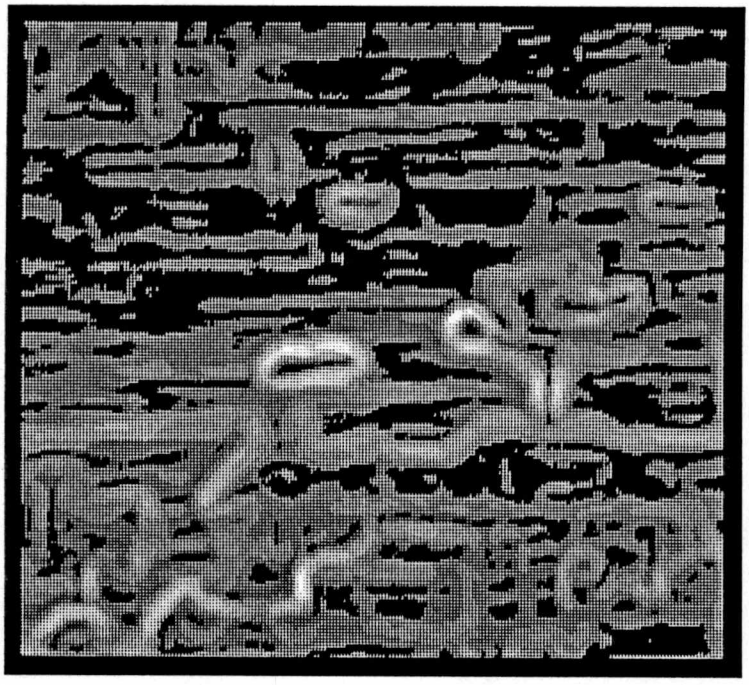
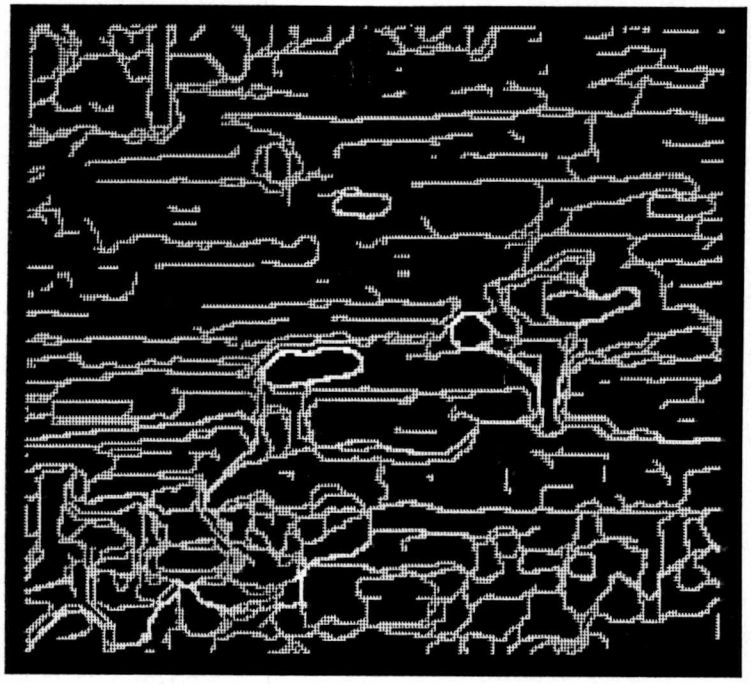
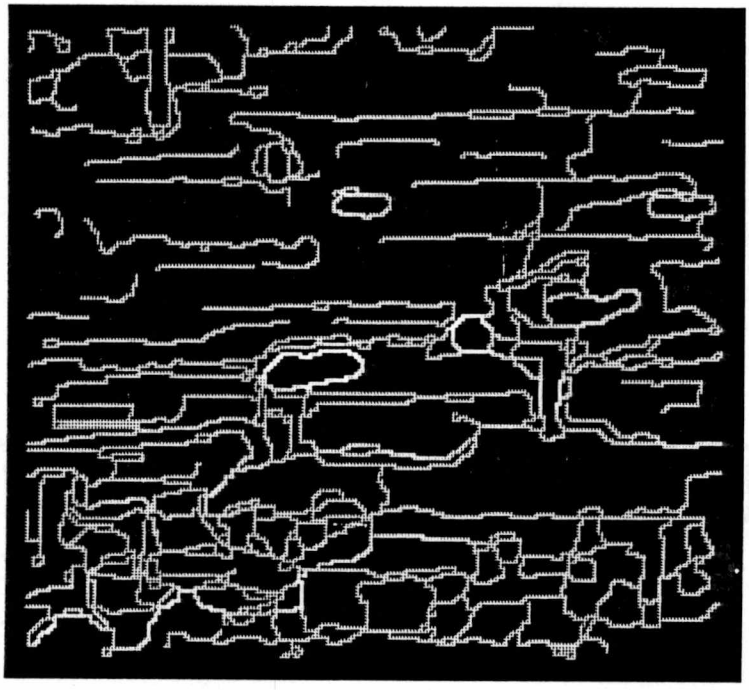


Figure 5.21: The edge image of Figure 5.17 using the 5 – 7 – 9 algorithm.



(a)

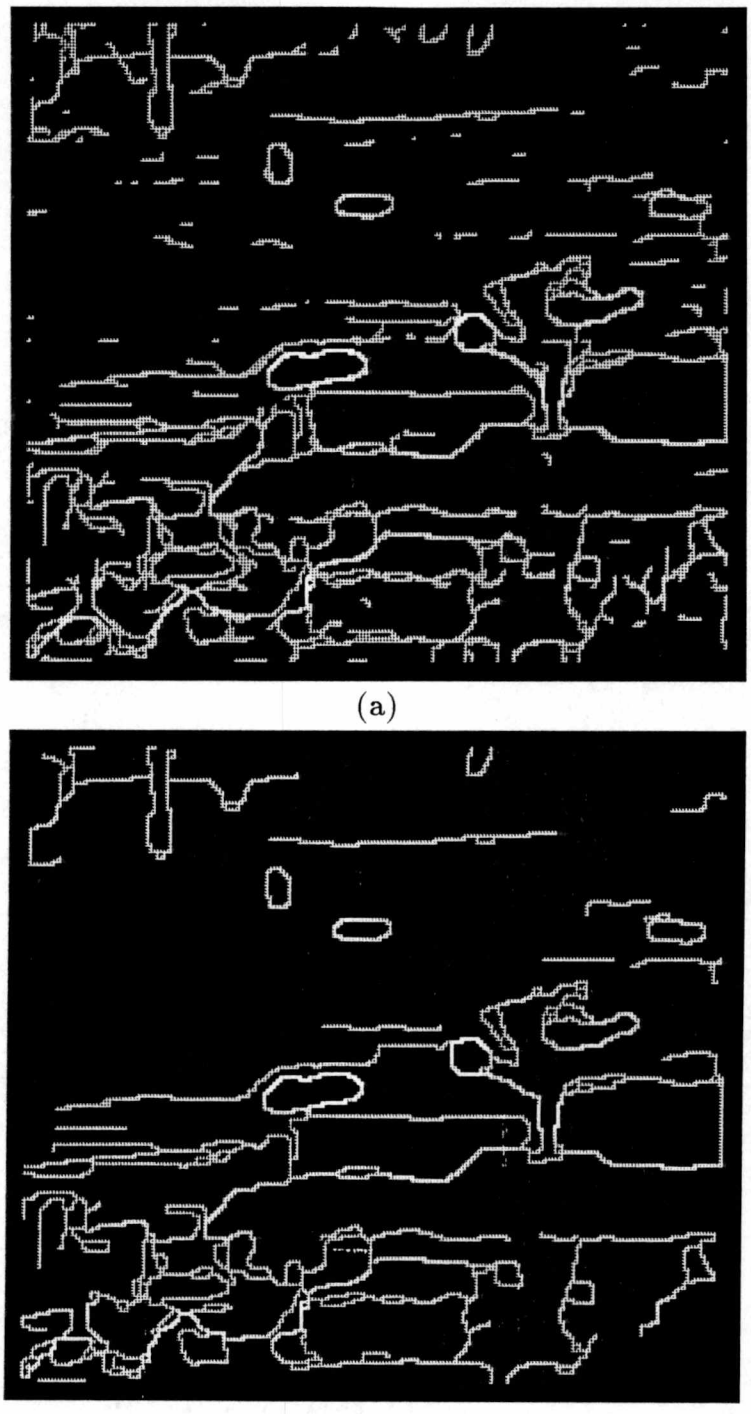


(b)

Figure 5.22: (a) The one-pixel-wide edge of Figure 5.21. (b) The result of (a) after edge refinement procedure with threshold TSL = 20 pixels.



Figure 5.23: The edge image of Figure 5.17 using the 3 – 5 – 7 – 9 algorithm.



(a)

(b)

Figure 5.24: (a) The one-pixel-wide edge of Figure 5.23. (b) The result of (a) after edge refinement procedure with threshold TSL = 20 pixels.

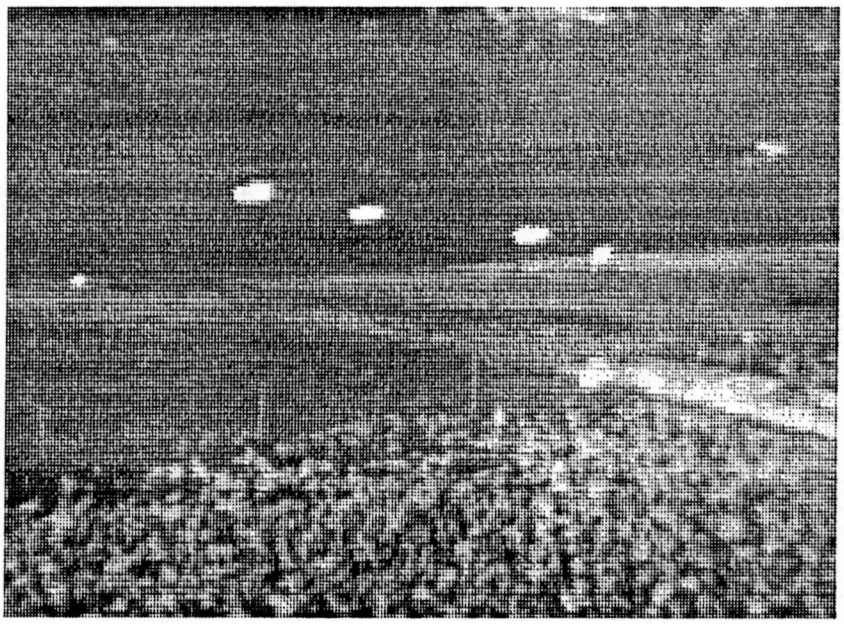


Figure 5.25: An infrared image of size 256 $\times$ 256 pixels and intensities from 0 to 255.

