

INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

Bell & Howell Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

UMI[®]

Automatic Target Recognition Using Location Uncertainty

Gang Liu

A dissertation submitted in partial fulfillment
of the requirements for the degree of

Doctor of Philosophy

University of Washington

2000

Program Authorized to Offer Degree: Department of Electrical Engineering

UMI Number: 9995402



UMI Microform 9995402

Copyright 2001 by Bell & Howell Information and Learning Company.

All rights reserved. This microform edition is protected against
unauthorized copying under Title 17, United States Code.

Bell & Howell Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

In presenting this dissertation in partial fulfillment of the requirements for the Doctoral degree at the University of Washington, I agree that the Library shall make its copies freely available for inspection. I further agree that extensive copying of this dissertation is allowable only for scholarly purposes, consistent with "fair use" as prescribed in the U.S. Copyright Law. Requests for copying or reproduction of this dissertation may be referred to Bell and Howell Information and Learning, 300 North Zeeb Road, Ann Arbor, MI 48106-1346, to whom the author has granted "the right to reproduce and sell (a) copies of the manuscript in microform and/or (b) printed copies of the manuscript made from microform."

Signature



Date

12/12/2000

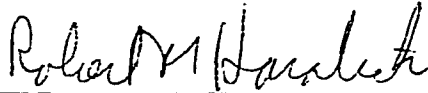
University of Washington
Graduate School

This is to certify that I have examined this copy of a doctoral dissertation by

Gang Liu

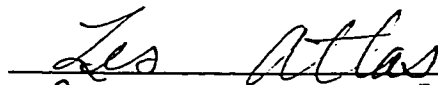
and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Chair of Supervisory Committee:

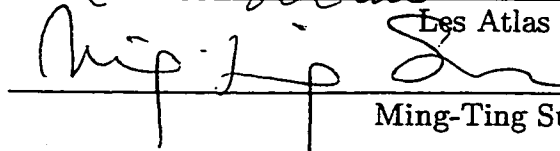


Robert M. Haralick

Reading Committee:



Les Atlas



Ming-Ting Sun

Date:

12-13-2000

University of Washington

Abstract

Automatic Target Recognition Using Location Uncertainty

by Gang Liu

Chair of Supervisory Committee:

Professor Robert M. Haralick
Department of Electrical Engineering

In this dissertation, we present a framework for using location uncertainty information in computer vision applications. This framework is applied to the military automatic target detection and recognition application. We take a model-based approach to accumulating weak but consistent target evidence. Reliable target detection and recognition is achieved by making use of the location uncertainty information not being utilized by existing algorithms. The development of the location uncertainty measure consists of three major pieces: the relative importance of boundary points as determined by the geometric relationship between the location uncertainty of the centroid and boundary points, the relationship between the signal-to-noise ratio and the location uncertainty at the boundary points, and the optimal estimation of the image gradient at the boundary points. With sound mathematical models, the study of these sub-problems yield meaningful results useful not only in this framework, but in many other general problems as well.

The results of our experiments with real and simulated image data show that the centroid location uncertainty feature computed by the proposed framework is very effective in target detection and recognition. As a powerful addition to existing

automatic target recognition algorithm modules, it has been successfully combined with the traditional matched filter to give further improved target detection and recognition performance.

Performance evaluation is always an important part in any new algorithm development. For characterizing the detection and recognition performance of computer vision algorithms, a new methodology is developed to overcome some problems with existing methods. An optimal matching problem is formulated to describe the situation. It is then transformed into an unconstrained assignment problem which enjoys an efficient solution technique: the Hungarian algorithm. This results in a one-to-one correspondence between ground-truth and declared entities and yields more precise performance measures.

TABLE OF CONTENTS

List of Figures	v
List of Tables	x
Chapter 1: Introduction	1
1.1 Computer vision research	1
1.2 The FLIR ATR application	2
1.3 Overview of previous work	7
1.4 Motivations for this study	12
1.5 Summary of major contributions	17
1.6 Organization of the dissertation	19
Chapter 2: Literature review	21
2.1 Traditional algorithms	22
2.2 Model-based algorithms	24
2.3 Neural networks in ATR	28
2.4 Multi-sensor data fusion in ATR	29
2.5 Performance characterization in ATR	31
Chapter 3: Polygon Centroid Location Uncertainty	33
3.1 Introduction	33
3.2 Geometric relationship between a polygon's vertices and its centroid .	34
3.3 Location uncertainty on the vertices	35

3.4	Uncertainty on the centroid location	37
3.5	Centroid uncertainty validation experiment	41
3.6	Detection performance prediction	44
Chapter 4:	Optimal Estimation of Image Gradient	55
4.1	Introduction	56
4.2	Gradient operator performance characterization	61
4.3	Optimal gradient operator design	72
4.4	Optimizing the Sobel-class gradient operator	75
4.5	Optimizing the IDGO	80
4.6	Optimizing the dG gradient operator	84
4.7	Discussions	86
4.8	Validation experiment for the MSE computation	95
4.9	Experiment on the smoothed edge	99
4.10	Conclusion	104
Chapter 5:	Edgel Location Uncertainty Characterization	106
5.1	Introduction and background	106
5.2	Edgel location estimation	111
5.3	Tabulating edgel location uncertainty	118
5.4	Validation experiment and correlated noise	122
5.5	Effect of slope and displacement of the zero-crossing	127
5.6	Summary	130
Chapter 6:	Constrained Matching Problem in Performance Evalua-	
	tion	134
6.1	Introduction	134
6.2	Performance evaluation problems in edge detection	135

6.3	Unconstrained optimal assignment problem	139
6.4	The Hungarian algorithm	140
6.5	Constrained case — optimal matching problem	141
6.6	Association procedure	145
6.7	Experiment	147
6.8	ATR performance evaluation	151
6.9	Summary	154
Chapter 7:	Simulated FLIR scene generation	159
7.1	Introduction	159
7.2	Scene background	160
7.3	Insertion of targets	160
7.4	Target signature simulation	162
7.5	Target contrast and dynamic range	165
Chapter 8:	FLIR ATR Experiments	171
8.1	Centroid uncertainty based FLIR ATR algorithm	172
8.2	Competing algorithms	176
8.3	Results and discussions	186
8.4	Experiments on uncooled FLIR image sequence data	195
8.5	Experiments on vehicle detection in aerial imagery	205
Chapter 9:	Conclusion	214
9.1	Brief summary	214
9.2	Short-comings and future work	216
Bibliography		218

Appendix A: Facet model order determination by MDL principle	237
A.1 The MDL principle	237
A.2 Facet model order determination	240
A.3 Experiment	241

LIST OF FIGURES

1.1	Example of a second generation FLIR image of a ground vehicle. . . .	4
3.1	Illustration of the computation of polygon centroid.	36
3.2	Data used in polygon centroid uncertainty validation experiment . . .	43
3.3	sample histograms for test statistic in centroid uncertainty validation experiment	45
3.4	Sample distribution of local SNR for target boundary and background in simulated FLIR image chips.	47
3.5	Distributions of the centroid uncertainty from Monte Carlo experiment with independent sampling and from values directly computed from image chips.	48
3.6	Correlation coefficient for SNR values at adjacent locations along the boundary of a particular target.	49
3.7	Distributions of the centroid uncertainty from Monte Carlo experiment with correlated SNR sampling and from values directly computed from image chips.	50
4.1	Comparison of two estimators for the input SNR.	66
4.2	Probability density and variance for the orientation distribution. . . .	69
4.3	Illustration of the square-aperture sampling	71
4.4	K_3 , κ_0 and errors for the optimal 3×3 operator.	78
4.5	K_5 , κ_0 and errors for the optimal 5×5 Sobel-class operators.	79

4.6	Illustration of the IDDGO. The integration of the directional derivative is done within the dashed line denoted rectangular area.	80
4.7	L , κ_0 and errors for the optimal 5×5 IDDGO.	85
4.8	s , κ_0 and errors for the optimal 5×5 dG operator.	87
4.9	Coefficients in the mask and estimation errors for the optimal fourth class operator.	89
4.10	Comparison of gradient operator classes optimally tuned to given input SNR levels.	90
4.11	Row derivative kernels for the 5×5 Sobel operator	92
4.12	Row derivative kernel for the 5×5 IDDGO with $L = 1.841$	92
4.13	Row derivative kernels for the 5×5 dG operator	93
4.14	Row derivative kernels for the optimal 5×5 operator obtained by numerical search (the fourth class operator).	94
4.15	Comparison of gradient operators for the low SNR range.	96
4.16	Comparison of gradient operators for the high SNR range.	97
4.17	Comparison of gradient operator optimization results by Monte Carlo simulation and by numerical evaluation of analytical expressions. . . .	100
4.18	Plots of sigmoid function with different blur parameters.	101
4.19	Comparison of gradient operator classes optimally tuned to given blur parameters.	103
4.20	Intensity values of very significantly blurred edge and the Prewitt gradient operator mask.	104
5.1	Step edge detection scenario.	111
5.2	Illustration of the approximation error in assuming the independence of \hat{v} from $\hat{\theta}$	114

5.3	The distribution of orientation and location on the noisy unit-contrast step edge.	116
5.4	The distribution of orientation and location on the unit-variance Gaussian noise.	117
5.5	Uncertainty of the edgel location estimate.	120
5.6	Simplified situation that explains the increase in edgel location uncertainty along the true gradient orientation.	121
5.7	Empirically observed scatter of edgel location estimate.	123
5.8	Comparison of empirical and predicted edgel location uncertainty. . .	124
5.9	Impact of noise correlation on the relationship between the edgel location uncertainty and the input SNR.	126
5.10	Probability for observing positively sloped zero-crossing for the authentic step edge.	129
5.11	Edgel location uncertainty increases as the true edgel location moves away from the center of the neighborhood.	131
6.1	Example showing the failure of the distance transform based method to give accurate performance measures.	136
6.2	Edge maps used in comparing performance evaluation methods. . . .	149
6.3	Misdetetection and false alarm edge pixels in the edge maps.	150
6.4	Targets in the ATR performance evaluation example.	155
7.1	Images in the Chinalake data set used as background scenes in the FLIR scene generation.	161
7.2	Some of the hand-segmented target image chips from the MURI FLIR data set.	164
7.3	Synthesized texture for target signature simulation.	166

7.4	Simulated target signature with corruption of additive texture.	167
7.5	Example of simulated FLIR scene.	170
8.1	The centroid uncertainty needs to be normalized to reduce improperly biased preference toward certain target shapes over others.	174
8.2	Examples in computing the matched filters for targets and background.	181
8.3	Feature vector distribution in combining centroid uncertainty and matched filter for FLIR ATR.	184
8.4	ROC curves for classifier performance in combining centroid uncer- tainty and matched filter for FLIR ATR.	185
8.5	Detection performance (ROC curves) of the ATR algorithms.	190
8.6	Recognition performance of the ATR algorithms.	191
8.7	Localization performance of the ATR algorithms.	192
8.8	Target instances for three different poses used in modeling the target in the Raytheon data experiment.	198
8.9	First frame in the t613loop1 sequence and detection result by the combined algorithm.	201
8.10	Detection performance of the ATR algorithms on the t613loop1 se- quence.	202
8.11	Example of simulated scene and detection result by the combined al- gorithm.	204
8.12	Detection performance of the ATR algorithms on data simulated from the scenes in the t613loop1 sequence with 10% texture.	206
8.13	Detection performance of the ATR algorithms on data simulated from the scenes in the t613loop1 sequence with 30% texture.	207
8.14	Detection performance of the ATR algorithms on data simulated from the scenes in the t613loop1 sequence with 50% texture.	208

8.15	Example aerial image for vehicle detection.	209
8.16	Vehicle detection ROC curves.	212
A.1	Test image and estimates made by the cubic facet model.	242
A.2	Adaptive model order determination by the MDL-based procedure. .	244

LIST OF TABLES

3.1	Polygon and perturbation in centroid uncertainty validation experiment	42
6.1	Performance measures on edge detection by two performance evaluation methods.	148
6.2	Targets in the ATR performance evaluation example	156
6.3	Distance matrix for the first-stage (recognition) correspondence . . .	157
6.4	Targets that are both correctly detected and correctly recognized. . .	157
6.5	Distance matrix for the second-stage correspondence	157
7.1	Geometric shape measurements made for the MURI target chips. . .	163
7.2	Relative contrast and dynamic range of the targets in the MURI FLIR data set.	168
8.1	Linear classifier performance in combining centroid uncertainty and matched filter for FLIR ATR	183
8.2	Best performance of the algorithm combining the centroid uncertainty and matched filter (comb-cen-mf) on the simulated FLIR test sets. . .	187
8.3	Best performance of centroid uncertainty based algorithm (cencovdi-ratr) on the simulated FLIR test sets.	188
8.4	Best performance of matched filter (tmatr) on the simulated FLIR test sets.	188
8.5	Best performance of the Maryland algorithm (matcher) on the simulated FLIR test sets.	188

8.6	Best performance of the maximum likelihood ratio algorithm (mlratr) on the simulated FLIR test sets.	189
8.7	Best performance of the centroid uncertainty algorithm with 1-D cubic spline model for boundary uncertainty estimation (polyspatr) on the simulated FLIR test sets.	189
8.8	Sensitivity of algorithm performance to image quality.	189
8.9	Recognition performance (confusion matrix) for the proposed algorithm	195
8.10	Recognition performance (confusion matrix) for the Maryland algorithm	196
8.11	Results from Z -test for assessing the statistical significance of the difference in the the detection rates.	202
8.12	Some statistics for the VVFH data set.	211

ACKNOWLEDGMENTS

First I would like to thank my advisor Professor Robert M. Haralick for leading me through the process of learning to keep my mind flow and becoming an independent and disciplined thinker, which is what this Ph. D. education is all about. I am greatly inspired by his continued push for excellence, elegance and perfection. This five-year long endeavor of mine would not have been possible had I not received continuous financial support from his research projects. I also greatly appreciate his spending long hours in discussions of both technical and philosophical issues with me. This dissertation was made much better by his careful reading and commentary.