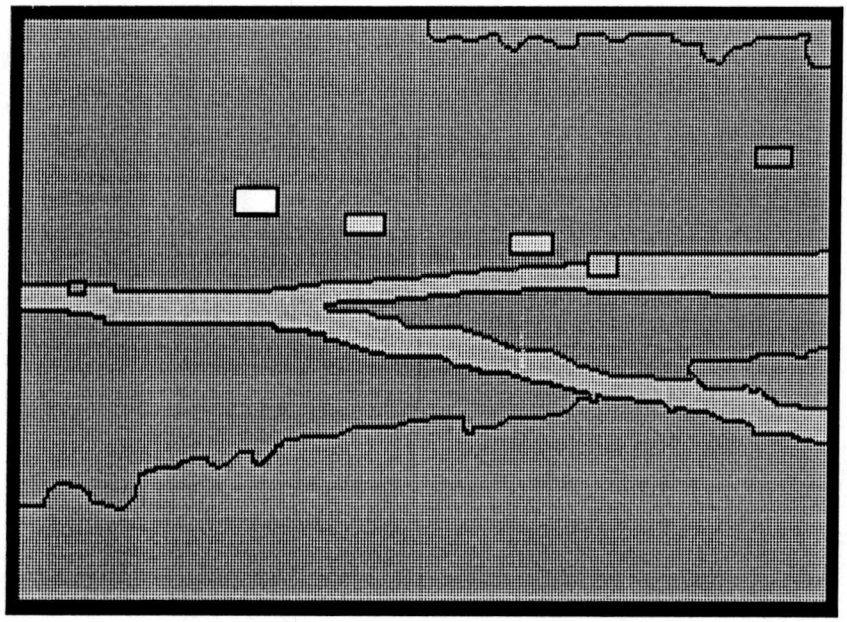


Figure 5.26: The ideal segmentation of image in Figure 5.25.

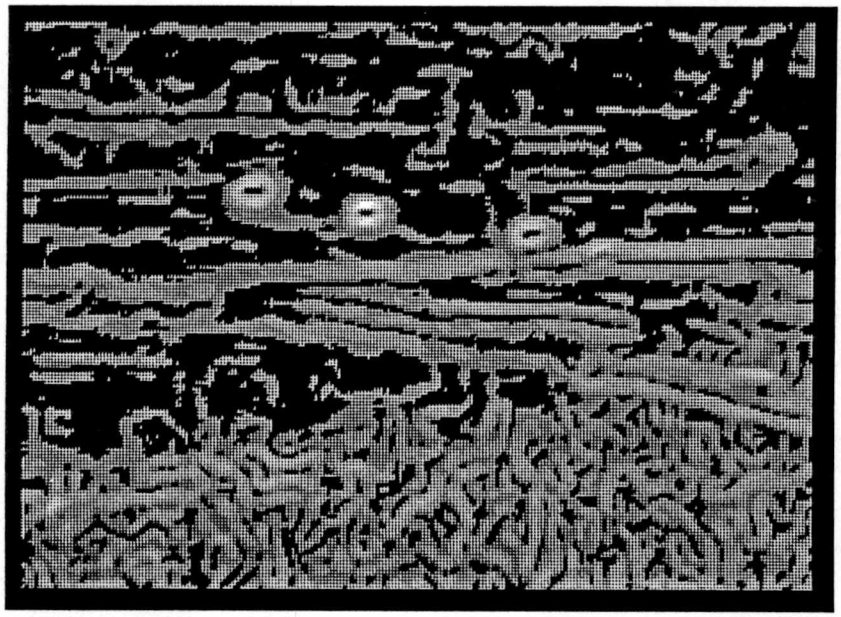
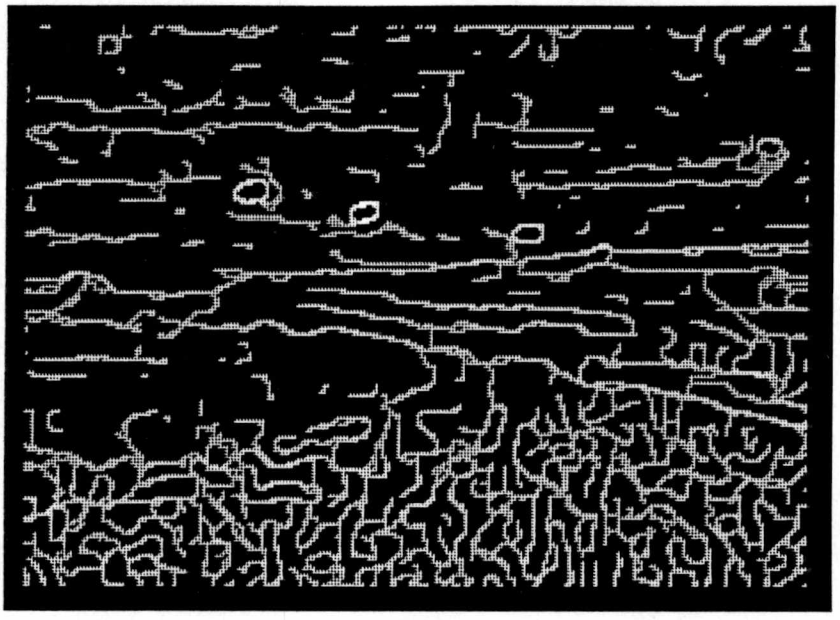
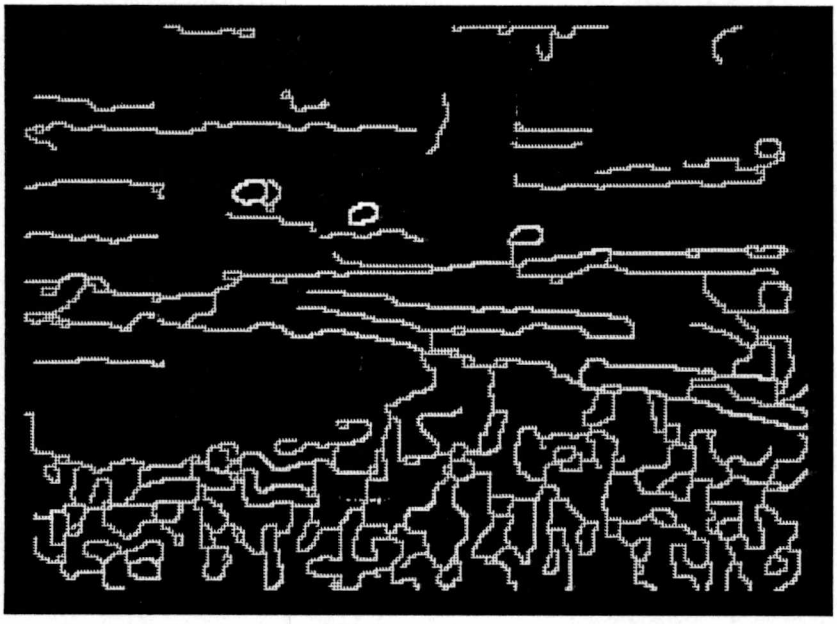


Figure 5.27: The edge image of Figure 5.25 using the 3 – 5 – 7 algorithm.



(a)



(b)

Figure 5.28: (a) The one-pixel-wide edge of Figure 5.27. (b) The result of (a) after edge refinement procedure with threshold TSL = 20 pixels.

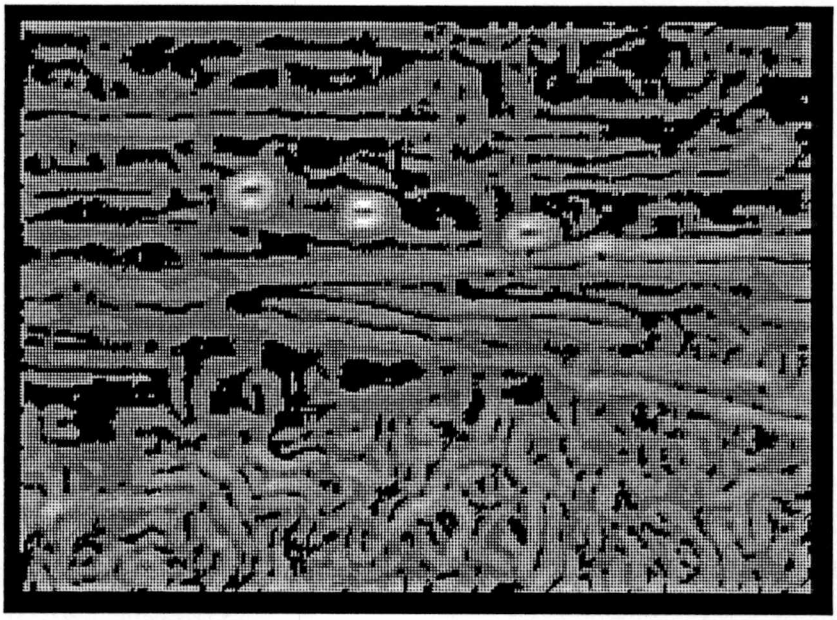
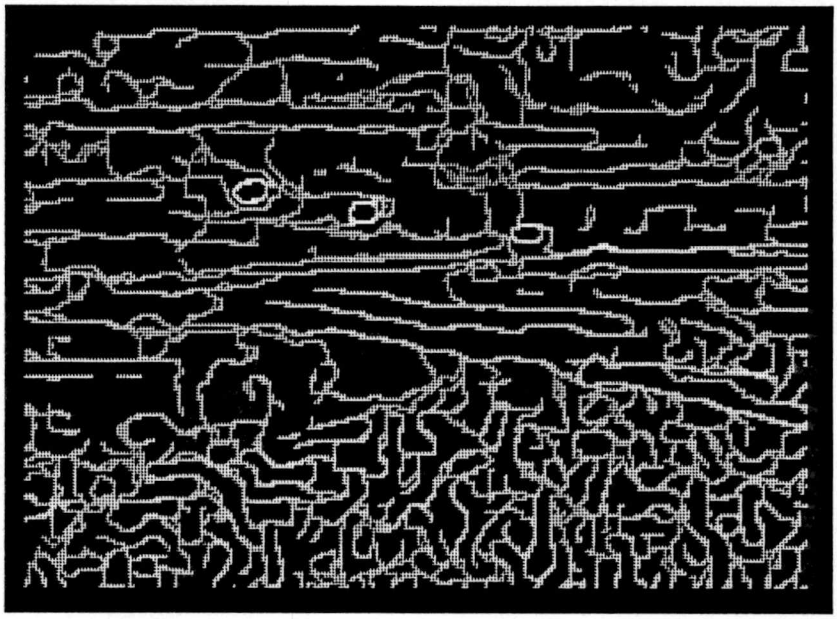
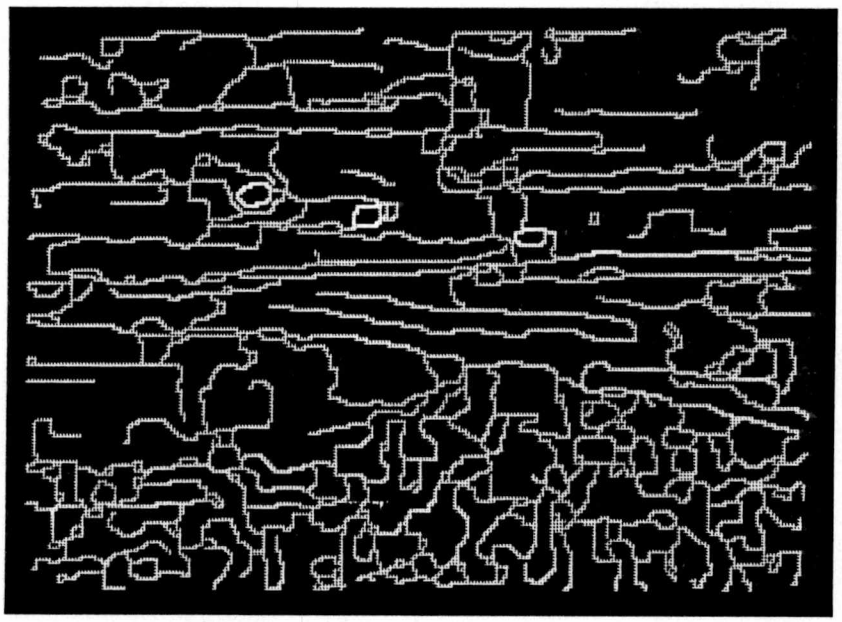


Figure 5.29: The edge image of Figure 5.25 using the 5 – 7 – 9 algorithm.



(a)



(b)

Figure 5.30: (a) The one-pixel-wide edge of Figure 5.29. (b) The result of (a) after edge refinement procedure with threshold TSL = 20 pixels.

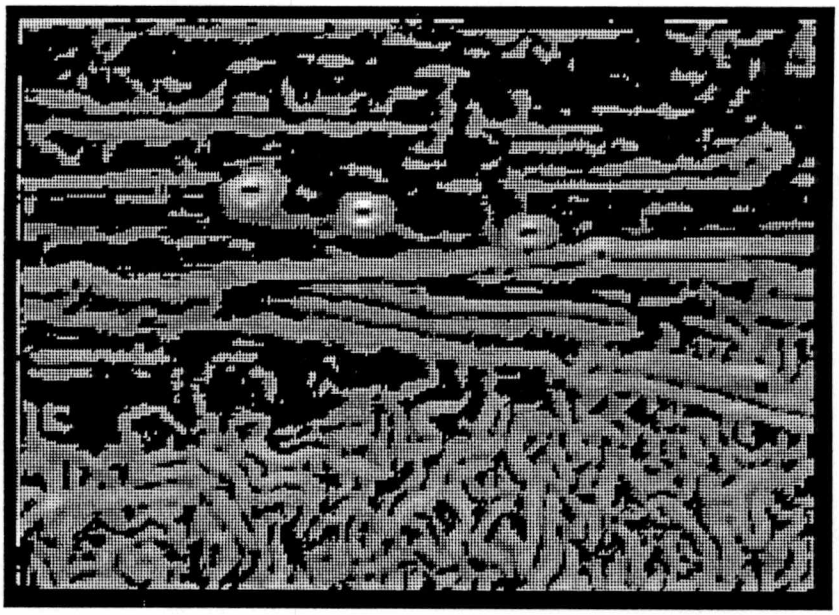
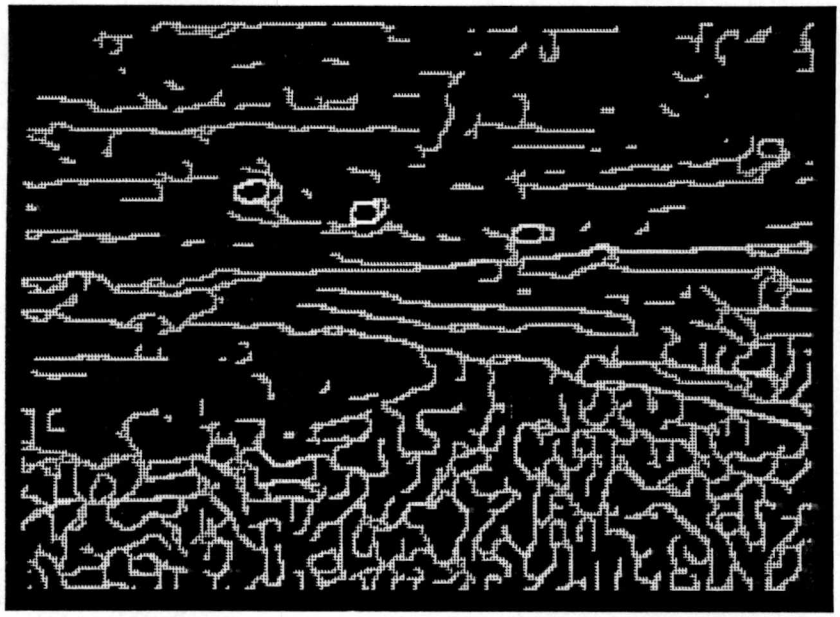
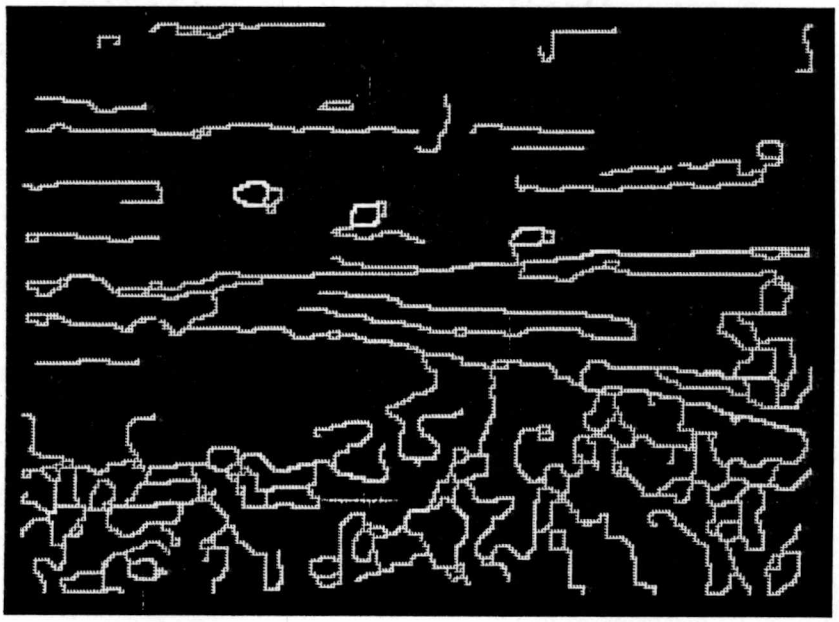


Figure 5.31: The edge image of Figure 5.25 using the 3 – 5 – 7 – 9 algorithm.



(a)



(b)

Figure 5.32: (a) The one-pixel-wide edge of Figure 5.31. (b) The result of (a) after edge refinement procedure with threshold TSL = 20 pixels.

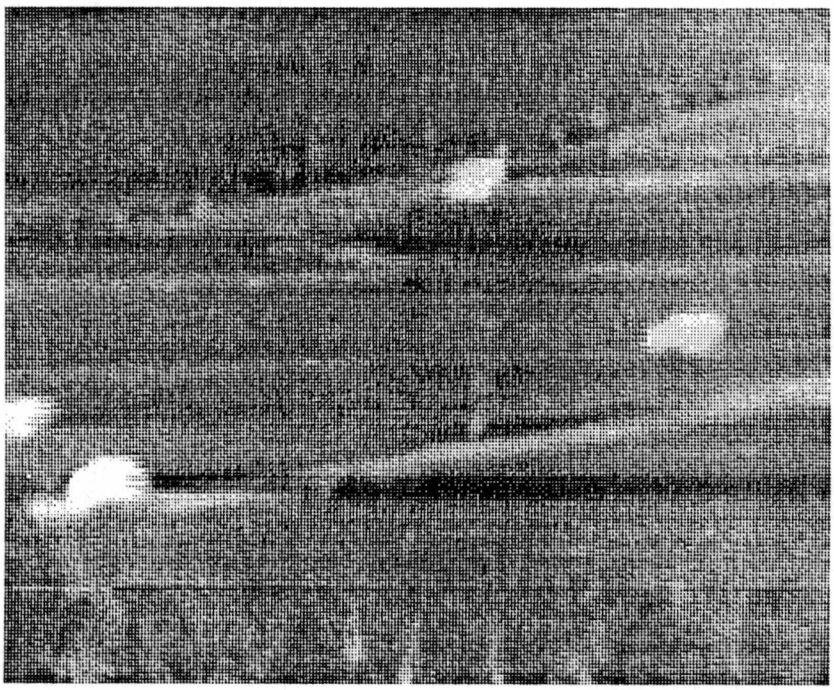


Figure 5.33: An infrared image of size 256 $\times$ 256 pixels and intensities from 0 to 255.