I want to thank my committee members for serving on my committee and commenting on the draft of my dissertation. I also want to thank the following people for kindly providing data and other materials that were useful to this research: Joseph Kitrosser of NVESD, Dr. Harry Schmitt and Mary Cassabaum of Raytheon, Dr. Richard Sims of MICOM, and Dr. Mark Stevens formerly with Colorado State University.

My research also benefited significantly from discussions with current and former members of the Intelligent Systems Laboratory, in particular, Selim Aksoy, Lixin Gong, Ming Ye, Mingzhou Song, Desika Nadadur, Dr. Lei Sui, Dr. Jisheng Liang, Dr. Qiang Ji and Dr. Visvanathan Ramesh. I also appreciate the advice on work, research and personal life from Dr. Su Chen. I thank Professor Terry Rockafellar of the Mathematics Department for answering some optimization related questions that were important to my research, especially the ones related to the optimal matching problem. I received editorial suggestions from Mr. Clay Cooper for improving the

quality of this dissertation.

I received help and friendship that were so important to me from a number of families during my five-year stay in Seattle. Lixin Gong and Tianshu Wang gave me all kinds of help and support and became my great friends. Hao Wu and Maggie Zeng made my life so much easier and more colorful. I enjoyed the good time spent with the Liang family of Boulder, CO, and appreciate their making me realize how much work I needed to put into this daunting task of finishing the doctoral education. I thank Roy and Gabrielle Raudebaugh, Jr. of Lynnwood, WA for hosting me when I first arrived in the Seattle area, and for becoming my good friends ever since. Through them, I got to know Kurt and Brigitte Heinrich of Mukilteo, WA and enjoyed their warm hospitality and genuine friendship. I also wish to thank my good neighbors and friends, Jon van Hise in E and Ed Smith in F, for their help and friendship.

Last, but never the least, I thank my parents and brothers for their love, support, understanding and patience.

DEDICATION

To my father and mother.

Chapter 1

INTRODUCTION

1.1 Computer vision research

One of the major objectives of the engineering disciplines is to find new ways of doing things more efficiently. All forms of automation have been among the direct driving forces behind the advancement of human productivity. The digital computer has played a crucial role in the implementation of these automation systems. In order to function well, these systems need sensor subsystems to gather relevant information about the processes which are being automated.

For many automation systems, the overall performance depends vitally on the performance of the sensor subsystems. A few examples of such systems would be various medical image analysis systems, industrial production assembly and inspection systems, satellite image analysis systems in remote sensing, and automatic target detection and recognition systems in the defense industry. These systems all share one common feature, which is that their sensor subsystems all try to extract information from some input data captured by some kind of camera. Such data is very similar to the image formed on the retina of the human eye when that human is looking at some scene. The desired function of the sensor subsystems is to mimic the capability of the human vision system, i.e., to understand the content of the scene as represented by the input imagery, or to make certain measurements and/or judgments on it. For this reason, such sensor subsystems are often referred to as computer vision or image analysis/understanding systems.

Although it appears to be easy for the human vision system to do such an excellent job in understanding a scene, it has been an extremely difficult task to develop computer vision systems whose performance can approach that of the human vision system. There are many reasons for this difficulty. Some are due more or less to the deep understanding we lack of the human vision system. Answers to many basic questions are still being actively sought. A few of these questions are: What are the measurements to make on the imagery that are most relevant to its understanding? How to utilize partial information to infer information that is not readily available from the imagery? What is the proper way to iterate the processes of collecting local evidence and inferring global configuration, and let them benefit from each other? Besides the difficulties in finding the answers to such questions, the computing power of today's most advanced digital computer is still no match of that of the human brain.

This dissertation presents a theory that attempts to answer some of the basic questions found in computer vision research. Specifically, we propose the use of a location uncertainty measure for model-based detection and recognition applications. The location uncertainty measure carries a source of information that has not been utilized by existing vision systems. Experiments confirmed the usefulness of such information. The theory is presented in the context of military automatic target recognition applications. However, the theory itself is general and can be applied to other applications, such as the medical image analysis applications and the industrial inspection applications.

1.2 The FLIR ATR application

Automatic target recognition (ATR) is a very important application of the computer vision/image understanding research in the defense industry. It concerns the development of automatic procedures that can detect and recognize targets of interest from

some sensory input without human intervention. Among others, the forward-looking infrared (FLIR) is an important modality commonly used in ATR applications. Other commonly used modalities include visible light (TV), laser radar (LADAR), and synthetic aperture radar (SAR) [125, 11, 13]. Due to its low cost, passive work mode, high angular resolution, day/night operation capability, and long atmospheric range (10 ~ 15 km), FLIR is widely used in ATR applications. It is often used for ground vehicle and airborne target detection and recognition. Response of the 8–12 micron FLIR sensor to these targets is generally distinct from the response of the sensor to the natural background [76, 11].

Due to the nature of the ATR application, the sensory images are usually taken in uncontrolled environments where the conditions of the image forming process may vary greatly. Images may be taken of outdoor scenes with any variety of targets on complicated terrain with significant amounts of clutter, under all kinds of weather conditions, in all seasons and under various lighting conditions, with occlusion and camouflage of the targets. This has made ATR a distinctive application where the input image is characterized by extremely high variability in content and quality. For the same reason, ATR has been one of the most difficult areas in computer vision research.

With the advancement of the imaging technology, the FLIR sensor has evolved from the first generation to the third generation. Earlier ATR algorithms through the mid-1980's consider the data from the first generation FLIR sensor, where lower spatial resolution make targets appear as bright blobs in the image. Contrast-based algorithms are used for detecting such blobs and segmenting them from the rest of the image. Features on the geometric shape and on the gray scale values of these segmented regions are extracted and used by statistical pattern recognition methods for recognition of the targets. Aside from the low quality of the image, the lower signal-to-noise ratio and reduced number of pixels on the target regions make the FLIR ATR task extremely difficult. The proposed algorithms, many with *ad hoc*



(a) the original image

(b) with target boundary outlined

Figure 1.1: Example of a second generation FLIR image of a ground vehicle.

selection of features on the segmented regions, did not yield acceptable performance [125].

The second generation FLIR sensor renders much better images with higher resolution, higher signal-to-noise ratio, and better stability. Figure 1.1 shows such an image from the DARPA URI data set. The better image quality, especially the increased number of pixels on targets, reveals more structure of the target appearance in the image, and allows the design of knowledge-based and model-based algorithms. The situation is further improved by the dramatically improved computing power which supports sophisticated algorithms.

It should be noted that it is mainly the change in the basic approach in the algorithm design, instead of just the improvement in the image quality, that caused enhanced performance [125]. Further improvement in algorithm performance should not be expected to come mainly from further improved image quality. In fact, the second generation FLIR sensor has approached the optical limits of sensor resolution, and further improvement in the sensor quality by itself is not likely to greatly improve

the ATR performance [125]. Besides the use of multi-sensor data fusion, the theory of which is yet to be developed, model-based approaches to the ATR problem are the most promising in developing high performance systems.

1.2.1 Difficulties

Most difficulties in the FLIR ATR problem come from the extremely high variability in the scene content and appearance. Because of this, targets of interest take on very different appearance in different situations. Their inconsistent appearance makes characterizing targets in the imagery very difficult. The problem is made more complicated by the presence of clutter that resemble targets, and by the low signal-to-noise ratio of the imagery. This is beyond the capacity of traditional 3D object recognition based approaches, which depend heavily on fairly good low-level feature extraction results.

The ability of low-level feature extraction algorithms is restricted by their limited information input from the local image neighborhood. High-level information can be very useful in exceeding this limitation. This is typically done in the model-based approaches. Either the target models are directly used in the low-level information extraction modules, or hard decisions about the information content in the imagery are avoided on low-level measurements until they are combined, via the use of target models, into medium- or high-level evidence. In doing this, weak but consistent evidence in low quality imagery can be reliably recovered.

In this dissertation, we take the model-based approach to the FLIR ATR problem. We make improvement both in the low-level feature extraction process and in the accumulation of low-level evidence.

1.2.2 Target models for FLIR ATR

To recognize targets of interest, features that are invariant to contrast, rotation, scale and all kinds of imaging conditions have to be identified. Targets of interest are then

modeled by these features. A universal model good enough for the detection and recognition task for all conditions, even for a single target, has proved impractical. Conditions of specific applications have to be considered, and assumptions made, in order for a model to be useful enough for the ATR application.

Different assumptions on the conditions lead to dramatically different models for the target models. In some applications [79, 31, 49], the thermal conditions of the target and the surrounding area change so much from scene to scene that virtually no assumption can be reliably made on the appearance of the structure inside the target boundary. In this case, only the 2D geometric shape of the target silhouette can be used for target detection and recognition. The underlying assumption is that there are some differences in some properties between the areas inside and outside the boundary of the target silhouette. These differences give the rise to the possibility of using the target silhouette for detection and recognition.

This dissertation is also concerned with this type of most adverse application conditions where only few assumptions on the image forming process can be reliably made. Only reliable information obtained with the help of these reliable assumptions, although small in amount, is used. Other sources of information are ignored due to their unreliability. The silhouettes of the targets in the 2D images are used to model the targets, which are used in the accumulation of weak but consistent contrast evidence along the target boundary.

In some other applications where the appearance of the targets from scene to scene is fairly consistent, features of the regions inside the target area may be used in the region-based models. One way this can be accomplished is to use mathematical transforms, e.g., the wavelet transform [158], to detect feature points. These feature points form the invariant features to model the targets. Here more assumptions are made than in the modeling using only the boundary. When the assumptions are satisfied by the application, knowing that they are true gives ATR algorithms some highly desired capabilities such as handling targets with different aspects and scales.

However, when a significant portion of the assumptions do not hold true, algorithms based on the model will fail miserably.

3D CAD models for targets in the ATR application have also been used [31, 147]. Simulation of the image forming process, in conjunction with computer graphics technique, can be used to predict the presence of features and help in their detection. These models are used in the most benign situations where the image forming process is more predictable.

Since the usefulness of multi-sensor data fusion has been demonstrated in ATR applications, efforts have been made at modeling targets using their signatures from more than one sensor [2, 31, 147]. With proper architecture, the information gained from one modality can be used to help infer information from other modalities. Active research is being undertaken to gain a deeper understanding of this topic.

1.3 Overview of previous work

As in many other areas of the computer vision / image understanding research, there is a very large number of published works in the ATR applications area. For new ATR algorithm development, the works can be roughly divided into two categories, i.e., the traditional approach and the newer model-based approach. In this section, we give a brief review of these approaches and related issues. A more detailed literature review is given in Chapter 2.

1.3.1 Traditional approach

The traditional algorithms for the ATR problem can largely be summarized by a four-stage procedure consisting of detection, segmentation, feature extraction, and classification. The detection stage delineates certain areas of the input image to constitute the regions of interest (ROI), which potentially contain some targets. Significant contrast is assumed to exist between targets and their immediate background

regions. Algorithms for finding high contrast such as the double-window operator are carried out to find blob-like regions comparable to the targets of interest [11, 138, 120]. These detected areas are then segmented into target regions and background regions. Simple thresholding, spoke filter [31], edge-based, region growing [81], and relaxation-based algorithms [42, 12] have been used for this task. The segmented target regions are then treated in the feature extraction stage and are represented by the extracted feature vectors. The extracted features mainly describe the 2D geometric shape of the segmented region, which include the moment invariants [64, 35], the Fourier descriptor [118] of the boundary, and some other methods describing the shape [47, 152, 139]. Some kind of classifier is finally used to determine the class of the feature vectors and give the final recognition result.

The traditional approach is actually very ambitious in trying to build an almost universal ATR system. It places very few restrictions on the application and does not need to know much about the target geometry until the late stage of classification. The difference between targets of interest is only reflected by the different representative feature vectors in the high dimensional feature space. The entire procedure is almost universal for many different kinds of targets.

It would be great if this approach would work well. However, due to the poor quality of the input imagery, the earlier stages of detection and segmentation often fail to produce a satisfactory result. Failures come in the form of misdetections and false alarms, often because only parts of the targets get segmented, or because background clutter gets segmented as part of the targets. These incorrectly segmented regions will in most cases cause errors in the final detection and recognition performance.

1.3.2 Model-based approach

Aware of these pitfalls, researchers have come to realize that the great complexity in the scene content and the extremely high level of variability in the image forming process require a more application-specific approach to the ATR problem. This ap-

proach needs to give a more detailed characterization of the target, background, and clutter. In other words, modeling of the target, background, and clutter has become the most fundamental issue to address. Having appropriate models in the early stages of processing can be of significant help in the information recovery process.

However, it has proved to be a difficult task to find a systematic and effective way for building precise mathematical models for the target, background, and clutter. Since almost all targets of interest are man-made objects, their modeling is relatively easier. The 3D geometry and the properties of the material are among the most important known factors affecting their appearance in the imagery. Hence, they are most commonly used in target modeling. Depending on the different levels of stability of target appearance in different applications, models of different complexity have been proposed [79, 158, 147, 155]. The use of these models in ATR systems has resulted in improved target detection and recognition performance, albeit at the cost of more computation and sometimes certain restrictions on the application, e.g., limited number of known target classes [155].

Correlation-based techniques are among the first model-based techniques. These algorithms assume that some features, which are invariant to scale, target pose, lighting, and certain other conditions, of the target can be reliably extracted from the 2D appearance. These features, most often the coefficients from some mathematical transform of the pixel intensity values [154, 106, 158, 40, 9], are used to model the targets. This approach to modeling the target is one of the first proposed for ATR, and is still being used in many algorithms and applications, such as the Gabor grid model used in [158], where the Gabor wavelet transform is used to extract feature points to model the target.

The requirement on the invariance in target appearance can be loosened in certain situations where it is possible to predict target appearance with relative reliability. These situations are more or less characterized by simple scene content and benign imaging conditions. Complex models using the 3D CAD model of the target and the

heat emission and reflection properties of the material have been used [147]. At the other end of the spectrum, where the 2D appearance of the target within its boundary is highly unpredictable, the aspect of the targets are assumed to be known, and the silhouette of the target is used as the model [79, 32, 49].