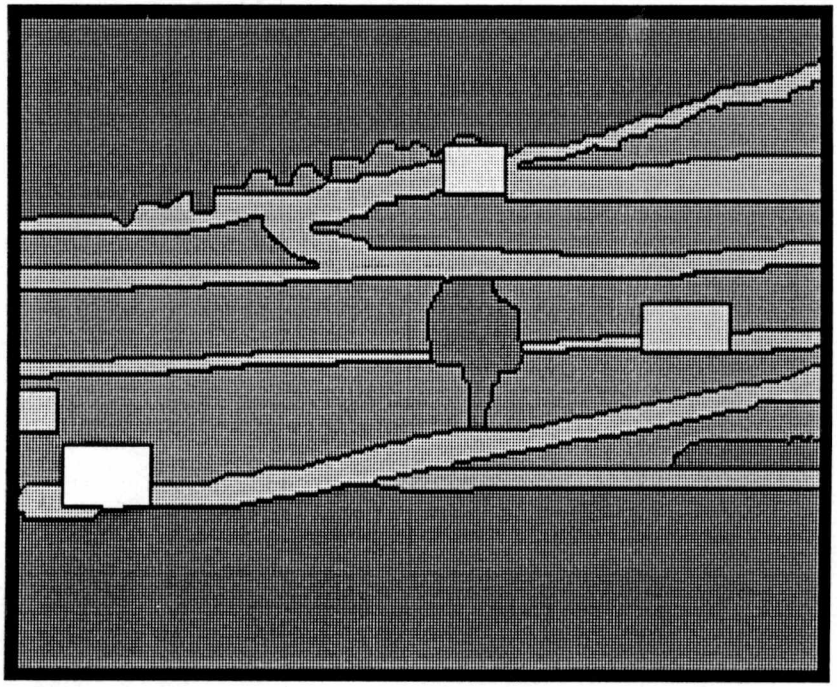


Figure 5.34: The ideal segmentation of image in Figure 5.33.

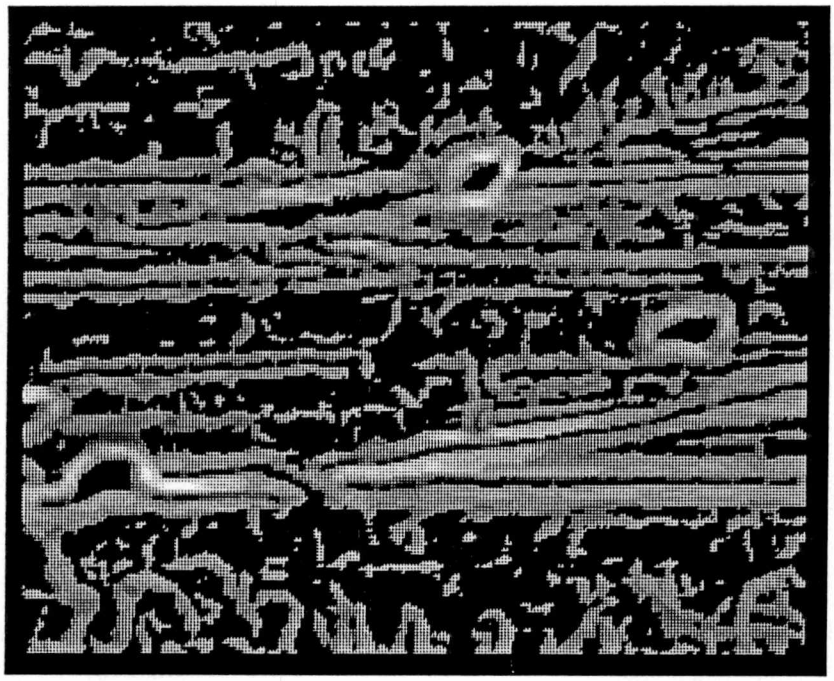
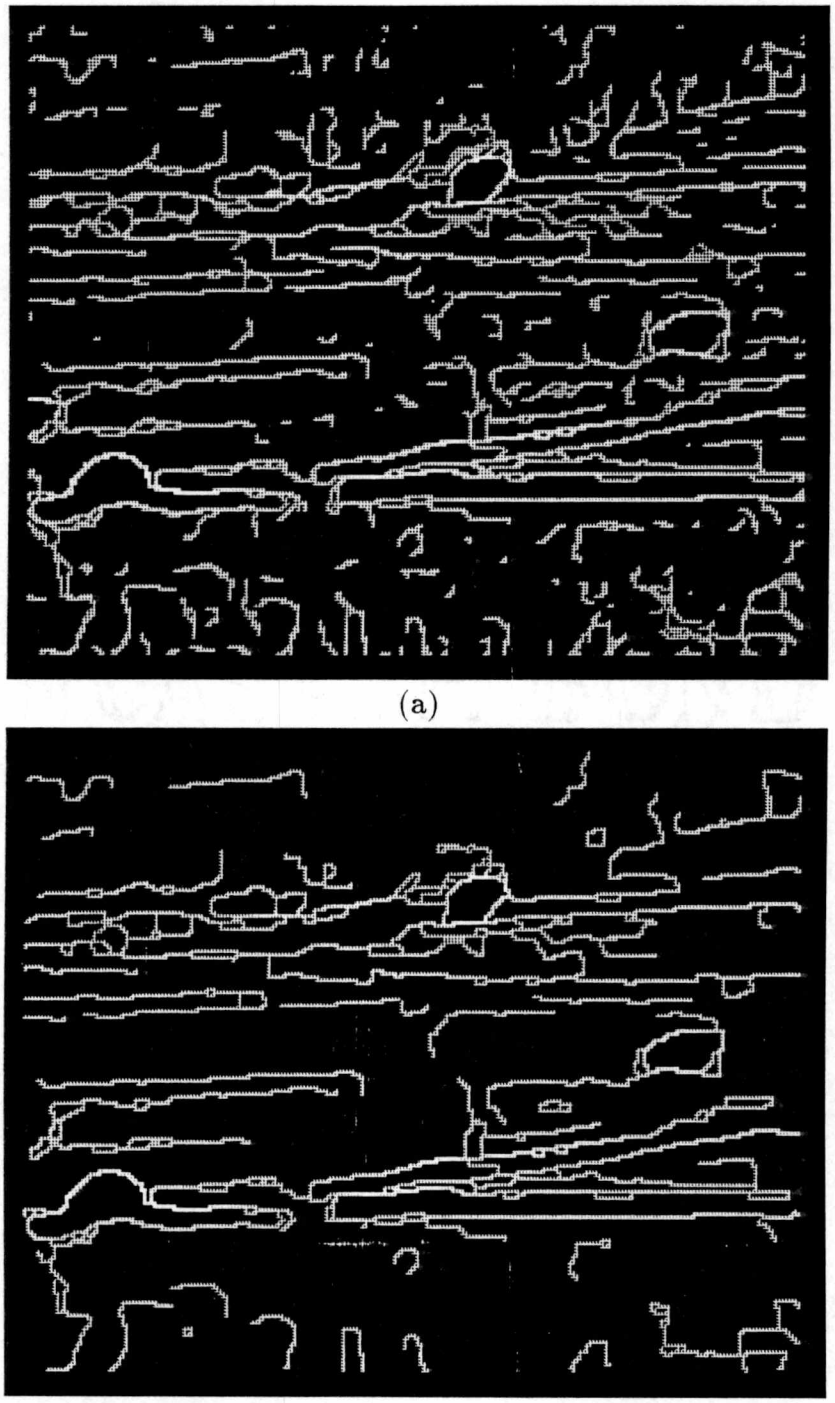


Figure 5.35: The edge image of Figure 5.33 using the 3 – 5 – 7 algorithm.



(a)

(b)

Figure 5.36: (a) The one-pixel-wide edge of Figure 5.35. (b) The result of (a) after edge refinement procedure with threshold TSL = 20 pixels.

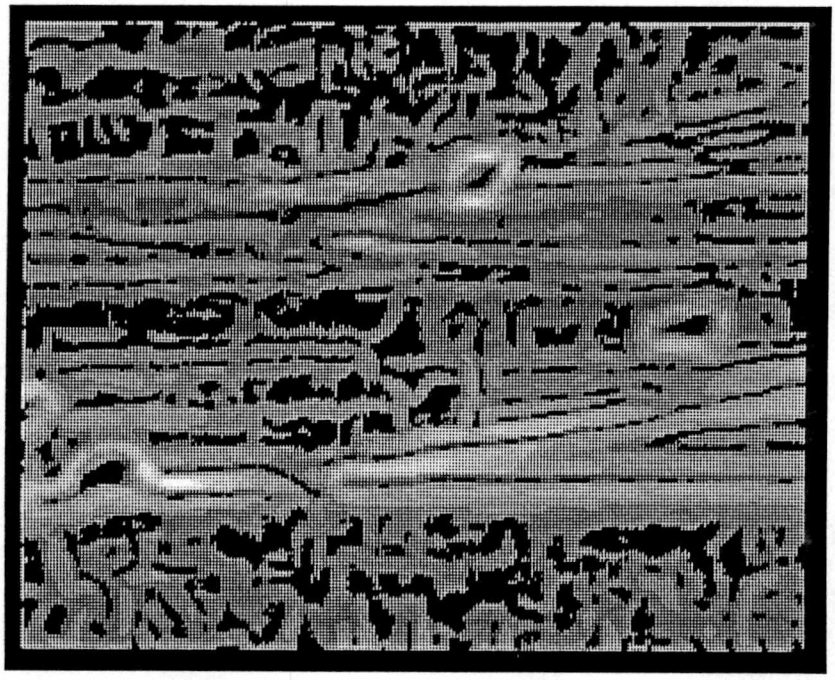
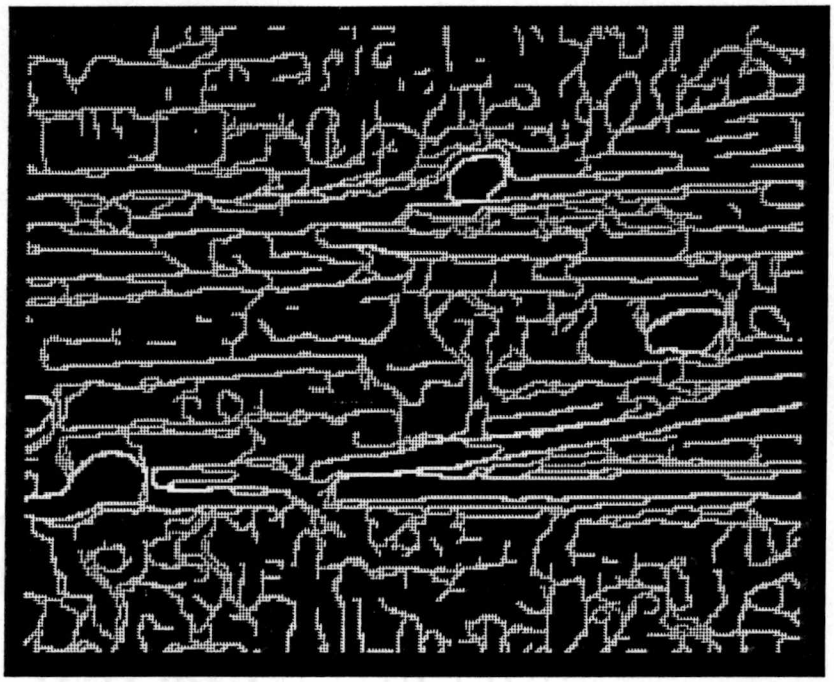
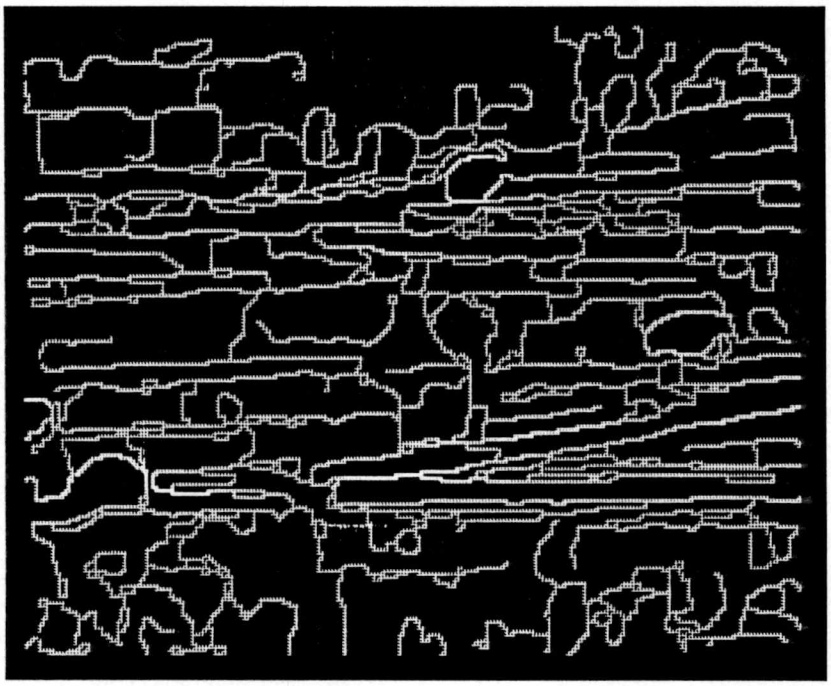


Figure 5.37: The edge image of Figure 5.33 using the 5 – 7 – 9 algorithm.



(a)



(b)

Figure 5.38: (a) The one-pixel-wide edge of Figure 5.37. (b) The result of (a) after edge refinement procedure with threshold TSL = 20 pixels.

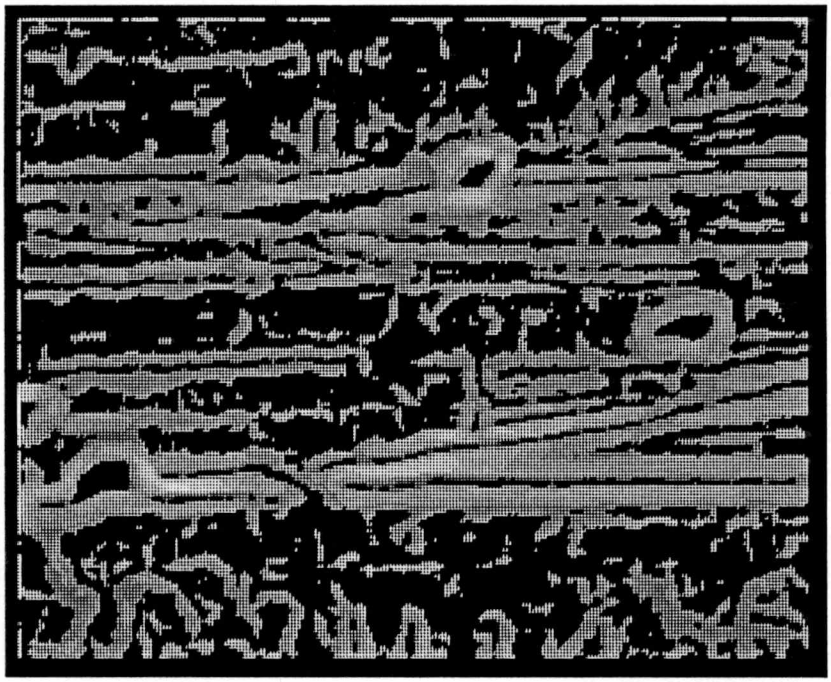
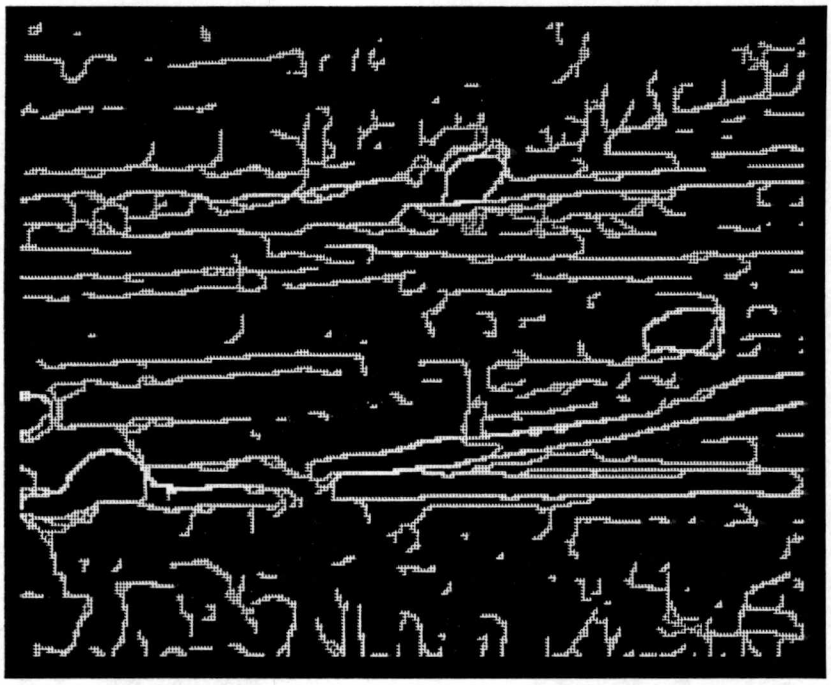
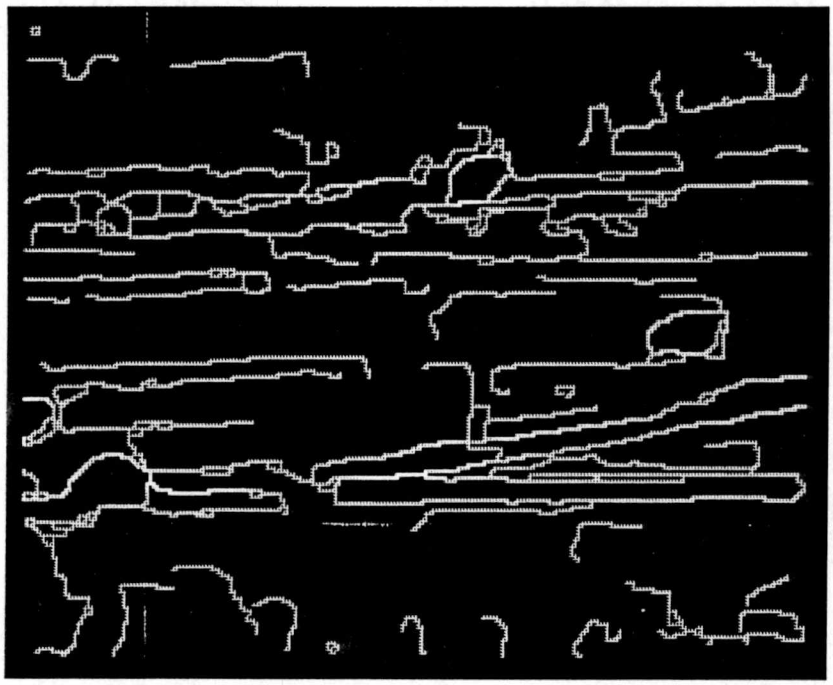


Figure 5.39: The edge image of Figure 5.33 using the 3 – 5 – 7 – 9 algorithm.



(a)



(b)

Figure 5.40: (a) The one-pixel-wide edge of Figure 5.39. (b) The result of (a) after edge refinement procedure with threshold TSL = 20 pixels.

5.2.1 Comparison

Observing the detected objects in the three infrared images in Figures 5.17, 5.25, and 5.33, we see that the resulting edge images of the 3 – 5 – 7 and the 3 – 5 – 7 – 9 combinations are similar; however, the 3 – 5 – 7 combination missed one object which is detected by the 3– 5 – 7 – 9 combination in each of these three infrared images. The three missing objects, the results obtained by the implementation of the 3 – 5 – 7 – 9 combination, and the subareas of the original infrared images are shown in Figure 5.41.

In finding roads, both the 3 – 5 – 7 and the 3 – 5 – 7 – 9 combinations detect and keep the connectivity of the roads. A special case occurs in the results of Figure 5.17. Both the 3 – 5 – 7 and the 3 – 5 – 7 – 9 combinations find connected roads in the central area of Figure 5.17, as shown in Figures 5.19 and 5.23, respectively. Because of different final edge direction assignments, the thinned edges of the 3 – 5 – 7 combination have a discontinuity, as shown in Figure 5.20(b). In contrast, the 3 – 5 – 7 – 9 combination detects continuous road edges, as shown in Figure 5.24(b).