1.3.3 Input image and clutter characterization

Characterizing the input image for its information content and complexity is of fundamental importance in ATR research [125, 11, 26]. The busyness [157], the average variance [140], the probability of an edge (POE) metric [135, 20], texture [143], fractal processes and Markov random fields [135], transform based features [132], and information-theoretic measures have been proposed to characterize the overall information content as well as the clutter level of the input image. The insight they provide into this issue has been limited. To our best knowledge, there has been no significant reports of their successful use in guiding the design of ATR algorithms. Even after more than a decade of research, the term clutter still lacks a precise definition. The vague expression of “non-target objects looking like targets of interest” is commonly used and reluctantly accepted in the literature. Due to the dependence of the “definition” of clutter on the targets, it is believed that a meaningful definition and measurement of clutter is an application-specific issue and is closely related to the modeling of the targets of interest [140, 26, 135].

1.3.4 Neural networks in ATR

As in all other fields of signal processing and pattern recognition applications, neural networks have received a significant amount of attention in ATR algorithm development. Promising results have been reported in the literature. Roth [134] and Rogers [131] give surveys of the research efforts in this area. This is still a very active area of research. A few examples of recent publications in this area are Ernisse *et al* [39], Rong and Bhanu [132], and Gader *et al* [44].

One of the most important advantages of the neural networks is that very powerful classifiers can be built from some universal, simple structures. Through a carefully designed training process, neural networks can be made to learn very complicated data that is difficult to model using traditional approaches. However, the disadvantage of the neural networks is also associated with the training process. Due to the lack of deep understanding, the design of a successful training procedure is more or less an art that requires very much experience as well as trial-and-error.

In almost all of the efforts, neural networks are mainly used as non-linear classifiers, or as tools for non-parametric modeling of data. For this reason, at an abstract level neural networks in ATR algorithms can largely be considered as a way of implementation, rather than an entirely different approach to the problem.

1.3.5 Multi-sensor data fusion in ATR

Multi-sensor data fusion [2, 109, 11, 147, 31] is about getting the sensory inputs from independent sensors for the same scene and letting them complement each other in the information recovery process. The commonly used sensory modalities for fusion in the ATR applications include FLIR, TV, LADAR, SAR, and MMW radar. In addition to the data from imaging sensors, various kinds of auxiliary information, such as the time of the day, weather conditions, and geographical location, are also subject to fusion.

Currently, the theory of multi-sensor data fusion is still very much under-developed. Its usefulness in improving ATR algorithm performance mainly comes from the intuition that “it should work better” and from the observed performance improvement of implemented systems on test data. Further study into this area is needed before a theory is established which can indicate how to optimally perform the fusion, and can predict the amount of performance improvement due to the use of fusion.

1.3.6 Performance characterization in ATR

Performance characterization of ATR algorithms addresses the issues of predicting the performance of the algorithms on specified data, and also of comparing the performance of different algorithms. There are mainly two approaches to computer vision algorithm performance characterization: the theoretical approach and the empirical approach.

In the theoretical approach, Haralick [55, 56] suggests studying the effect of imperfection and perturbation of the ideal input on the output of the algorithm. Analytical error propagation [160, 57] through the computer vision algorithm modules is the main tool used in this approach. Although a number of standard algorithm modules have been studied [123, 122], the performance evaluation of a general computer vision algorithm sequence in real imagery is still not practical.

In the empirical approach, judgments are made on the quality of the algorithm output. Objective and quantitative judgments are preferred over subjective and qualitative judgments whenever possible. The objective performance measures are mostly obtained by comparing the algorithm output with the ground-truth or reference information for the data on which the algorithm is applied. This approach brings up the need for large data sets replete with ground-truth. However, such data sets are often extremely expensive to prepare. As a compromise, simulated data sets are often used in studies.

1.4 Motivations for this study

From earlier discussion of FLIR ATR application, we see that the extremely high complexity of the scene content and the image forming process result in highly variable target appearance in very low quality imagery. Various research results [79, 49, 50, 32, 31, 30] suggest that the most reliable visual cue for the targets in this kind of imagery is the existence of some contrast between the inside and outside of the silhouettes of

the targets. This is reliable because of the difference in the materials, hence physical properties, between the targets and the natural background. Although the actual appearance of these materials in the FLIR imagery may vary greatly in different imaging conditions, noticeable difference should exist between them.

In this situation, a robust FLIR ATR algorithm should try to infer target existence by accumulating the evidence of contrast along the target boundary. This introduces three immediate problems to be solved. The first problem is that targets are 3D objects and will have many distinctly different 2D silhouettes due to its relative pose to the imaging sensor. Practically, although the number of silhouettes of interest is finite, the amount of computation for processing all silhouettes can be very heavy even for one target. Some decision tree type algorithm [79] can be used to reduce the amount of computation by considering only a subset of silhouettes at each level. The computational load then seems to be acceptable to the US military research labs [79].

There are two other problems in designing boundary contrast based FLIR ATR algorithms. How will the contrast along the boundary be combined into a single evidence measure for the target's existence, and how will the contrast at each boundary point be estimated and represented to facilitate the evidence accumulation? To answer these two questions, we study the polygon centroid location uncertainty, image intensity gradient estimation, and edgel location uncertainty.

An important issue in computer vision algorithm development is how to characterize algorithm performance [34, 54, 53, 122, 22, 110, 161, 136, 76, 73, 144]. We use quantitative objective measures that result from comparing algorithm output with the ground-truth. Data sets with ground-truth are needed in this process. Furthermore, to investigate the algorithm behavior for different levels of scene complexity and image quality, multiple sets of such data with ground-truth are required. The cost is prohibitive in creating such data sets from real imagery. As a result, the use of simulated or synthetic data is necessary [162, 31, 117, 111, 27, 159, 82]. We also create simulated scenes in our experiments.

In computing the performance measures, some correspondence between the ground-truth and the targets declared by the algorithm need to be established. The establishment of such correspondence is not a trivial problem. We introduce a general methodology to solve this problem.

1.4.1 Motivation for studying polygon centroid location uncertainty

Intuitively, some form of weighted average seems to be appropriate for the combination of boundary contrast into a single target evidence measure. The higher the contrast along the boundary, the more likely the target is present. The reason for a weighted average as opposed to a simple average, such as the arithmetic or geometric average, is that generally not all boundary points are equally informative for the existence of the target. Also, when only a subset of the boundary points is used to represent the target, different subsets of the same number of boundary points carry different amount of evidence for discriminating this target from other types of targets and from the background clutter. This is because the contribution of the boundary points to the detection of the target significantly depend on each other. The relative importance of a particular boundary point should be decided by the 2D shape of the entire target boundary.

We need to find a way of assigning relative importance for target boundary points that both “makes sense” and is mathematically sound. In doing this, we need a geometric relationship that relates some overall property of the entire target boundary to the some property at individual locations along the target boundary. The only geometric properties that a point on the target boundary possesses are its location and the uncertainty of that location due to some perturbation. Two kinds of closely related and highly useful relationships exist between the overall target boundary and the individual points on that boundary: one for the boundary locations and the other for the uncertainty of the boundary locations. Both the boundary locations and the uncertainty in the boundary locations will be of direct use to us.

It is an accepted principle in signal and image processing that prominent features for any detection problem should both have high signal-to-noise ratio (SNR) and be well localized in its domain. This is also true for the target boundary points. The boundary points of high-contrast targets in the FLIR imagery can be localized with high precision, while high uncertainty of the boundary point location estimates results from low-contrast points which weaken the evidence of the target's presence.

When the locations of the computed boundary points can be determined with an associated uncertainty, a good measure of the overall uncertainty of a hypothesized target is the location uncertainty of the centroid of the target. When the majority of boundary points have high location uncertainty due to their low contrast, the centroid uncertainty is also high, indicating low evidence for the target's presence. On the other hand, when the contrast at many boundary points increases significantly, the decrease in the location uncertainty at these places would normally cause the location uncertainty of the centroid to decrease noticeably, showing stronger evidence for the target's presence. Therefore, using the uncertainty of the centroid is a good way of combining the boundary information into a single evidence measure of the target's presence.

We use polygons to model target shapes, where the polygon vertices are specified by the target boundary points. The centroid location is a function of the vertex locations. In Chapter 3, we derive the formulae for computing the centroid location uncertainty measure as a function of the vertex location uncertainty.

1.4.2 Motivation for studying gradient estimation and edgel location

The location uncertainty of the boundary points can be used to compute the centroid uncertainty as the evidence measure of the target's presence. However, the location uncertainty of the boundary points cannot be measured directly, but is closely related to the contrast and noise level. The local contrast is the difference in the intensity value between the inside and outside of the target across a local segment of the

target boundary. In order to use boundary point location uncertainty, the relationship between the signal-to-noise ratio (SNR) — defined as the ratio of contrast divided by the standard deviation of the noise — and boundary point location uncertainty needs to be worked out.

The quality of the contrast estimation has a very significant effect on the overall ATR algorithm performance. Since contrast and gradient estimates are very closely related and since there have been extensive studies on gradient estimation in the literature, we will make use of some of their results to compute high-quality contrast estimates. This is made possible by using a step edge pattern to model the intensity change across the local segment of the target. By any chosen gradient estimation scheme, there is a one-to-one relationship between the contrast and the image gradient.

We need a gradient operator that has low bias and high output SNR. There are numerous gradient operators, e.g., the Sobel and Prewitt operators [60, 67], the integrated directional derivative operator [163, 60], and the derivative of Gaussian operator [19]. The performance of these operators differs quite noticeably under different situations. There have been many studies [96, 163, 98, 28, 45] that try to characterize the performance of such operators. These studies include both analytical efforts and empirical comparisons of operators. However, they fail to give a precise explanation of why some operators perform better than others.

In Chapter 4, we describe a general framework for studying a broad class of gradient operators. A decisive factor in the quality of gradient estimation is identified, which offers a precise explanation of why certain gradient operators perform better than others. Based on this result, the tuning parameters for gradient operators can be optimally chosen. As an example, the optimal integration domain size is found for the integrated directional derivative operator. The resulting operator is used for gradient and contrast estimation in this dissertation.

To relate the SNR with the location uncertainty of the boundary points, we fol-

low the widely accepted practice of defining the edgel location as the zero-crossing of the second directional derivative in the direction of the gradient. Unlike in the edge detection applications, where only the negatively sloped zero-crossings are of interest, the sign of the slope is considered unimportant for contrast evidence accumulation via the location uncertainty. What is important are the intrinsic errors, due to the noise in the image, in the zero-crossing location estimate and in the gradient orientation estimate. These errors contribute to the uncertainty of the edgel location. In Chapter 5 we derive precisely how this uncertainty is influenced by the SNR.

1.4.3 Motivation for studying the constrained matching problem

In performance characterization of ATR algorithms, the numbers of detected targets, correctly recognized targets, misdetected targets and false alarm declarations need to be computed by comparing the algorithm output against the ground-truth. There is not yet a widely accepted way of doing this comparison. An accurate, consistent, and intuitive way of doing it is needed to give accurate and consistent performance characterization that closely follows the intuition.

In Chapter 6 we introduce a constrained matching problem to model the process of establishing correspondence between ground-truth and declared targets. The solution to the problem provides a natural maximal one-to-one correspondence. This enables a very reasonable and consistent performance evaluation of ATR algorithms.

1.5 Summary of major contributions

This dissertation contains original work in image gradient estimation, uncertainty in edgel location, model-based FLIR ATR algorithm, and correspondence problem in detection and recognition algorithm performance evaluation.

In particular, this dissertation contains

- the development of a general methodology for characterizing the performance

of gradient operators. This methodology provides a precise explanation of why some operators are superior than others. As an application of the methodology, the optimal integration domain size of the integrated directional derivative operator is analytically derived;

- the introduction of a polygon centroid uncertainty measure for combining target boundary contrast evidence in FLIR ATR. The relationship between edgel SNR and location uncertainty is obtained. Formulae are derived for computing the centroid location uncertainty from the boundary point location uncertainty. Experiments have been conducted in verifying the effectiveness of this feature for target detection and recognition, and in combining this feature with traditional features to further improve algorithm performance;
- the development of a centroid uncertainty based FLIR ATR algorithm with significantly improved detection and recognition performance over competing algorithms. For the test data set with medium target contrast and appearance deviation (average contrast 1.8, target appearance composition with 50% texture), the performance at the optimal operating point is: detection rate 67% with 5 false alarms per frame. (The optimal operating point is chosen by the minimum cost criterion, in which the cost of a misdetection is twice the cost of a false alarm. Therefore, in general, the optimal operating points of different algorithms are at different detection and false alarm levels. See Section 8.1.3 for details on the criterion.) The performance of the best competing algorithm is: detection rate 54% with 3.3 false alarms per frame. The centroid uncertainty is then successfully combined with the traditional matched filter technique which results in further improved performance: detection rate 81% with 4.2 false alarms per frame;
- the introduction of the constrained matching problem to model the correspon-

dence process in FLIR ATR algorithm performance evaluation. By transforming the problem into an assignment problem, the efficient Hungarian algorithm is used to find a maximal one-to-one correspondence between ground-truth and declared targets. Performance characterization based on this procedure is precise, consistent and follows the intuition. This model is readily applicable to many other detection and recognition applications; and

- the procedure for generating simulated FLIR scenes with controlled target contrast, dynamic range, and amount of deviation in the target appearance. This produces test data sets with ground-truth with extremely low cost, and hence allows easy study of algorithm behavior in response to the change in those factors.

1.6 Organization of the dissertation

This dissertation is organized as follows. A literature review on ATR and the related issues is given in Chapter 2. Chapter 3 presents the development of the polygon vertex-to-centroid covariance propagation for combining target boundary contrast. Chapter 4 studies the gradient estimation problem and explains why some operators are better than others and how gradient operators can be optimized. Chapter 5 establishes the relationship between edgel SNR and location uncertainty. Chapter 6 discusses the correspondence problem in detection and recognition algorithm performance evaluation. A constrained matching problem is introduced, transformed, and finally solved using the Hungarian algorithm. Chapter 7 describes a procedure for generating simulated FLIR scenes with controlled difficulty levels for target detection and recognition. The experiments are described in Chapter 8 where competing algorithms are briefly described along with the observed performance of the algorithms on test data sets of different difficulty levels. The dissertation concludes in Chapter 9 wherein possible future research directions are identified in order to further improve

the algorithm performance.

The materials in Chapters 3 and 5 were published in [92]. Earlier versions of these materials also appeared in [90] and [88]. The materials in Chapter 6 was published in [91]. The materials in Chapter 4 is in preparation for publication. The aerial image ground-truth database used in one set of the experiments (Section 8.5) was announced to be freely available to the public in [94].

Chapter 2

LITERATURE REVIEW

Driven by the need for developing military applications, research for computer aided or automatic target recognition has been active for more than three decades. A casual literature search will reveal hundreds or even thousands of publications related to this topic. For computer aided manual target recognition, researchers study the target detection and recognition performance by human operators looking at images shown on the display of the sensor system. Among others, interesting and important topics are the modeling of the human performance and what can be done to the sensor system to improve that performance [5, 148, 20, 101, 102, 104].

More research publications are concerned with automatic target recognition (ATR) systems than the computer aided manual systems, although only the latter are practical for battlefield deployment at the present time. Many approaches and techniques are developed for detecting and recognizing targets in sensory images, such as visible light, infrared (IR), synthetic aperture radar (SAR), and laser radar (LADAR). Existing algorithms can be roughly divided into two general categories: traditional algorithms and model-based algorithms. The traditional algorithms use the bottom-up structure for collecting information leading to target detection and recognition. Information of specific potential target classes is not heavily used in the early stages of target detection and feature extraction. The model-based approaches try to make use of that information right in the early stages of processing. With the cost of more computation, and often times restrictions on the application, model-based approaches in general can offer much more reliable detection and feature extraction, which in turn helps improve recognition performance as well.

In the following sections, we briefly review some representative work in both categories. These are followed by a brief discussion of the artificial neural network techniques employed by ATR algorithms. Although performance evaluation acts as a supporting role in computer vision research, it is an immature and active area of research itself. We also review some of the research efforts on that topic.

2.1 *Traditional algorithms*

The traditional FLIR ATR algorithms assume that targets in the scene appear as blob-like regions with roughly homogeneous gray scale values on a roughly homogeneous contrasting background, either significantly brighter or darker [11, 138, 120]. For detection, the main effort is made to segment out these target regions from the rest of the scene. Since the contrast between target and background is not always very high, and grayscale values inside the target area are often not uniform, reliable segmentation of target regions is an extremely difficult task. The situation is made worse by the clutter in the images.

The segmented regions are used for target recognition. Techniques for recognizing geometric shapes are applied on the segmented regions. In some situations, the grayscale values in the segmented regions are also used. The main challenge here is to handle regions that are not well segmented.

Bhanu and Holben [12] try to obtain good segmentation. They propose to combine pixel intensity and edge strength in defining the inconsistency and ambiguity of edge pixel classification in the segmentation process. The so-called gradient relaxation procedure is used to minimize the inconsistency and ambiguity. In the experiment, the procedure is applied on 7500 FLIR images to segment out ship targets. Through a qualitative evaluation of the result, the authors claim the procedure to be very effective. However, it appears that, in order for the procedure to work well, appropriate setting of initial probability values and the compatibility function is critical.

In applying thresholding for image segmentation, Pham *et al* [119] use grayscale morphological operations to enhance the target appearance in the images. One-dimensional structuring elements of the target length and height are applied on the images for horizontal and vertical opening and closing operations. The processed images are properly combined and thresholded to give segmentation results. A generic rectangular shaped target is then used in template matching to perform the final detection. No recognition operation is involved. In the experiment on 147 images with a total of 630 targets, the algorithm performs slightly better than a double-window pixel classifier based segmenter.

Ernisse *et al* [39] enhance the first-generation FLIR imagery by the hit/miss filter. Objects of interest are identified by combining the results of a global and a local thresholding procedure applied to the enhanced image. Another set of objects are identified by thresholding the difference of Gaussian (DOG) filtered image. Parameters of the two procedures including the thresholds are determined by a genetic algorithm. Regions of interest are identified by comparing the two sets of objects. Their geometric measurements are used to form the feature vector to be used by a multi-layer perceptron (MLP) neural network for target recognition.

Kitrosser [76] evaluated a region-growing based algorithm for its detection and recognition performance. According to him, good performance of the algorithm depends heavily on the art of tuning the algorithm parameters. Markham [99] gives a comparison of three classes of segmentation algorithms. Haralick and Shapiro [59] give a survey on classical segmentation techniques. Haddon and Boyce [48] study the problem of classifying segmented regions. The edge co-occurrence matrix is used for preliminary characterization of texture. The Hermite functions are used to decompose the co-occurrence matrix, and the important coefficients from the decomposition are used to form the feature vector. The MLP neural network was observed to perform well in the experiment with 98 grass regions and 55 tree regions from real IR video.

Wang, Der and Nasrabadi [156] consider the recognition of segmented target image

chips only. Two classifiers, namely the learning-vector quantization (LVQ) and modular neural network (MNN), operate on the features extracted for the image chips. A cascade architecture for classifier combination is proposed for good performance and relative higher efficiency. In the experiment, the fairly large-sized real image data sets SIG and ROI are used. 80% of the data is used for training and 20% is used for testing. The proposed cascade architecture performs only marginally less than the best performer, the stacked generalization, which is an MLP neural network that takes the outputs of the two classifier modules and gives the final classification result.

There are many other works that try to accomplish good segmentation and recognition, e.g., the references [81, 41, 152, 139]. Due to the poor quality of the input imagery, the earlier stages of detection and segmentation often fail to produce a satisfactory result. As a result, there is a strong need for significant improvement in general ATR performance.

2.2 Model-based algorithms

In the model-based approach, it is believed that having appropriate models in the early stages of processing can be of significant help in the information recovery process. The use of such models, however, can sometimes pose certain restrictions on the application since the assumptions of the models have to be satisfied. Nevertheless, the use of the models can often result in great improvement in the algorithm's ability to work with the low-quality imagery encountered in ATR applications. As a result, more and more research is conducted with the model-based approach. The best performance achieved by model-based algorithms is significantly superior to that achieved by traditional algorithms [125].

Herstein *et al* [63] describe a practical system developed by the Defense Systems and Electronics Group of Texas Instruments. The system considers detection and recognition of critical mobile targets, e.g., mobile missile launchers. For the ROI de-

tection, knowledge of approximate target size and grayscale statistics is assumed and used in a correlation-based grayscale corner detector. Diagonal corners are paired and corner pairs are merged for target hypothesis if they are properly located spatially. Segmentation of the target silhouette is by over-segmentation followed by pairwise region merging, which combines some edge and contrast scores. Four groups of features are computed for the extracted ROI: the silhouette region, the high-contrast regions (most likely for the engine and the exhaust system), the size and inclination angle of the missile if present, and the number, size and positions of the wheels. Three-dimensional target models are projected to the imaging plane with the aspect angle estimated from the height-to-width ratio of the segmented silhouette. The intensity values are computed from the Bartlett model [8]. The differences in the four groups of feature values between the image and the projected target are computed, whose probabilistic distributions are obtained from training data.

With assumptions on some statistical independence and some chosen forms for the prior probabilities, the posteriori probability for the target is obtained. It is used for target identification. To improve the reliability of the computation, temporal fusion is used to combine the results from a sequence of multiple images.

Although vehicle models are used in detection and recognition, the basic features upon which the system is built are still segmentation based, a process that is not model-based. As a consequence, the system performs well for high contrast imagery when segmentation quality is high, and poorly in low contrast situations. The use of models should go further into earlier feature extraction steps.

Friedland and Rosenfeld [42] use simulated annealing to optimize a cost function that is a linear combination of low and high level confidence of the shapes of interest. The low level component is model-free and tries to improve contour smoothness and edge sharpness. The high level computes the similarity of the contour to shapes of interest. The optimization procedure puts more emphasis on the low level component in the earlier stage and shifts over to the high level component when the confidence

in a match increases. This methodology seems to offer a good compromise between the computational load, which is often high for model-based approaches, and target detection and recognition performance.

Der [31, 32] uses the target silhouette to gather weak but consistent evidence of the contrast across the target boundary. A probe is used for estimating local contrast along four major orientations. (Eight probe values are computed to allow directed contrast along the four orientations.) The histograms of the probe values are obtained locally from the image, and are used to approximate the distribution of the probe values computed in the background region. These are used in an independent model for computing the background likelihood from the probe values computed along the hypothetical target boundary. The target likelihood is computed from an assumed uniform distribution for the probe values. The likelihood ratio test is used for target detection and recognition.

In order to improve the noise resistance, Der argues that the probe values need to be thresholded at appropriate levels. The binomial model is used to compute the false alarm probability associated with the chosen threshold. The final threshold is chosen to minimize the false alarm probability. Target detection and recognition are based on the estimate of the achieved minimum false alarm probability.

Correlation based algorithms assume that some invariant features can be reliably extracted from the 2D appearance of the targets of interest. Fazlollahi and Javidi [40] study the abstract problem. They assume known target shape and intensity structure, but with unknown contrast. This is used together with background and noise models to set up a hypothesis testing problem. The solution to the problem is a generalized likelihood ratio test. In this classical approach, the issues that yet need to be addressed are the validity and estimation of background and noise models.

Zhou and Gutschow [162] perform feature extraction using the Gabor functions. The resulting feature vector is correlated with that for the target template. Non-maxima suppression is used for ROI identification. The overall ATR performance is

claimed to be not very sensitive to the threshold setting due to the use of the non-maxima suppression. Two MLP neural networks are applied to the feature vector of Gabor coefficients for false alarm rejection and target recognition. Uenohara and Kanade [154] use the K-L decomposition for invariant feature extraction. The result is used for template construction. Ben-Arie and Rao [9] use nonorthogonal expansion for template matching. Wu and Bhanu [158] use the Gabor grid to represent targets. Feature points are extracted using the Gabor wavelet transform.

There are many favorable reports of correlation based algorithms. However, when confronted with large variation in the target appearance, the invariance features required by these algorithms cannot be extracted robustly or do not exist anymore. This causes significant performance degradation.

In the event of high quality imagery, complex 3D target models and the heat flow properties of the material are utilized for predicting the targets' appearance in the images. Stevens and Beveridge [147] combine bore-sight color, range and IR images in a target-pose optimization procedure. Target detection is accomplished using the color image only via some color look-up table. Target recognition and pose estimation are accomplished at the same time and use all three images. Specifically, the IR image is used only in regard of the target boundary.

Another example of using the predicted target signature in target recognition is the system described by Herstein *et al* [63], which has been discussed earlier in this section.

At the other end of the image quality spectrum, the 2D appearance of the target is assumed to be highly unpredictable. Various research results [79, 49, 50, 32, 31, 30] suggest that the most reliable visual cue for the targets in this kind of imagery is the existence of relatively high contrast between the inside and outside of the boundary of a target. This contrast is in general higher than the contrast to be expected in the background. This is reliable because of the difference in the materials, hence physical properties, between the targets and the natural background. Although the actual ap-

pearance of these materials in the FLIR imagery may vary greatly in different imaging conditions, noticeable difference should exist between them within each image. To make use of this, the aspect of and approximate distance to the targets are assumed to be known. (This may pose serious restrictions on the applications. However, for the most severe imaging conditions being considered, this assumption helps greatly in the collection of weak evidence.) Targets are modeled by their silhouettes [79, 32, 49]. This modeling takes the view point that only reliable information obtained by means of reliable assumptions, although small in amount, should be used. Other sources of information are ignored due to their unreliability.

2.3 Neural networks in ATR

As in all other fields of signal processing and pattern recognition applications, neural networks receive a significant amount of attention in ATR algorithm development. It is particularly popular among the military research community. Mostly, neural networks are used as classifiers either for target detection or for target recognition.

Hamilton and Kipp [49] feed eleven features into an MLP network to perform preliminary target detection. Ernisse *et al* [39] and Franques and Kerr [41] use MLP networks for target recognition. Gader *et al* use the shared weight networks for vehicle detection, where the convolutional mask is obtained through training, as opposed to the pre-specified mask in traditional correlation methods. de Ridder *et al* [29] also use shared weight networks for vehicle detection. Instead of trying to train the network to detect the vehicle (tank) as a whole, the network is trained to detect the wheels. The radial basis functions arrangement in the network does not seem to offer much improvement in the performance.

Rong and Bhanu [132] use a Kohonen self-organizing map to learn the feature distribution from training samples for feature grouping. Li *et al* [84] report favorable recognition performance of a six-layer convolutional neural network in comparison

with a number of other ATR algorithms on a same large-sized test data set. Roth [134] and Rogers [131] give surveys of the research efforts in neural nets in ATR. A more recent survey by Amoozegar [3] reviews neural-network-based tracking algorithms since 1986. Neural networks in ATR is still a very active area of research [39, 132, 44, 162, 111, 48, 156, 23, 29, 151].

Popular as it is, the neural networks can mainly be regarded as non-linear classifiers or as tools for non-parametric modeling of data. For this reason, on the abstract level, neural networks in ATR can largely be considered as a way of implementation, rather than an entirely different approach to the problem.

2.4 Multi-sensor data fusion in ATR

Multi-sensor data fusion [2, 109, 11, 147, 31] combines the sensory inputs from independent sensors for the same scene and lets them complement each other in the information recovery process. The commonly used sensory modalities for fusion in ATR applications include FLIR, TV, LADAR, SAR, and MMW radar. Nandhakumar and Aggarwal [108] consider thermal physical properties of materials under direct sun light. The information in thermal and visual images is combined for material identification, which can then be used in target recognition. Specifically, the visual image is used to estimate the reflectivity and relative orientation of the surface of the object. The surface temperature is estimated from the thermal image. These estimates are used together in estimating the heat fluxes. A feature based on the heat fluxes is computed to characterize materials with different thermal properties.