In the case of three infrared images, the performance of the 5 – 7 – 9 combination is similar to the performance on the synthetic binary images described in Section 5.1.2. Most of the objects and roads are found by the 5 – 7 – 9

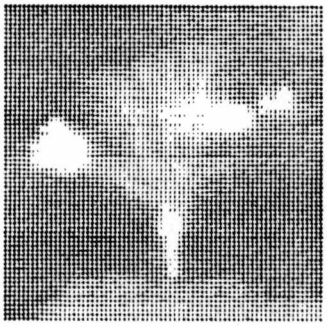
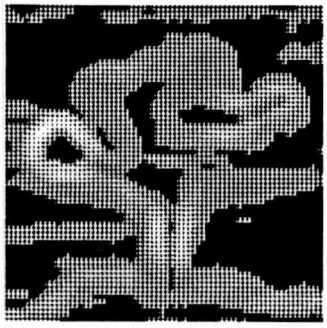


Figure 5.17



3 - 5 - 7
combination



3 - 5 - 7 - 9
combination

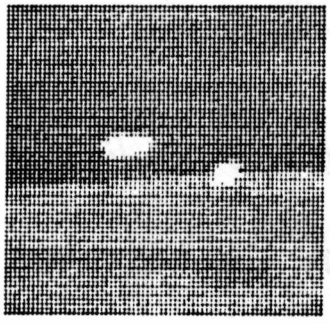


Figure 5.25



3 - 5 - 7
combination



3 - 5 - 7 - 9
combination

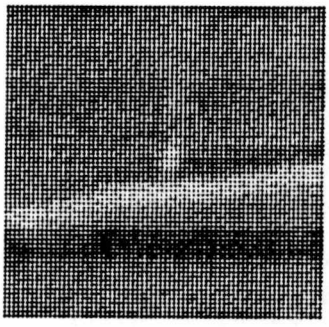
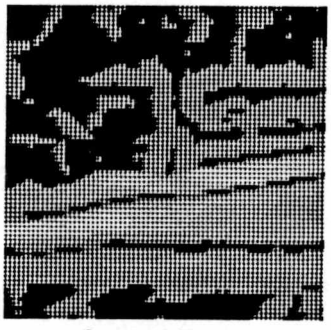


Figure 5.33



3 - 5 - 7
combination



3 - 5 - 7 - 9
combination

Figure 5.41: The comparison of the results of the 3 – 5 – 7 and the 3 – 5 – 7 – 9 combination algorithms.

combination. However, because of the large template size, many edge pixels are detected, which complicates the detection of objects of interest and roads. Even after the edge refinement with the segment length threshold TSL = 20 pixels, as shown in Figures 5.22(b), 5.30(b), and 5.38(b), the objects and roads are still ambiguous.

From the resulting edge images of the three infrared images, we can conclude the following: the 3 – 5 – 7 and the 3 – 5 – 7 – 9 rule-based edge detection algorithms are more effective than the 5 – 7 – 9 rule-based edge detection algorithm. Furthermore, the 3 – 5 – 7 – 9 rule-based edge detection algorithm yields better results than the 3 – 5 – 7 rule-based edge detection algorithm.

CHAPTER 6

CONCLUSIONS

This thesis presents an algorithm that combines local and regional edge detectors to locate the positions and directions of edge pixels in an image. The objective of this algorithm is to: 1) achieve low noise sensitivity; and 2) detect weak edges and maintain the connectivity within the detected edges. In the proposed algorithm, each image pixel is processed by using a set of templates to gain better localization and minimal noise sensitivity. Two sets of decision rules are used for each image pixel to determine whether the image pixel is an edge pixel or not. One set of decision rules examines the magnitude relationships between different template sizes for each image pixel, and the other set of decision rules examines the direction relationships. Each qualified edge pixel is assigned a final edge strength and an edge direction.

The proposed algorithm is flexible and detects edges with different edge complexities, intensity contrasts, and noise densities. The flexibility is provided by two adjustable parameters, the minimum increment threshold T_{edge} and the edge confidence threshold T_{ec}. Depending on the contextual structure of an image, optimal values of parameters can be used to extract the edges of interest in the image. However, for some very weak edges located in very noisy environments, detected edges may be disconnected. This problem can be overcome by using

an appropriate edge enhancement preprocessing procedure.

The experiments in Chapter 5 show that the proposed multi-template edge detection algorithm can extract the desired edges from both synthetic and infrared images. In general, the proposed algorithm is superior to edge detectors that employ single size templates. When processing synthetic images, the 3 – 5 – 7 template combination performs better than the 5 – 7 – 9 template combination, and the four-template combination (3 – 5 – 7 – 9) performs better than the two three-template combinations. Similar conclusions are drawn from the results of processing infrared images. Hence, the 3 – 5 – 7 template combination is the better of the two three-template combinations. However, the four-template edge detection algorithm, 3 – 5 – 7 – 9 template combination, is the most powerful multi-template edge detection algorithm.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] I. E. Abdou and W. K. Pratt, "Quantitative Design and Evaluation of Enhancement/Thresholding Edge Detectors," *Proceedings of IEEE*, vol. 67, no. 5, pp. 753-763, May 1979.

[2] A. Arcelli , L. P. Corolella, and S. Levialdi, "From Local Maximum to Connected Skeletons," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-3, no. 2, pp. 134-143, March 1981.

[3] C. Arcelli and G. S. D. Baja, "A Thinning Algorithm Based on Prominence Detection," *Pattern Recognition*, vol. 13, pp. 225-235, March 1981.

[4] C. Arcelli and G. S. *d.* Baja, "A Width-Independent Fast Thinning Algorithm," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-7, no. 4, pp. 463-474, July 1985.

[5] E. Argyle, "Techniques for Edge Detection," *Proceedings of IEEE*, vol. 59, pp. 285-286, February 1971.

[6] G. P. Ashkar and J. W. Modestino, "The Contour Extraction Problem with Biomedical Applications," *Computer Graphics and Image Processing*, vol. 7, no. 3, pp. 331-355, June 1978.

[7] D. H. Ballard, "Generating the Hough Transform to Detect Arbitray Shapes," *Pattern Recognition*, vol. 13, pp. 111-122, March 1981.

[8] D. H. Ballard and C. M. Brown, *Computer Vision*, Prentice-Hall Book Company, Englewood Clifs, New Jersey, 1982.

[9] M. Basseville, B. Espiau, and J. Gasnier, "Edge Detection Using Sequential Methods for Change in Level - Part I: A Sequential Edge Detection Algorithm," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 29, no. 1, pp. 24-31, February 1981.

[10] F. Bergholm, "Edge Focusing," *Proceedings of IEEE 8th International Conference on Pattern Recognition*, Paris, France, pp. 597-600, October 1986.

[11] T. O. Binford, "Inferring Surfaces from Images," *Artificial Intelligence*, vol. 17, pp. 205-244, August 1981.

[12] A. C. Bovik, T. S. Huang, and D. C. Munson, Jr., "A Generalization of Median Filtering Using Linear Combinations of Order Statistics," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-31, no. 6, pp. 1342-1350, December 1983.

[13] A. C. Bovik, T. S. Huang, and D. C. Munson, "The Effect of Median Filtering on Edge Estimation and Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-9, no. 2, pp. 181-194, March 1987.

[14] M. Brady, "Computational Approaches to Image Understanding," *Computing Surveys*, vol. 14, no. 1, pp. 3-71, March 1982.

[15] M. Brady and B. K. P. Horn, "Rotationally Symmetric Operators for Surface Interpolation," *Computer Vision, Graphics, and Image Processing*, vol. 22, no. 1, pp. 70-94, April 1983.

[16] J. B. Burns, A. R. Hanson, and E. M. Riseman, "Extracting Straight Lines," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 4, pp. 425-455, July 1986.

[17] J. F. Canny, "A Computational Approach to Edge Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 6, pp. 679-698, November 1986.

[18] P. C. Chen and T. Pavlidis, "Image Segmentation as an Estimation Problem," *Computer Graphics and Image Processing*, vol. 12, no. 2, pp. 153-172, February 1980.

[19] F. S. Cohen, D. B. Cooper, J. F. Silverman, and E. B. Hinkle, "Simple Parallel Hierarchical and Relaxation Algorithms for Segmenting Textured Images Based on Noncasual Markovian Random Field Models," *Proceedings of IEEE 7th International Conference on Pattern Recognition*, Montreal, Canada, pp. 1104-1107, July-August 1984.

[20] E. R. Davies, "Constraints on the Design of Template Masks for Edge Detection," *Pattern Recognition Letters*, vol. 4, pp. 111-120, April 1986.

[21] L. S. Davis, "A Survey of Edge Detection Techniques," *Computer Graphics and Image Processing*, vol. 4, no. 3, pp. 248-270, September 1975.

[22] E. S. Deutsh and J. R. Fram, "A Quantitative Study of the Orientation Bias of Some Edge Detector Schemes," *IEEE Transactions on Computers*, vol. C-27, no. 3, pp. 205-213, March 1978.

[23] R. O. Duda and P. E. Hart, "Use of the Hough Transformation to Detect Lines and Curves in Pictures," *Communications of the Association for Computing Machinery*, vol. 15, no. 1, pp. 11-15, January 1972.

[24] S. M. Dunn, D. Harwood, and L. S. Davis, "Local Estimation of the Uniform Error Threshold," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-6, no. 6, pp. 742-747, November 1984.

[25] C. R. Dyer and A. Rosenfeld, "Thinning Algorithms for Gray-Scaled Pictures," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-1, no. 1, pp. 88-89, January 1979.

[26] M. P. Ekstrom, *Digital Image Processing Techniques*, Academic Press Book Company, Orlando, Florida, 1984.

[27] D. Ernst, B. Bargel, and F. Holdermann, "Processing of Remote Sensing Data by a Region Growing Algorithm," *Proceedings of IEEE 3rd International Joint Conference on Pattern Recognition*, Coronado, California, pp. 679-683, November 1976.

[28] J. A. Feldman and Y. Yakimovsky, "Decision Theory and Artificial Intelligence - I: A Semantics-Based Region Analyzer," *Artificial Intelligence*, vol. 5, pp. 349-371, Winter 1974.

[29] J. R. Fram and E. W. Deutsch, "On the Quantitative Evaluation of Edge Detection Schemes and Their Comparison with Human Performance," *IEEE Transactions on Computers*, vol. C-24, no. 6, pp. 616-628, June 1975.

[30] W. Frei and C. C. Chen, "Fast Boundary Detection: A Generalization and a New Algorithm," *IEEE Transactions on Computers*, vol. C-26, no. 10, pp. 988-998, October 1977.

[31] K. S. Fu, R. C. Gonzalez, and C. S. G. Lee, *Robotics: Control, Sensing, Vision, and Intelligence*, McGraw-Hill Book Company, New York, New York, 1987.

[32] M. A. Furst and P. E. Caines, "Edge Detection with Image Enhancement via Dynamic Programming," *Computer Vision, Graphics, and Image Processing*, vol. 33, no. 1, pp. 263-279, January 1986.