Stevens *et al* [147] present results on applying a 3D object recognition algorithm with pose estimation on the Fort Carson data set [10] of bore-sight imagery from three sensor modalities: range, thermal and color. Target detection is done using only the color imagery. Data from all three modalities are used in target pose estimation. Specifically, the thermal data is used only with respect to the boundary of the target

silhouette.

Der [31] combined FLIR and LADAR imagery to improve ATR performance. A first-order Markov model is used for the list of so-called probe values in the two coregistered sensor images. The model parameters are estimated from training data. These are used in computing the joint likelihood ratio, which is used for target detection and recognition. The number of false alarms is reduced from about 1 per frame to about 0.2 per frame for the detection probability of 0.9.

In what Schutte [141] claims the fusion of IR imagery with range data, no dense range image is available. Instead, only the rough distance to the target is available at coarse temporal intervals. The rough distance is simply used for choosing the appropriate scale for target detection and tracking.

Dutkiewicz *et al* [36] describe the plan and some details of using satellite or airborne sensors for ship detection and recognition. They consider visible light, infrared, and radar (SAR) sensors and propose three levels of fusion: pixel level, symbolic level, and decision level. For image coregistration, geometric shapes of prominent ground entities (e.g., lake, coastal line) are used, which requires SAR image segmentation. Different channels of the hyper-spectral optical imagery in the visible-light spectrum are also found useful in identifying different aspects of ship targets.

In general, not only the data from imaging sensors are used in data fusion, various kinds of auxiliary information, such as the time of the day, weather conditions, and geographical location, are also subject to fusion.

Currently, the theory of multi-sensor data fusion is still well under-developed. Further study into this area is needed before a theory is established. The theory needs to indicate how to optimally perform the fusion, and to predict the amount of performance improvement due to the use of fusion.

2.5 *Performance characterization in ATR*

Haralick [55, 56] suggests an approach that studies how the imperfection and perturbation in the ideal input will affect the output of the algorithm. Analytical error propagation [160, 57] through computer vision algorithm modules is the main tool used in this approach. The main pieces of the procedure include the specification of the ideal input and ideal output, as well as the appropriate perturbation models for the input and output. Linear approximation of the perturbation is most often used. Although a number of standard algorithm modules have been studied [123, 122], the performance evaluation of a general computer vision algorithm sequence in general real applications is still not practical. One of the important reasons for this is the difficulty in finding the systematic and practical way of mathematically specifying the ideal input, which leads us back to the issue of modeling targets, background, and clutter in real images. Without a quantitative characterization of the input imagery, it is meaningless to talk about quantitative prediction of the algorithm performance. The other major difficulty with this approach is the non-linearity of the algorithm sequences and the statistical dependence between the input data. These make the analytical study of the algorithm behavior very difficult. Due to these reasons, this approach has not yet been widely used in ATR research.

Currently, most researchers report the performance of algorithms using the experimental result on some test data set [147, 84, 23, 76, 155, 119, 27], in terms of some performance measures such as the detection rate, false alarm rate, and correct classification rate, or some variations of these. Although the notion of these objective and quantitative measures are widely used, their exact definitions and the ways their values are computed, have not been standardized. To make things worse, the precise definitions of these terms being used in the reported experimental results are often not given. This makes it impossible to have a clear judgment of performance of the reported algorithm relative to other algorithms. In Chapter 6, we describe a general

methodology for the definition and computation of the performance measures. Performance measures based on this correspondence are accurate and consistent. We hope it will be adopted by the research community as a standard methodology for comparing algorithm performance.

Right now, the amount of real data with ground-truth is still quite limited. Deterred by the difficulty and often prohibitive cost in preparing ground-truth for large-sized real image data sets, researchers often make use of simulated imagery for evaluating ATR algorithm performance. For example, the terrain board imagery is used by McManamey [103], Der [31], and Peli *et al* [116]. Zhou and Gutschow [162] also use the terrain board imagery, where 80% of the data is used in training and 20% is used in testing. Nolan and William [111] use the Georgia Tech thermal model for simulating the target signature, while Lanterman *et al* [82] study the target orientation estimation using the target signature generated by the PRISM software. Cyganski *et al* [27] use images generated by planting target signature chips into real clutter scenes. Boriotti and Fereydoun [14] and Mohd [105] also use simulated images in their experiments.

Chapter 3

POLYGON CENTROID LOCATION UNCERTAINTY

3.1 *Introduction*

In using boundary contrast for target detection and recognition in the FLIR imagery, we combine the contrast evidence along the target boundary into a single saliency measure for the target's presence. Intuitively, some form of weighted average seems to be appropriate for the combination of boundary contrast into a single target saliency measure. The higher contrast along the boundary, the more likely the target is present. The reason for a weighted average instead of a simple average, such as the arithmetic or geometric average, is that generally not all boundary points are equally informative for the existence of the target. Also, when only a subset of the boundary points is used to represent the target, different subsets of the same number of boundary points carry different amount of evidence for discriminating this target from other types of targets and from the background clutter. For example, a good polygon approximation algorithm can use a small number of points to fairly well represent the boundary of compact a 2D shape, while randomly picked points will not do as well. The shape may be badly represented if the chosen points are clustered at unimportant places on the boundary, or along collinear or parallel lines. This is because the contribution of the boundary points to the detection of the target significantly depend on each other. The relative importance of a particular boundary point should be decided by the 2D shape of the entire target boundary. A mathematically sound way of modeling the relative importance of the boundary points needs to be worked out.

In studying the relationship between a polygon's centroid location and those of its

vertices, we realize that the relative contribution of the vertices to the location uncertainty of the centroid, which is considered as a measure of the geometric stability of the polygon, can be used to model the relative importance of the vertices. Since the location uncertainty of the target boundary points is closely related to the contrast across the target silhouette (see Chapter 5), the relative importance of the polygon vertices derived from the polygon centroid uncertainty model can be used, indirectly via the relationship between the boundary contrast and boundary point location uncertainty, as appropriate weights in combining the contrast at the boundary points into a single saliency measure for the target's presence. The centroid uncertainty is an inverted saliency measure for the target's presence — the more salient a target is, the smaller centroid uncertainty it has.

In this chapter, we present the model for polygon centroid uncertainty and work out and validate the formulae for computing the centroid uncertainty from target shapes and the location uncertainty of the target boundary points. The distribution of the resulting normalized centroid uncertainty is studied in a Monte Carlo experiment. For the normalized centroid uncertainty to be used as a feature for target detection, its detection performance can be predicted by a Monte Carlo method with a correlated contrast model for the target boundary.

3.2 Geometric relationship between a polygon's vertices and its centroid

Let the silhouette of the target of interest be represented by a simple polygon ¹. A simple polygon in the 2D plane is represented by an ordered list of its vertices. The vertices of an N -side polygon are denoted by P_1, \dots, P_N . Either clockwise or counter-clockwise ordering can be used. It does not affect the final result.

Let O denote the origin of the coordinate system, and let the coordinates of the N

¹A simple polygon is one whose sides do not cross each other. Within this dissertation, we are only interested in simple polygons.

vertices be $(x_1, y_1), \dots, (x_N, y_N)$. Let $\mathbf{x} = (x_1, \dots, x_N)^T$ and $\mathbf{y} = (y_1, \dots, y_N)^T$ be the vectors of the x and y coordinates of the vertices. Once \mathbf{x} and \mathbf{y} are given, the centroid of the polygon is fully determined and is denoted by $Q(\mathbf{x}, \mathbf{y}) = (Q_x(\mathbf{x}, \mathbf{y}), Q_y(\mathbf{x}, \mathbf{y}))^T$.

In finding the centroid of the polygon, we use triangles. A general case is illustrated in Figure 3.2. Consider connecting each vertex with the origin using straight line segments. These N line segments and the N sides of the polygon form N triangles $\{\Delta OP_i P_{i+1}, i = 1, \dots, N\}$ where the notation of $P_{N+1} = P_1$ is used. The centroid and the signed area of triangle $\Delta OP_i P_{i+1}$ are $(\frac{x_i + x_{i+1}}{3}, \frac{y_i + y_{i+1}}{3})$, and $\frac{x_i y_{i+1} - x_{i+1} y_i}{2}$, respectively. The centroid of the polygon is computed as the weighted sum of the centroids of the triangles, each weighted by its signed area, divided by the sum of the signed areas of the triangles. Hence, the centroid of the polygon is found to be

$$Q_x(\mathbf{x}, \mathbf{y}) = \frac{K_x}{3S} \quad Q_y(\mathbf{x}, \mathbf{y}) = \frac{K_y}{3S} \quad (3.1)$$

where

$$K_x = \sum_{i=1}^N (x_i y_{i+1} - x_{i+1} y_i) (x_i + x_{i+1}) \quad (3.2)$$

$$K_y = \sum_{i=1}^N (x_i y_{i+1} - x_{i+1} y_i) (y_i + y_{i+1}) \quad (3.3)$$

$$S = \sum_{i=1}^N (x_i y_{i+1} - x_{i+1} y_i) \quad (3.4)$$

Here we use the convention that $x_{N+1} = x_1$ and $y_{N+1} = y_1$. This computation is valid regardless of whether the polygon is convex or not, or whether the centroid falls within the boundary of the polygon or not, or whether the vertices are ordered clockwise or anti-clockwise.

3.3 Location uncertainty on the vertices

As will become clear in Chapter 5, the SNR at the target boundary is closely related to the location uncertainty of the boundary points. This location uncertainty is modeled

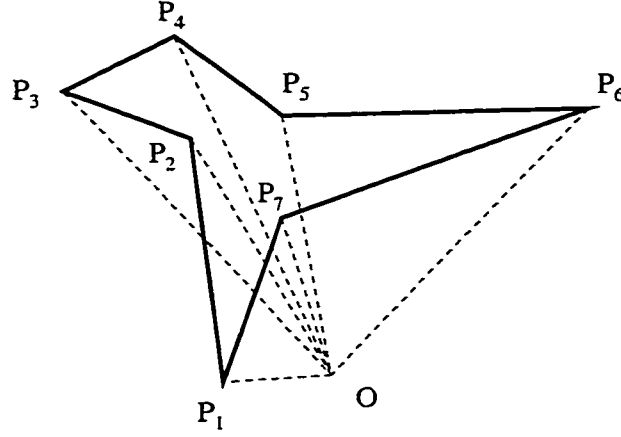


Figure 3.1: Illustration of the computation of polygon centroid.

here as a zero-mean additive perturbation on the polygon vertices. When the vertex P_i is affected by this perturbation, its perturbed location $\hat{P}_i = (\hat{x}_i, \hat{y}_i)^T$ is given by $\hat{P}_i = P_i + \Delta P_i$, or

$$\begin{pmatrix} \hat{x}_i \\ \hat{y}_i \end{pmatrix} = \begin{pmatrix} x_i \\ y_i \end{pmatrix} + \begin{pmatrix} \Delta x_i \\ \Delta y_i \end{pmatrix}. \quad (3.5)$$

Here the superscript T denotes the matrix transpose.

Let this perturbation $\Delta P_i = (\Delta x_i, \Delta y_i)^T$ on P_i be called the input perturbation. We assume that ΔP_i and ΔP_j are independent for $i \neq j$, but Δx_i and Δy_i may be correlated for $i = 1, \dots, N$.

Let $\Delta \mathbf{x} = (\Delta x_1, \dots, \Delta x_N)^T$, and $\Delta \mathbf{y} = (\Delta y_1, \dots, \Delta y_N)^T$. Because the perturbations Δx_i and Δy_i are zero-mean, we have $\mathbf{E}(\Delta \mathbf{x}) = \mathbf{E}(\Delta \mathbf{y}) = 0$. Since ΔP_i and ΔP_j are independent for $i \neq j$,

$$\mathbf{Cov}(\Delta \mathbf{x}, \Delta \mathbf{x}) = \text{Diag}(\sigma_{x,1}^2, \sigma_{x,2}^2, \dots, \sigma_{x,N}^2) \quad (3.6)$$

$$\mathbf{Cov}(\Delta \mathbf{y}, \Delta \mathbf{y}) = \text{Diag}(\sigma_{y,1}^2, \sigma_{y,2}^2, \dots, \sigma_{y,N}^2) \quad (3.7)$$

$$\begin{aligned} \mathbf{Cov}(\Delta \mathbf{x}, \Delta \mathbf{y}) &= \mathbf{Cov}(\Delta \mathbf{y}, \Delta \mathbf{x}) \\ &= \text{Diag}(\sigma_{xy,1}^2, \sigma_{xy,2}^2, \dots, \sigma_{xy,N}^2) \end{aligned} \quad (3.8)$$

are diagonal matrices. The diagonal elements of these matrices are specified by the noise model and will be given by the edgel location characterization discussed in Chapter 5.

3.4 Uncertainty on the centroid location

Being a deterministic function of $\{P_i\}$, the centroid Q is affected by the perturbation on $\{P_i\}$. Let the centroid perturbation be denoted by

$$\Delta Q = Q(\mathbf{x} + \Delta \mathbf{x}, \mathbf{y} + \Delta \mathbf{y}) - Q(\mathbf{x}, \mathbf{y}) \quad (3.9)$$

or, in terms of its elements,

$$\Delta Q_x = Q_x(\mathbf{x} + \Delta \mathbf{x}, \mathbf{y} + \Delta \mathbf{y}) - Q_x(\mathbf{x}, \mathbf{y}) \quad (3.10)$$

$$\Delta Q_y = Q_y(\mathbf{x} + \Delta \mathbf{x}, \mathbf{y} + \Delta \mathbf{y}) - Q_y(\mathbf{x}, \mathbf{y}) \quad (3.11)$$

Within this chapter, this perturbation is called the output perturbation. The quantity of interest is $\mathbf{Cov}(\Delta Q)$, namely the covariance of the output perturbation. This is a full measure for the target centroid location uncertainty, the trace of which is to be used as an inverted saliency measure for the target's presence.

3.4.1 Finding the covariance of the output perturbation

We have just defined the measure for the centroid location uncertainty. In order to compute its value, we need to express $\mathbf{Cov}(\Delta Q)$ in terms of the locations of the unperturbed vertices and the covariance of the input perturbation, which are given and computable entities, respectively.

Let us first study ΔQ_x . Let $Q_x(\mathbf{x} + \Delta \mathbf{x}, \mathbf{y} + \Delta \mathbf{y})$ be approximated by the linear terms in the Taylor series expansion.

$$Q_x(\mathbf{x} + \Delta \mathbf{x}, \mathbf{y} + \Delta \mathbf{y}) \approx Q_x(\mathbf{x}, \mathbf{y}) + A^T \cdot \Delta \mathbf{x} + B^T \cdot \Delta \mathbf{y} \quad (3.12)$$

where $A = \frac{\partial Q_x(\mathbf{x}, \mathbf{y})}{\partial \mathbf{x}}$ and $B = \frac{\partial Q_x(\mathbf{x}, \mathbf{y})}{\partial \mathbf{y}}$ are $N \times 1$ real vectors, which can be computed using Equations (3.21) and (3.22). Then

$$\Delta Q_x \approx A^T \cdot \Delta \mathbf{x} + B^T \cdot \Delta \mathbf{y} \quad (3.13)$$

$$\mathbf{E}(\Delta Q_x) \approx 0 \quad (3.14)$$

$$\begin{aligned} \mathbf{Var}(\Delta Q_x) \approx & A^T \mathbf{Cov}(\Delta \mathbf{x}, \Delta \mathbf{x})A + 2A^T \mathbf{Cov}(\Delta \mathbf{x}, \Delta \mathbf{y})B \\ & + B^T \mathbf{Cov}(\Delta \mathbf{y}, \Delta \mathbf{y})B \end{aligned} \quad (3.15)$$

Similarly, we obtain

$$\mathbf{E}(\Delta Q_y) \approx 0 \quad (3.16)$$

$$\begin{aligned} \mathbf{Var}(\Delta Q_y) \approx & C^T \mathbf{Cov}(\Delta \mathbf{x}, \Delta \mathbf{x})C + 2C^T \mathbf{Cov}(\Delta \mathbf{x}, \Delta \mathbf{y})D \\ & + D^T \mathbf{Cov}(\Delta \mathbf{y}, \Delta \mathbf{y})D \end{aligned} \quad (3.17)$$

$$\begin{aligned} \mathbf{Cov}(\Delta Q_x, \Delta Q_y) \approx & A^T \mathbf{Cov}(\Delta \mathbf{x}, \Delta \mathbf{x})C + A^T \mathbf{Cov}(\Delta \mathbf{x}, \Delta \mathbf{y})D \\ & + B^T \mathbf{Cov}(\Delta \mathbf{y}, \Delta \mathbf{x})C + B^T \mathbf{Cov}(\Delta \mathbf{y}, \Delta \mathbf{y})D \end{aligned} \quad (3.18)$$

where $C = \frac{\partial Q_y(\mathbf{x}, \mathbf{y})}{\partial \mathbf{x}}$ and $D = \frac{\partial Q_y(\mathbf{x}, \mathbf{y})}{\partial \mathbf{y}}$ are $N \times 1$ real vector, which can be computed using Equations (3.23) and (3.24). With these values computed, we obtain approximately

$$\mathbf{E}(\Delta Q) = 0 \quad (3.19)$$

$$\mathbf{Cov}(\Delta Q, \Delta Q) = \begin{pmatrix} \mathbf{Var}(\Delta Q_x) & \mathbf{Cov}(\Delta Q_x, \Delta Q_y) \\ \mathbf{Cov}(\Delta Q_y, \Delta Q_x) & \mathbf{Var}(\Delta Q_y) \end{pmatrix} \quad (3.20)$$

where all terms have been obtained in terms of the locations of the unperturbed vertices and the covariance of the input perturbation. The 2×2 covariance matrix $\mathbf{Cov}(\Delta Q, \Delta Q)$ is a full characterization of the location uncertainty of the centroid. It describes how the centroid's likely position is spread out in the 2D domain. A proper scalar measure of the uncertainty is the trace of the matrix $\mathbf{Var}(\Delta Q_x) + \mathbf{Var}(\Delta Q_y)$, i.e., the sum of the two diagonal elements.

In the ATR algorithm, the locations of the unperturbed vertices are given by the locations of the target silhouette boundary points relative to the target's centroid. The covariance of the input perturbation is determined by the SNR estimated at the boundary points and the characterization of boundary location uncertainty as given in Chapter 5. The trace of the output covariance matrix is inversely related to the saliency of the target's presence. A strong target appearance will result in a small trace of the centroid covariance.

3.4.2 Expressions for the partial derivatives

We now give the exact expressions for the partial derivatives involved in the equations given above.

When the locations of the polygon's vertices are perturbed, K_x , K_y , and S in Equation (3.1) are all perturbed. Let the perturbed quantities be \hat{K}_x , \hat{K}_y , and \hat{S} , respectively. Let ΔK_x , ΔK_y , and ΔS be the difference between the perturbed values and the ideal values, respectively.

$$\begin{aligned}
 \hat{S} &= \sum_{i=1}^N (\hat{x}_i \hat{y}_{i+1} - \hat{x}_{i+1} \hat{y}_i) \\
 &= \sum_{i=1}^N [(x_i + \Delta x_i)(y_{i+1} + \Delta y_{i+1}) - (x_{i+1} + \Delta x_{i+1})(y_i + \Delta y_i)] \\
 \Delta S &= \hat{S} - S \\
 &\approx \sum_{i=1}^N [(x_i \Delta y_{i+1} + y_{i+1} \Delta x_i) - (x_{i+1} \Delta y_i + y_i \Delta x_{i+1})] \\
 \hat{K}_x &= \sum_{i=1}^N (\hat{x}_i \hat{y}_{i+1} - \hat{x}_{i+1} \hat{y}_i)(\hat{x}_i + \hat{x}_{i+1}) \\
 &= \sum_{i=1}^N [(x_i + \Delta x_i)(y_{i+1} + \Delta y_{i+1}) - (x_{i+1} + \Delta x_{i+1})(y_i + \Delta y_i)] \\
 &\quad \cdot (x_i + \Delta x_i + x_{i+1} + \Delta x_{i+1}) \\
 \Delta K_x &= \hat{K}_x - K_x
 \end{aligned}$$

$$\begin{aligned}
& \approx \sum_{i=1}^N \{ (x_i y_{i+1} - x_{i+1} y_i) (\Delta x_i + \Delta x_{i+1}) \\
& \quad + (x_i + x_{i+1}) [(x_i \Delta y_{i+1} + y_{i+1} \Delta x_i) - (x_{i+1} \Delta y_i + y_i \Delta x_{i+1})] \} \\
\hat{K}_y &= \sum_{i=1}^N (\hat{x}_i \hat{y}_{i+1} - \hat{x}_{i+1} \hat{y}_i) (\hat{y}_i + \hat{y}_{i+1}) \\
&= \sum_{i=1}^N [(x_i + \Delta x_i)(y_{i+1} + \Delta y_{i+1}) - (x_{i+1} + \Delta x_{i+1})(y_i + \Delta y_i)] \\
& \quad \cdot (y_i + \Delta y_i + y_{i+1} + \Delta y_{i+1}) \\
\Delta K_y &= \hat{K}_y - K_y \\
&\approx \sum_{i=1}^N \{ (x_i y_{i+1} - x_{i+1} y_i) (\Delta y_i + \Delta y_{i+1}) \\
& \quad + (y_i + y_{i+1}) [(x_i \Delta y_{i+1} + y_{i+1} \Delta x_i) - (x_{i+1} \Delta y_i + y_i \Delta x_{i+1})] \}
\end{aligned}$$

Using the convention that $x_0 = x_N, y_0 = y_N$, we obtain

$$\begin{aligned}
\frac{\partial S}{\partial x_i} &= y_{i+1} - y_{i-1} \\
\frac{\partial S}{\partial y_i} &= -(x_{i+1} - x_{i-1}) \\
\frac{\partial K_x}{\partial x_i} &= [2x_i y_{i+1} + x_{i+1}(y_{i+1} - y_i)] - [2x_i y_{i-1} + x_{i-1}(y_{i-1} - y_i)] \\
\frac{\partial K_x}{\partial y_i} &= -(x_i + x_{i+1})x_{i+1} + (x_{i-1} + x_i)x_{i-1} \\
\frac{\partial K_y}{\partial x_i} &= (y_i + y_{i+1})y_{i+1} - (y_{i-1} + y_i)y_{i-1} \\
\frac{\partial K_y}{\partial y_i} &= -[2x_{i+1}y_i + y_{i+1}(x_{i+1} - x_i)] + [2x_{i-1}y_i + y_{i-1}(x_{i-1} - x_i)]
\end{aligned}$$

for $i = 1, 2, \dots, N$.

Now the expressions

$$\frac{\partial Q_x(\mathbf{x}, \mathbf{y})}{\partial x_n} = \frac{\frac{\partial K_x}{\partial x_n} S - K_x \frac{\partial S}{\partial x_n}}{3S^2} = \frac{1}{3S} \cdot \left\{ \frac{\partial K_x}{\partial x_n} - \frac{K_x}{S} \cdot \frac{\partial S}{\partial x_n} \right\} \quad (3.21)$$

$$\frac{\partial Q_x(\mathbf{x}, \mathbf{y})}{\partial y_n} = \frac{\frac{\partial K_x}{\partial y_n} S - K_x \frac{\partial S}{\partial y_n}}{3S^2} = \frac{1}{3S} \cdot \left\{ \frac{\partial K_x}{\partial y_n} - \frac{K_x}{S} \cdot \frac{\partial S}{\partial y_n} \right\} \quad (3.22)$$

$$\frac{\partial Q_y(\mathbf{x}, \mathbf{y})}{\partial x_n} = \frac{\frac{\partial K_y}{\partial x_n} S - K_y \frac{\partial S}{\partial x_n}}{3S^2} = \frac{1}{3S} \cdot \left\{ \frac{\partial K_y}{\partial x_n} - \frac{K_y}{S} \cdot \frac{\partial S}{\partial x_n} \right\} \quad (3.23)$$

$$\frac{\partial Q_y(\mathbf{x}, \mathbf{y})}{\partial y_n} = \frac{\frac{\partial K_y}{\partial y_n} S - K_y \frac{\partial S}{\partial y_n}}{3S^2} = \frac{1}{3S} \cdot \left\{ \frac{\partial K_y}{\partial y_n} - \frac{K_y}{S} \cdot \frac{\partial S}{\partial y_n} \right\} \quad (3.24)$$

only involve the known locations of the unperturbed vertices, and hence are readily computable.

3.5 Centroid uncertainty validation experiment

We have just derived the formulae for computing the polygon centroid uncertainty from the unperturbed vertex locations and the covariance of the perturbations on the vertices. In the derivation, linear approximations were involved. In order to verify the validity of the derivation results, we conduct controlled experiments and compare empirically observed results with the theoretical predictions given by the formulae.

Given the vertices of a polygon as well as the perturbation on each vertex, the covariance matrix of the centroid location is computed according to the formulae obtained in Section 3.4.1. On the other hand, we may carry out simulations that actually generate the specified perturbations on the polygon vertices. The centroid of the perturbed polygon is computed easily. By repeating this many times, we can obtain the sample covariance matrix of the centroid. If the derived formulae for the centroid uncertainty are correct, the observed sample covariance matrix should agree with the matrix given by the formulae. Then we consider the result represented by the formulae as valid.

3.5.1 Polygon and perturbation used in the experiments

In the validation experiment we use a quadrilateral as the polygon. The coordinates of its vertices and the perturbation on the vertices are given in Table 3.1. In this experiment, we use 1D perturbations on the vertices, i.e., each vertex is perturbed along a certain direction. This gives linearly correlated Δx_i and Δy_i with correlation coefficient 1. Specifically, two of the perturbations were close to being perpendicular

Table 3.1: Specification of the polygon and perturbation used in the validation experiment.

vertex	coordinates	direction (angle in degree)	variance
1	$(-10, -10)$	135	3^2
2	$(20, -5)$	60	2.4^2
3	$(10, 12)$	75	2.1^2
4	$(-15, 5)$	0	2.7^2

to the tangent direction of the smoothed polygon contour and two of the perturbations were close to the tangent direction of the smoothed polygon contour.

Figure 3.2(a) shows the test polygon. The origin $O(0, 0)$ is the meeting point of the four dotted lines connect the origin with the vertices. These along with the edges of the polygon form the triangles for finding the centroid of the polygon. The unperturbed centroid $Q(1.735, 0.409)$ is marked by the cross. The direction and standard deviation of the 1D perturbation on each vertex is shown by the direction and length of the line segment attached to the vertex.

3.5.2 Experiment procedure

The public domain software package RANDLIB.C [18] is used for generating multivariate normal (MVN) random variables to be the perturbation added to the location of the vertices. The mean value for the perturbation is zero, and the variances are as specified in Table 3.1

Each random sample generated is used to perturb the vertex locations. The location of the perturbed polygon's centroid \hat{Q} is calculated using Equation (3.1). This process is repeated 3,000 times. A sample scatter-gram of the perturbed centroid location is shown in Figure 3.2(b). The sample mean and sample covariance matrix of

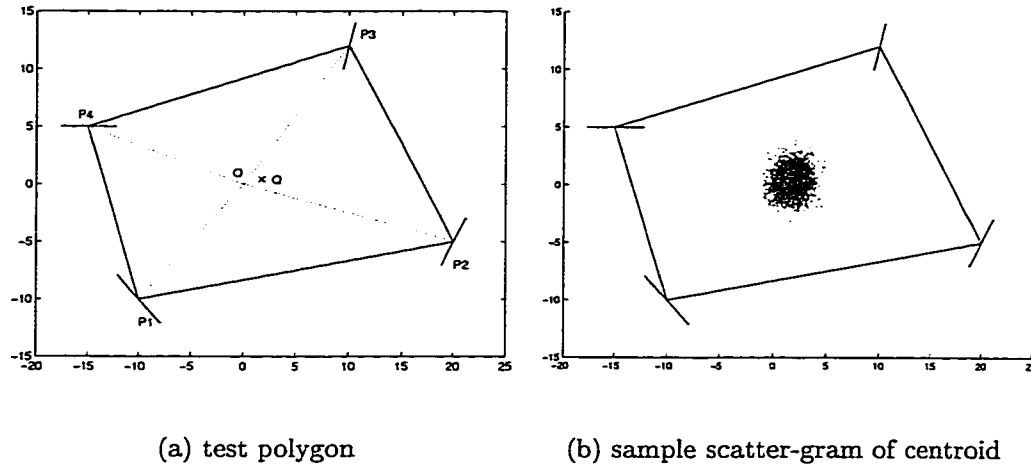


Figure 3.2: Polygon, perturbation on vertices, and sample centroid locations in the validation experiment. In (a), O is the origin; Q is the unperturbed centroid; the direction and length of the line segments at each vertex show the direction and the standard deviation of the perturbation affecting that vertex.

the scatter-gram are computed. The agreement of these sample parameters with the analytical prediction given by Equations (3.19) and (3.20) is tested by applying a statistical test [72, 4] that compares the sample mean and covariance with the predicted ones.