[33] N. C. Gallagher, Jr., and G. L. Wise, "A Theoretical Analysis of the Properties of Median Filters," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-29, no. 6, pp. 1136-1141, December 1981.

[34] S. Geman and D. Geman, "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-6, no. 6, pp. 721-741, November 1984.

[35] M. A. Gennert, "Detecting Half-Edges and Vertices in Images," *Proceedings of IEEE on Computer Vision and Pattern Recognition*, Miami Beach, Florida, pp. 552-557, June 1986.

[36] R. C. Gonzalez and P. Wintz, *Digital Image Processing*, Addison Wesley Book Company, Reading, Massachusetts, pp. 337, 426, 1987.

[37] A. K. Griffith, "Edge Detection in Simple Scenes Using a priori Information," *IEEE Transactions on Computers*, vol. C-22, no. 4, pp. 371-381, April 1973.

[38] J. N. Gupta and P. A. Wintz, "Computer Processing Algorithm for Locating Boundaries in Digital Pictures," *Proceedings of IEEE 2nd International Joint conference on Pattern Recognition*, Copenhagen, Denmark, pp. 155-156, August 1974.

[39] R. M. Haralick, "Digital Step Edges from Zero Crossing of the Second Directional Derivatives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-6, no. 1, pp. 58-68, January 1984.

[40] R. Hartley, "A Gaussian-Weighted Multiresolution Edge Detector," *Computer Vision, Graphics, and Image Processing*, vol. 30, no. 1, pp. 70-83, April 1985.

[41] E. C. Hildreth, "The Detection of Intensity Changes by Computer and Biological Vision Systems," *Computer Vision, Graphics, and Image Processing*, vol. 22, no. 1, pp. 1-27, April 1983.

[42] B. K. P. Horn, "Understanding Image Intensities," *Artificial Intelligence*, vol. 8, pp. 201-231, April 1977.

[43] T. Hong, M. Shneier, R. Hartley, and A. Rosenfeld, "Using Pyramids to Detect Good Continuation," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-13, no. 4, pp. 631-635, July-August 1983.

[44] S. L. Horowitz and T. Pavlidis, "Picture Segmentation by a Directed Split-and-Merge Procedure," *Proceedings of IEEE 2nd International Joint Conference on Pattern Recognition*, Copenhagen, Denmark, pp. 424-433, August 1974.

[45] S. L. Horowitz and T. Pavlidis, "Picture Segmentation by a Tree Traversal Algorithm," *Journal of the Association for Computing Machinery*, vol. 23, no. 2, pp. 368-388, April 1976.

[46] M. H. Hueckel, "An Operator which Locates Edges in Digitized Pictures," *Journal of Association for Computing Machinery*, vol. 18, no. 1, pp. 113-125, January 1971.

[47] M. H. Hueckel, "A Local Visual Operator which Recognizes Edges and Lines," *Journal of Association for Computing Machinery*, vol. 20, no. 4, pp. 634-647, October 1973.

[48] A. Huertas and G. Medioni, "Detection of Intensity Changes with Subpixel Accuracy Using Laplacian-Gaussian Masks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 5, pp. 651-664, September 1986.

[49] C. J. Jacobus and R. T. Chien, "Two New Edge Detectors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-3, no. 5, pp. 581-592, September 1981.

[50] B. I. Justusson, "Median Filtering: Statistical Properties," *Two-Dimensional Digital Signal Processing II: Transforms and Median Filters*, T. S. Huang, Editor, Springer-Verlag Book Company, New York, New York, pp. 161-196, 1981.

[51] R. Kirsch, "Computer Determination of the Constituent Structure of Biological Images," *Computers and Biomedical Research*, vol. 4, no. 3, pp. 315-328, June 1971.

[52] M. Kocher and M. Kunt, "A Contour-Texture Approach to Picture Coding," *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, Palais Des Congres, Paris, France, pp. 436-439, May 1982.

[53] J. J. Koenderink, "The Structure of Images," *Biological Cybernetics*, vol. 50, pp. 363-370, August 1984.

[54] A. Kundu and S. K. Mitra, "A New Algorithm for Image Edge Extraction Using a Statistical Classifier Approach," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-9, no. 4, pp. 569-577, July 1987.

[55] M. Kunt, "Source Coding of X-Ray Pictures," *IEEE Transactions on Biomedical Engineering*, vol. BME-25, no. 2, pp. 121-138, March 1978.

[56] M. Kunt, "Edge Detection: A Tutorial Review," *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, Palais Des Congres, Paris, France, pp. 1172-1175, May 1982.

[57] V. Lacroix, "A Three-Module Strategy for Edge Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-10, no. 6, pp. 803-810, November 1988.

[58] J. M. Lester, H. A. Williams, B. A. Weintraub, and J. F. Brenner, "Two Graph Searching Techniques for Boundary Finding in White Blood Cell Images," *Computers in Biology Medicine*, vol. 8, no. 4, pp. 293-308, October 1978.

[59] A. Lev, S. W. Zucker, and A. Rosenfeld, "Iterative Enhancement of Noisy Images," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-7, no. 6, pp. 435-442, June 1977.

[60] E. P. Lyvers and O. R. Mitchell, "Precision Edge Contrast and Orientation Estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-10, no. 6, pp. 927-937, November 1988.

[61] I. D. G. MacLeod, "Comments on 'Techniques for Edge Detection'," *Proceedings of IEEE*, vol. 60, p. 344, March 1972.

[62] A. R. Mansouri, A. S. Malowany, and M. D. Levine. "Line Detection in Digital Pictures: A Hypothesis Prediction/Verification Paradigm," *Computer Vision, Graphics, and Image Processing*, vol. 40, no. 1, pp. 95-114, October 1987.

[63] K. V. Mardia and T. J. Hainsworth, "A Spatial Thresholding Method for Image Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-10, no. 6, pp. 919-927, November 1988.

[64] D. Marr and T. Poggio, "A Computational Theory of Human Stereo Vision," *Proceedings of the Royal Society of London*, London, vol. B204, pp. 301-328, May 1979.

[65] D. Marr and E. Hildreth, "Theory of Edge Detection," *Proceedings of the Royal Society of London*, London, vol. B207, pp. 187-217, February 1980.

[66] A. Martelli, "Edge Detection Using Heuristic Search Methods," *Computer Graphics and Image Processing*, vol. 1, no. 2, pp. 169-182, August 1972.

[67] A. Martelli, "An Application of Heuristic Search Methods to Edge and Contour Detection," *Communications of the Association for Computing Machinery*, vol. 19, no. 2, pp. 73-83, February 1976.

[68] D. Mason, I. Lauder, D. Rutowitz, and G. Spowart, "Measurement of C-bands in Human Chromosomes," *Computers in Biology Medicine*, vol. 5, no.3, pp. 179-201, September 1975.

[69] G. Medioni and Y. Yasumoto, "Corner Detection and Curve Representation Using Cubic B-Splines," *Computer Vision, Graphics, and Image Processing*, vol. 39, no. 3, pp. 267-278, September 1987.

[70] L. Mero and Z. Vassy, "A Simplified and Fast Version of the Hueckel Operator for Finding Optimal Edges in Pictures," *Proceedings of 4th International Joint Conference on Artificial Intelligence*, Tbilisi, Georgia, USSR, pp. 650-655, September 1975.

[71] J. W. Modestino and R. W. Fries, "Edge Detection in Noisy Images Using Recursive Digital Filtering," *Computer Graphics and Image Processing*, vol. 6, no. 5, pp. 409-433, October 1977.

[72] J. Montanari, "On the Optimal Detection of Curves in Noisy Pictures," *Communications of the Association for Computing Machinery*, vol. 14, no. 5, pp. 335-345, May 1971.

[73] M. Nagao and T. Matsuyama, "Edge Preserving Smoothing," *Computer Graphics and Image Processing*, vol. 9, no. 4, pp. 394-407, April 1979.

[74] R. N. Nagel and A. Rosenfeld, "Steps Toward Handwritten Signature Verification," *Proceedings of IEEE 1st International Joint Conference on Pattern Recognition*, Washington, D.C., pp. 59-66, October-November 1973.

[75] V. S. Nalwa and T. O. Binford, "On Detecting Edges," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 6, pp. 699-714, November 1986.

[76] V. S. Nalwa, "Edge-Detector Resolution Improvement by Image Interpolation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-9, no. 3, pp. 446-451, May 1987.

[77] R. Nevatia, "Evaluation of a Simplified Hueckel Edge - Line Detector," *Computer Graphics and Image Processing*, vol. 6, no. 6, pp. 582-588, December 1977.

[78] R. Nevatia and K. R. Babu, "Linear Feature Extraction and Description," *Computer Graphics and Image Processing*, vol. 13, no. 3, pp. 257-269, July 1980.

[79] T. A. Nodes and N. C. Gallagher, "Two-Dimensional Root Structures and Convergence Properties of the Separable Median Filter," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-31, no. 6, pp. 1350-1365, December 1983.

[80] D. P. Panda, "Nonlinear Smoothing of Pictures," *Computer Graphics and Image Processing*, vol. 8, no. 2, pp. 259-270, October 1978.

[81] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*, McGraw-Hill Book Company, New York, New York, pp. 104, 151, 1984.

[82] R. Patton and D. Brzakovic, "Rule Based Composite Gradient Edge Operator," *Proceedings of SPIE's 1987 Technical Symposium Southeast on Optics, Electro-Optics, and Sensors*, Orlando, Florida, 1987.

[83] T. Pavlidis, "SURVEY: A Review of Algorithms for Shape Analysis," *Computer Graphics and Image Processing*, vol. 7, no. 2, pp. 243-258, April 1978.

[84] T. Peli and D. Malah, "A Study of Edge Detection Algorithms," *Computer Graphics and Image Processing*, vol. 20, no. 1, pp. 1-21, September 1982.

[85] I. Pitas and A. N. Venetsanopoulos, "Edge Detectors Based on Nonlinear Filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 4, pp. 538-549, July 1986.

[86] W. K. Pratt, *Digital Image Processing*, Wiley Book Company, New York, New York, pp. 307-344, 1978.

[87] J. M. Prewitt, "Object Enhancement and Extraction," *Picture Processing and Psychopictorics*, B. S. Lipkin and A. Rosenfeld, Editors, Academic Book Company, New York, New York, pp. 75-149, 1970.