In this test, a statistic is calculated that shows the significance of the difference between the sample parameters and the predicted parameters. One set of predicted parameters and one set of sample parameters are used to produce one single scalar value of the test statistic t . In this problem, t should follow the χ_5^2 distribution, namely the chi-squared distribution with 5 degrees of freedom.

3.5.3 Validation results

We repeat the whole procedure (perturbation generation – centroid calculation – sample parameter estimation – test statistic calculation) for 2,000 times and obtain

2,000 values of the test statistic t . The Kolmogorov-Smirnov (KS) test [121] is used to examine if the sample distribution is close enough to the χ^2_5 distribution. The KS test gives a p -value between zero and one indicating the significance of the agreement. Good agreement should have moderately large p -values, and very small (less than 0.01 or 0.05) p -values indicate bad agreement.

This is done four times using different seeds for the random number generator. The p -values from the KS test are 0.5596, 0.2008, 0.0725, and 0.2779. The fairly large p -values show strong support for the agreement between the theoretical prediction of the behavior of the centroid's location and the actual observed behavior. Some histograms of the test statistic and the underlying χ^2_5 probability density function (pdf) are plotted in Figure 3.3 to assist the reader in qualitatively judging the agreement.

3.6 Detection performance prediction

The computation of centroid covariance matrix developed and validated in the earlier parts of this chapter is to be used for the target saliency measure in target detection and recognition applications. As will be discussed in the following chapters, the contrast evidence along the hypothetical target boundary is measured by contrast and signal-to-noise ratio (SNR) estimation through the use of a gradient operator. This evidence is converted into the location uncertainty of the boundary point location. The formula derived in this chapter is then used to combine the boundary point location uncertainty into the centroid location uncertainty.

When the boundary points have higher contrast and SNR, their location uncertainty is smaller. (See Chapter 5 for the development of the relationship between location uncertainty and SNR.) This usually results in smaller centroid location uncertainty. Therefore, a small value of the centroid location uncertainty is an indication of the presence of the hypothetical target. A threshold can be applied to the computed uncertainty value. When the value is smaller than the threshold, we declare

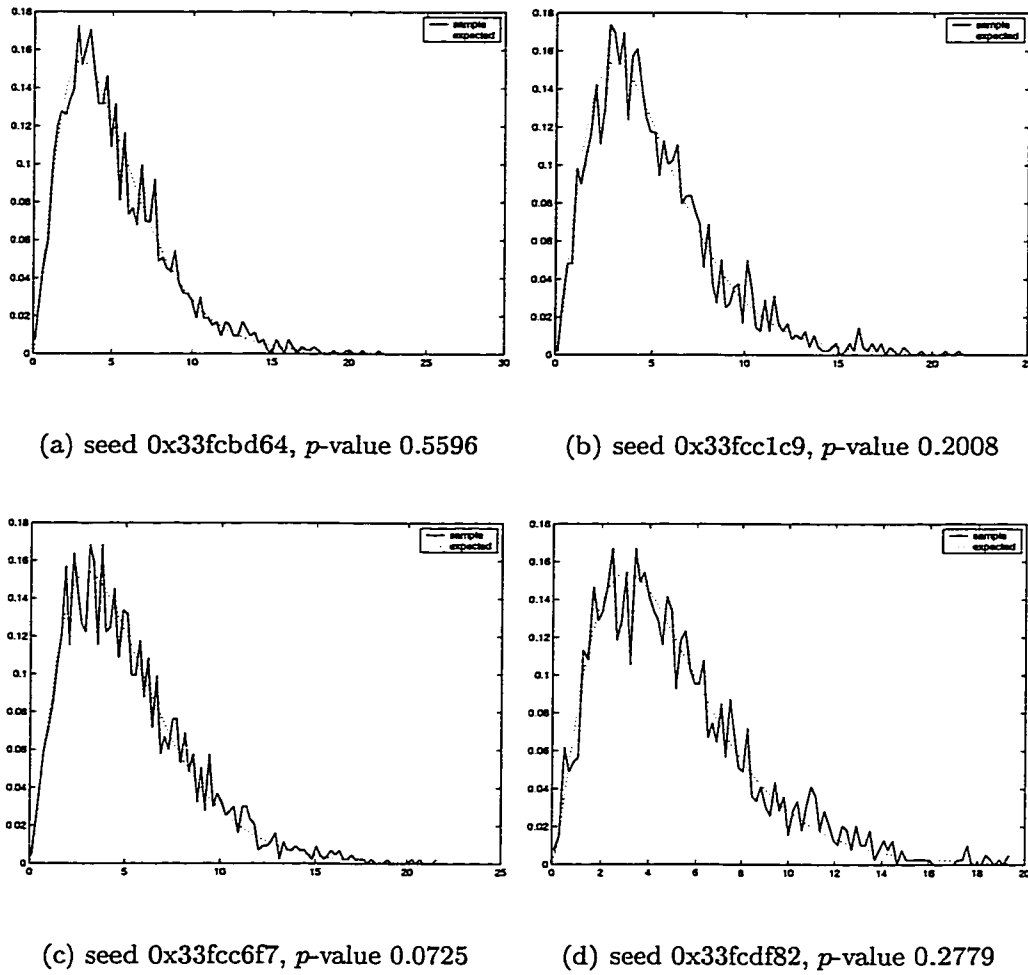


Figure 3.3: Sample and theoretical pdf of the test statistic t from the test on centroid uncertainty. Solid curves show the sample pdf, and the dotted curves show the pdf of χ^2_5 . The p -values are the results from the KS tests. Different seeds for the random number generator are used in obtaining the sample pdf's.

the presence of the target. Otherwise, no target is declared when the value is larger than the threshold.

As just mentioned, apart from the geometric shape of the target, the essential factor that determines the value of the centroid location uncertainty is the SNR along the target boundary. It is desirable to have a quantitative characterization of the centroid uncertainty when a quantitative characterization of the boundary SNR is given. Specifically, we consider characterizing the centroid uncertainty when the statistical distribution of the boundary SNR is given. A closed-form analytical expression for the distribution of the centroid uncertainty is difficult. This is due to the non-linear relationship between the boundary point location uncertainty and the SNR, and due to the linear combination of the boundary point location uncertainty in computing the centroid uncertainty. Instead, we consider obtaining the distribution of the centroid uncertainty using the Monte Carlo methods by resampling from the SNR distribution and gather samples of the computed centroid uncertainty values. The misdetection and false alarm performance from thresholding the centroid uncertainty value can then be predicted for the given SNR distribution. In this section, only one target is studied as an example.

3.6.1 Monte Carlo experiment with independently sampled boundary SNR

Figure 3.4 shows the sample distributions of the SNR for the target boundary and the background in image chips generated by the procedure described in Chapter 7. In obtaining the distribution of the centroid uncertainty value for the target, random samples of the SNR are obtained independently from the target boundary SNR distribution. These sampled values are used to compute the boundary location covariance matrices by the method and data described in Chapter 5. The boundary point location covariance matrices are then used to compute the centroid location covariance matrix for the target being studied. The normalized trace of the centroid location covariance matrix, which is normalized by the trace of the matrix for zero

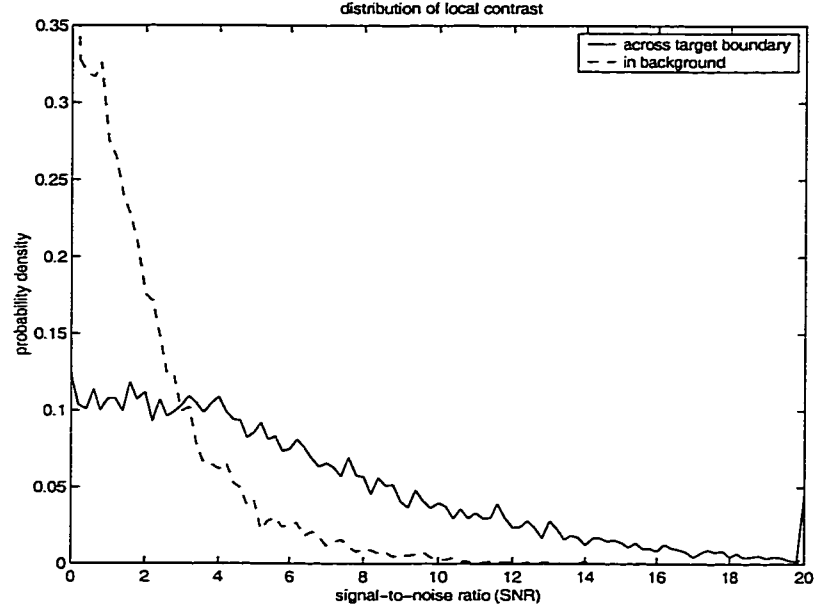


Figure 3.4: Sample distribution of local SNR for target boundary and background in simulated FLIR image chips.

contrast across the target boundary, is used as the measure of centroid uncertainty. This is repeated many times, which results in a Monte Carlo distribution of the centroid uncertainty for the target with the given boundary SNR distribution. A similar procedure for the background SNR distribution yields the Monte Carlo distribution of the centroid uncertainty for the background. These Monte Carlo distributions are shown in Figure 3.5(a).

In order to assess the validity of the Monte Carlo distributions, we compare them to actual distributions of the centroid uncertainty which are obtained by applying entire algorithm sequence (SNR estimation, conversion to boundary location covariance, centroid uncertainty computation) on the image chips. The difference here from the Monte Carlo experiments is that the SNR values are estimated locally in the image and not randomly sampled from the SNR distributions in Figure 3.4. This experiment is referred to as the non-Monte Carlo experiment. The observed centroid uncertainty

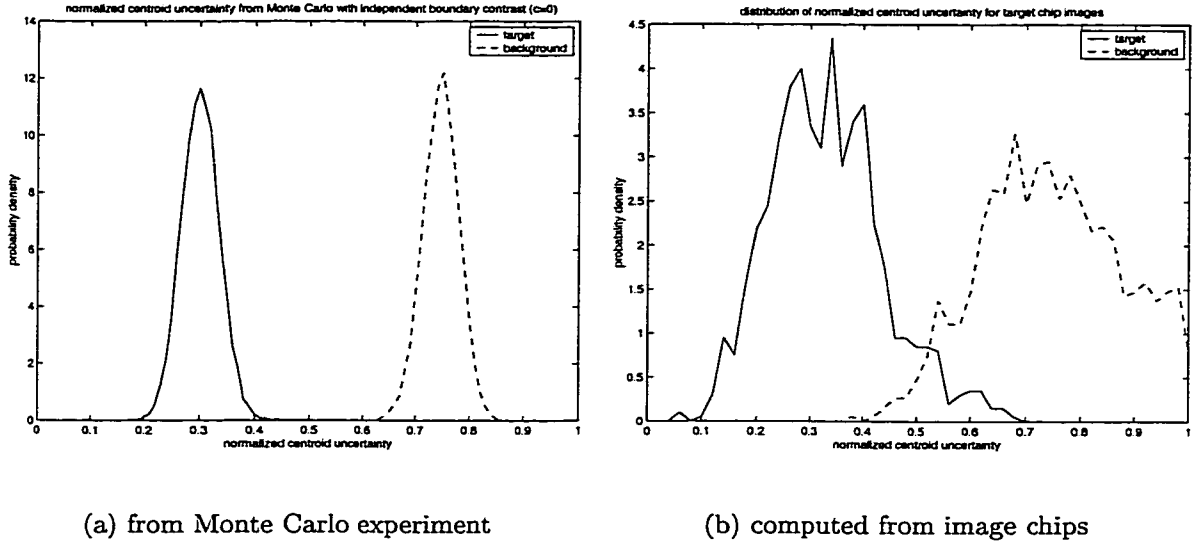


Figure 3.5: Distributions of the centroid uncertainty from Monte Carlo experiment with independent sampling and from values directly computed from image chips.

distributions in this experiment are shown in Figure 3.5(b). It is seen clearly that the real distributions are much more spread out than the ones obtained in the Monte Carlo experiment. Investigation reveals that the independent resampling of the SNR values is inappropriate. Figure 3.6 shows the correlation between the SNR values at adjacent locations along the target boundary. It is clear that for both real target boundary locations and background locations, the SNR values at adjacent locations are highly correlated. The mean and median of the correlation coefficient are both very close to 0.9 for both true target boundary and background.