[88] T. Pun, "Entropic Thresholding: A New Approach," *Computer Graphics and Image Processing*, vol. 16, no. 3, pp. 210-239, July 1981.

[89] T. W. Ridler and S. Calvard, "Picture Thresholding Using an Iterative Selection Method," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-8, no. 8, pp. 630-632, August 1978.

[90] L. G. Roberts, "Machine Perception of Three-Dimensional Solids," *Optical and Electro-Optical Information Processing*, J. T. Tippett *et al.*, Editor, Massachusetts Institute of Technology Press Book Company, Cambridge, Massachusetts, pp. 159-197, 1965.

[91] G. S. Robinson, "Edge Detection by Compass Gradient Masks," *Computer Graphics and Image Processing*, vol. 6, no. 5, pp. 492-501, October 1977.

[92] A. Rosenfeld, "A Nonlinear Edge Detection Technique," *Proceedings of IEEE*, vol. 58, pp. 814-816, May 1970.

[93] A. Rosenfeld, "Survey : Image Analysis and Computer Vision: 1987," *Computer Vision, Graphics, and Image Processing*, vol. 42, pp. 234-293, April 1988.

[94] A. Rosenfeld and M. Thurston, "Edge and Curve Detection for Visual Scene Analysis," *IEEE Transactions on Computers*, vol. C-20, no. 5, pp. 562-569, May 1971.

[95] A. Rosenfeld, R. A. Hummel and S. W. Zucker, "Scene Labeling by Relaxation Operations." *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-6, no. 6, pp. 420-453, June 1976.

[96] A. Rosenfeld and A. Kak, *Picture Processing by Computers*, Academic Book Company, New York, New York, 1982.

[97] A. Rosenfeld and A. Kak, *Digital Picture Processing*, vol. 2, Academic Book Company, New York, New York, pp. 84-112, 1982.

[98] E. Salari and P. Siy, "The Ridge-Seeking Method for Obtaining the Skeleton of Digital Images," *IEEE Transactions on System, Man, and Cybernetics*, vol. SMC-14, no. 3, pp. 524-528, May 1984.

[99] A. Scher, F. R. Dias Velasco, and A. Rosenfeld, "Some New Image Smoothing Techniques," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-10, no. 3, pp. 153-158, March 1980.

[100] K. S. Shanmugam, F. M. Dickey, and J. A. Green, "An Optimal Frequency Domain Filter for Edge Detection in Digital Images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-1, no. 1, pp. 37-49, January 1979.

[101] G. B. Shaw, "Local and Regional Edge Detectors: Some Comparisons," *Computer Graphics and Image Processing*, vol. 9, no. 2, pp. 135-149, February 1979.

[102] Y. Shirai, "Analyzing Intensity Arrays Using Knowledge About Scenes," *Psychology of Computer Vision*, P. H. Wintson, Editor, McGraw-Hill Book Company, New York, New York, pp. 93-113, 1975.

[103] J. Sklansky, "On the Hough Techniques for Curve Detection," *IEEE Transactions on Computers*, vol. C-27, no. 10, pp. 923-926, October 1978.

[104] V. Torre and T. A. Poggio, "On Edge Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 2, pp. 147-163, March 1986.

[105] S. G. Tyan, "Median Filtering: Deterministic Properties," *Two-Dimensional Digital Signal Processing II: Transforms and Median Filters*, T. S. Huang, Editor, Springer-Verlag Book Company, New York, New York, pp. 197-217, 1981.

[106] F. M. Wahl, *Digital Image Signal Processing*, Artech House Book Company, Norwood, Massachusetts, pp. 119-149, 1987.

[107] D. C. Wang and A. H. Vagnucci, "Gradient Inverse Weighted Smoothing Scheme and the Evalution of its Performance," *Computer Graphics and Image Processing*, vol. 15, no. 2, pp. 167-181, February 1981.

[108] S. Watanabe and the CYBEST Group, "An Automated Apparatus for Cancer Prescreening: CYBEST," *Computer Graphics and Image Processing*, vol. 3, no. 4, pp. 350-358, December 1974.

[109] J. S. Weszka, R. N. Nagel, and A. Rosenfeld, "A Threshold Selection Technique," *IEEE Transactions on Computers*, vol. C-23, no. 12, pp. 1322-1326, December 1974.

[110] J. S. Weszka, "A Survey of Threshold Selection Techniques," *Computer Graphics and Image Processing*, vol. 7, no. 2, pp. 259-265, April 1978.

[111] A. P. Witkin, "Scale Space Filtering," *Proceedings of 8th International Joint Conference on Artificial Intelligence*, Karlsruhe, West Germany, pp. 1019-1021, August 1983.

[112] Y. Yakimovsky, "On the Recognition of Complex Structures: Computer Software Using Artificial Intelligence Applied to Pattern Recognition," *Proceedings of IEEE 2nd International Joint Conference on Pattern Recognition*, Copenhagen, Denmark, pp. 345-353, August 1974.

[113] Y. Yakimovsky, "Boundary and Object Detection in Real World Images," *Journal of Association for Computing Machinery*, vol. 23, no. 4, pp. 598-619, October 1976.

[114] T. Y. Young and K. S. Fu, *Handbook of Pattern Recognition and Image Processing*, Academic Press Book Company, Orlando, Florida, 1986.

[115] T. Y. Zhang and C. Y. Suen, "A Fast Parallel Algorithm for Thinning Digital Patterns," *Communications of the Association for Computing Machinery*, vol. 27, no. 3, pp. 236-242, March 1984.

[116] S. Zucker, R. Hummel, and A. Rosenfeld, "An Application of Relaxation Labelling to Line and Curve Enhancement," *IEEE Transactions on Computers*, vol. C-26, no. 4, pp. 394-403, April 1977.

APPENDIX

APPENDIX A

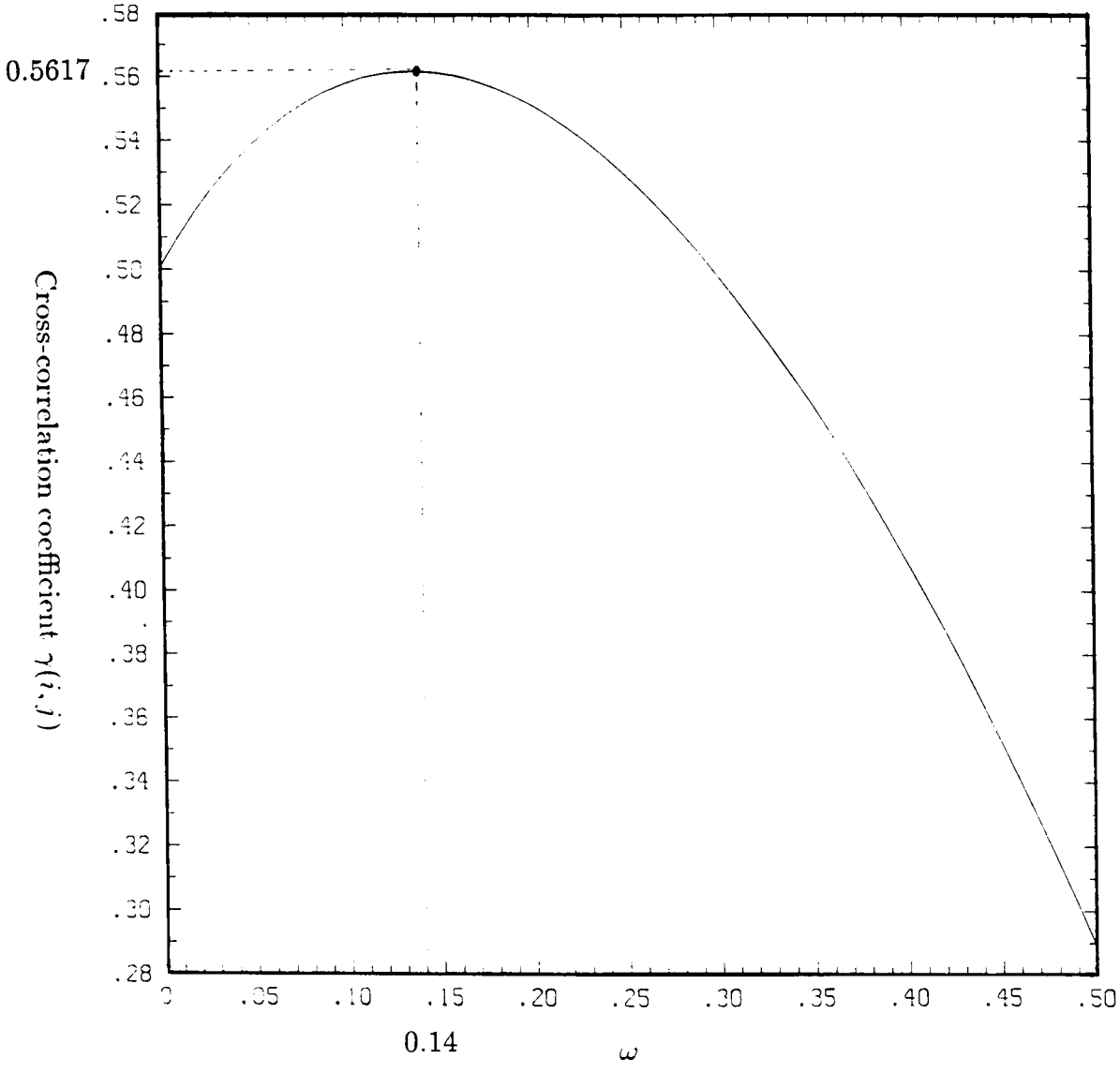


Figure A.1: The cross-correlation coefficient of Equation 3.32 on page 31.

VITA

Ruyih Louis Wang was born in Taipei, Taiwan, R.O.C., on January 31, 1961. He was raised and educated in Taipei. He entered the Chung-cheng High School in 1976. In 1977, he received the Mathematics Award, which was given each year to the student with the highest mathematics examination score. In 1978, he was the Thomas Alva Edison Scientific Award recipient. He graduated from high school in June 1979 and then entered the Tamkang University. In 1980, 1981, and 1982, he was a recipient of the Ta Yu Award which was distributed each semester to outstanding engineering students. In June 1983, he received a Bachelor of Science degree in Electronic Engineering.

From 1983 to 1985, he served in the R.O.T.C. as a communication officer. In December 1986, he entered The University of Tennessee, Knoxville, as a graduate student and became a graduate research assistant in January 1988. In May 1989, the author received his Master of Science degree in Electrical and Computer Engineering.