3.6.2 Monte Carlo experiment with correlated boundary SNR

Recognizing the high correlation between the SNR values at adjacent locations in the FLIR imagery, the Monte Carlo experiment is conducted again with correlation introduced in resampling the SNR values. The correlation coefficient is assumed to

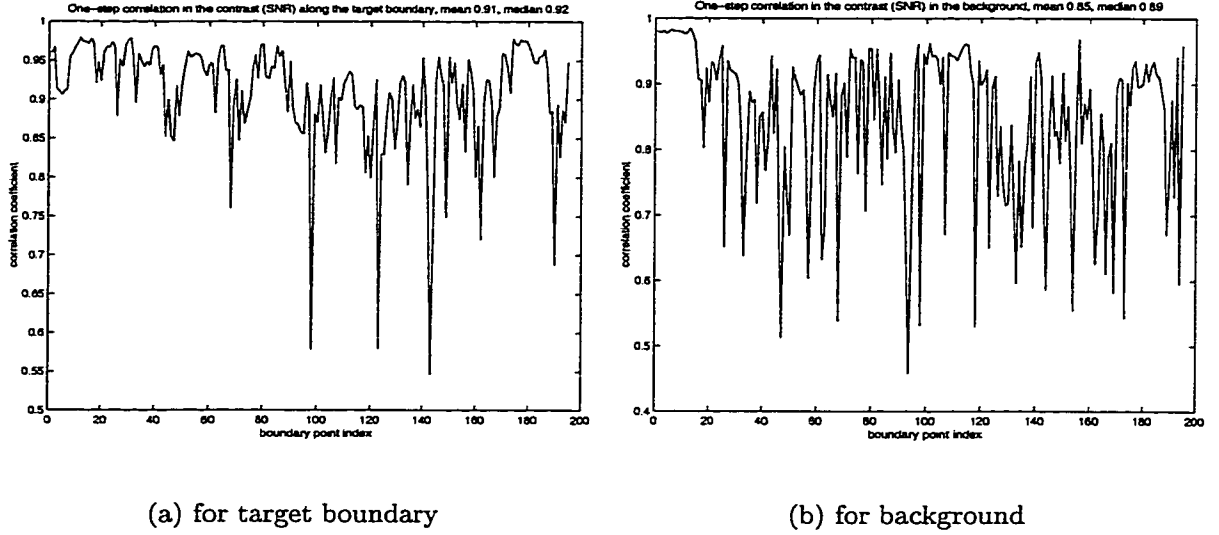


Figure 3.6: Correlation coefficient for SNR values at adjacent locations along the boundary of a particular target.

be an exponentially decaying function of the distance between pixel locations

$$c = \exp^{-d/d_0},$$

where d_0 is the constant controlling the rate of decay. As suggested by the sample correlation shown in Figure 3.6, the correlation coefficient c is set to 0.9 for adjacent pixels ($d = 1$), which leads to $d_0 = -1/\log 0.9 = 9.49$. The correlated SNR samples are drawn by the following procedure. The public domain software package RANDLIB.C [18] is used for generating multivariate normal random vectors with unit variance and the correlation just specified. Each element in the random vector is transformed independently to a value between 0 and 1 by the cumulative distribution function. This value multiplied by the number of total SNR value samples in pool of SNR samples from which the sample distributions in Figure 3.4 are obtained. The result is used as an index into the sorted list of the SNR values in that pool. The value retrieved from the list by that index replaces the value of the original normal

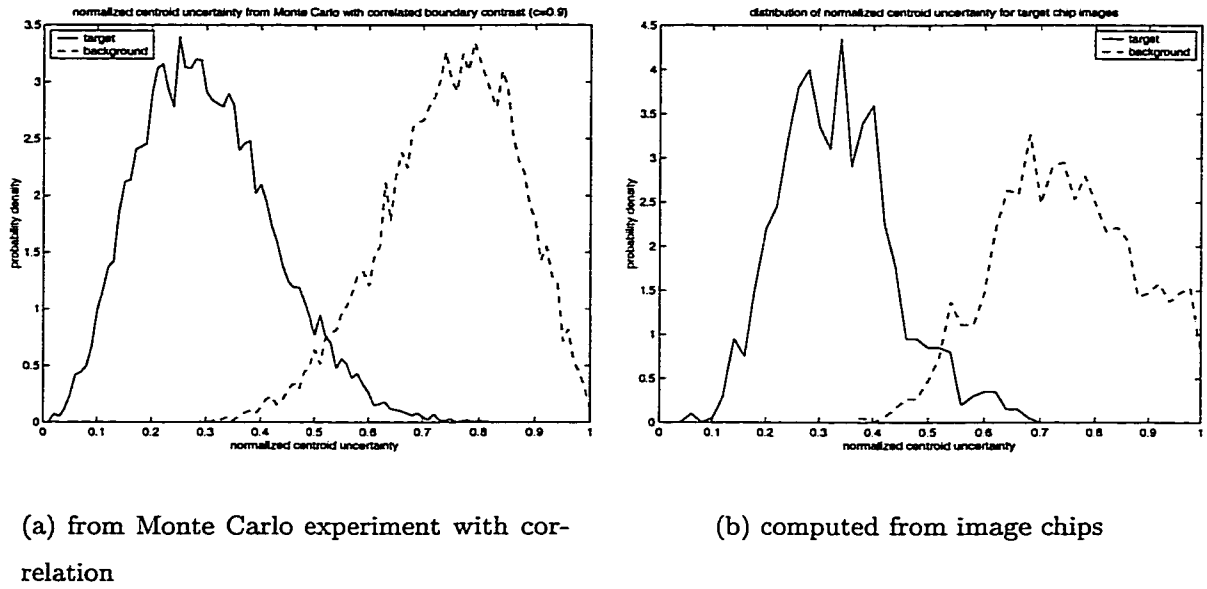


Figure 3.7: Distributions of the centroid uncertainty from Monte Carlo experiment with correlated SNR sampling and from values directly computed from image chips.

random variable. This way, the correlation between the elements of the random vector is retained, while the distribution of each element is changed from the standard normal distribution to the distribution shown in Figure 3.4.

After the introduction of correlation in the resampling of boundary SNR values, the distribution of the centroid uncertainty from the Monte Carlo experiment becomes very close to that from computed directly from the image chips. These are shown in Figure 3.7.

3.6.3 Error rates and confidence intervals

When the distributions of the centroid uncertainty value for the target and background overlap each other as shown in Figure 3.7, target detection by thresholding the centroid uncertainty value will inevitably produce misclassification errors, i.e.,

misdetectors and/or false alarms. A misdetection occurs when the centroid uncertainty computed for a target class instance (an image chip containing a target in the non-Monte Carlo experiment, and resampling from the target boundary SNR distribution in the Monte Carlo experiment) is larger than the threshold. A false alarm occurs when the centroid uncertainty computed for a non-target class (an image chip for the background in the non-Monte Carlo experiment, and resampling from the background SNR distribution in the Monte Carlo experiment) is smaller than the threshold. The misdetection rate is the ratio of misdetections in the total number of measurements made for the target class. The false alarm rate is the ratio of false alarms in the total number of measurements for the non-target class.

The curves in Figure 3.7 represent experimental results for 1000 target class instances and 1000 non-target class instances in the non-Monte Carlo experiment, and for 1,000,000 target class and non-target class instances in the Monte Carlo experiment. The threshold for the centroid uncertainty is determined by the minimum error criteria on the non-Monte Carlo experiment. This is the threshold associated with the smallest sum of misdetections and false alarms. Among all such threshold values, we choose one that gives approximately (in this case exactly) the same numbers of misdetections and false alarms. This threshold value is $\tau = 0.534$. The number of mis-classifications are 42 misdetections and 42 false alarms for the non-Monte Carlo experiment. Applying this same threshold value on the data from the Monte Carlo experiment, we find 41,349 misdetections and 53,762 false alarms. Therefore, the misdetection and false alarm rates for the non-Monte Carlo experiment are 0.042 and 0.042, respectively, and for the Monte Carlo experiment are 0.041 and 0.054, respectively.

One way of assessing the goodness of the prediction of the error rates, i.e., to judge the closeness of the error rates from the Monte Carlo experiment to those from the non-Monte Carlo experiment, is to find the confidence intervals [21] of the true error rates based on the data from the non-Monte Carlo experiment. The following is a

Bayesian formulation for the computation of the confidence intervals [53].

Let n be the total number of measurements made in the experiment (for computing either the misdetection rate or the false alarm rate, but not both). Let k be the number of errors observed. Let the true error rate be denoted by p . Assuming the measurements made in the experiments are independent, k follows the Binomial distribution $B(n, p)$ with the probability mass function (pmf) [21]

$$f(k|n, p) = \binom{n}{k} p^k (1-p)^{n-k}.$$

In the experiments, the total number of measurements n is a fixed constant and is not treated as a random variable. By the Bayes rule, the *a posteriori* probability density for the true error rate p is

$$f(p|k) = \frac{f(k|p)f(p)}{\int_0^1 f(k|p)f(p)dp}.$$

We take the uniform prior probability $f(p) = 1$ for no prior judgment of the likelihood of the true error rate. The denominator of the right-hand side of the above equation is computed as

$$\begin{aligned} \int_0^1 f(k|p)f(p)dp &= \int_0^1 \binom{n}{k} p^k (1-p)^{n-k} dp \\ &= \frac{1}{n+1} \int_0^1 \frac{1}{\text{Beta}(k+1, n-k+1)} p^k (1-p)^{n-k} dp \\ &= \frac{1}{n+1}. \end{aligned}$$

The last equation is due to the integrand being the pdf of the *beta* distribution [21] and

$$\text{Beta}(k+1, n-k+1) = \frac{\Gamma(k+1)\Gamma(n-k+1)}{\Gamma(n+2)} = \frac{k!(n-k)!}{(n+1)!}.$$

Therefore,

$$f(p|k) = (n+1)f(k|p)f(p) = \frac{1}{\text{Beta}(k+1, n-k+1)} p^k (1-p)^{n-k},$$

i.e., the *a posteriori* distribution of p given k is the beta distribution with parameters $k + 1$ and $n - k + 1$.

For an interval $(p_1, p_2) \subseteq [0, 1]$, the *a posteriori* probability that the true error rate belongs to this interval, which is also called the confidence of the interval, is then

$$\int_{p_1}^{p_2} f(p|k)dp = I_{p_2}(k + 1, n - k + 1) - I_{p_1}(k + 1, n - k + 1), \quad (3.25)$$

where $I_x(z, w)$ is the incomplete beta ratio function [1, 21].

Here, the information about the true error rates is carried by the data in the non-Monte Carlo experiment. We want to use that information to find the coverage probability of an interval centered on the predicted error rates from the Monte Carlo experiment. Let \hat{p} denote the predicted error rate from the Monte Carlo experiment. Let δ be some small positive constant. The coverage probability for the interval

$$(\hat{p} - \delta, \hat{p} + \delta),$$

around the predicted error rate is

$$I_{\hat{p}+\delta}(k + 1, n - k + 1) - I_{\hat{p}-\delta}(k + 1, n - k + 1),$$

where $n = 1000$ is the number of total samples in the non-Monte Carlo experiment, and $k = 42$ is the number of errors. (In general k is different for misdetections and false alarms. It just happens to be the same in this particular experiment.)

For the misdetection performance, $\hat{p} = 0.041$. Choosing $\delta = 0.02$, we find the coverage probability for the interval $(0.021, 0.061)$ to be 0.995. Considering the high coverage probability and the relative small size of the interval, the predicted misdetection rate is fairly close to the true error rate, the exact value of which is unknown.

For the false alarm performance, $\hat{p} = 0.054$. Also taking $\delta = 0.02$, we find the coverage probability for the interval $(0.034, 0.074)$ to be 0.926. Increasing the interval size by letting $\delta = 0.025$, the coverage probability for the interval $(0.029, 0.079)$ is increased to 0.992. This shows that the predicted false alarm rate is also close to the

unknown true false alarm rate, although the quality of the prediction is not as good as that for the misdetection performance.

Chapter 4

OPTIMAL ESTIMATION OF IMAGE GRADIENT

The problem of optimal estimation of image gradient is conceived as searching for the gradient operator that optimizes a properly defined performance measure. We base our discussion on a unified framework for characterizing the performance of a broad class of linear gradient estimators, which includes most popular gradient operators. This framework provides the analytic expressions for the bias and variance of the gradient direction estimate. The values computed from these expressions for various gradient operators agree very closely with previously published results which are all based on empirical simulations. Based on this framework, we also explicitly identify a direct relationship between the gradient magnitude estimation performance and the variance of the direction estimate.

We use the mean-squared error (MSE) of the gradient direction estimate to measure gradient operator performance. The MSE is computed from analytic expressions for the bias and variance. It is desirable for gradient operators to have small MSE for all possible situations, including edge orientation, location displacement, and different signal-to-noise ratio (SNR) levels. For a given SNR, the worst-case estimation error for the gradient direction is the maximum value of the MSE over all possible edge orientations and displacement. The optimal gradient operator design problem is formulated as a minimax optimization problem that minimizes the worst-case error by searching over the tuning parameters of the gradient operator. We solve this problem numerically for the neighborhood sizes of 3×3 and 5×5 . The performance of the resulting operator compares favorably with the optimally chosen members of three popular classes of gradient operators, namely the Sobel class, the integrated

directional derivative class, and the derivative of Gaussian class. The integrated directional derivative operators has the highest robustness in that its tuning parameter stays almost unchanged throughout a wide range of image signal-to-noise ratio, while the parameters of other classes of operators vary more with the image signal-to-noise ratio.

In the major parts of this chapter, the step edge model is used as the image intensity function of interest. However, the framework and general methodology introduced here is not in any way restricted to this edge model. The methodology is readily applicable to other image intensity patterns, e.g., the ramp edges. For any image intensity pattern of interest and any neighborhood size of interest, the performance of any given gradient operator can be characterized by directly applying the formulas introduced in this chapter. The same type of minimax problem can be set up and solved to find the optimal operator in that situation. As an example, the image intensity pattern of a class of smoothed edges is studied in the experiment. The optimal gradient operator from the solution of the minimax problem is again observed to out-perform the other three classes of operators. Furthermore, the derivative of Gaussian operator also adjusts well to the smoothing. The integrated directional derivative operator again shows the highest robustness in its tuning parameter in response to the amount of image smoothing.

4.1 Introduction

Like all other machine vision systems, ATR systems start their operations by the gathering of low level evidence and information from a sensory image. Generally, the operations for this purpose are referred to as feature extraction, and they often appear as earlier / lower-level modules in computer vision systems. Change detection is a very important and major issue in feature extraction. In the FLIR ATR problem that we consider in this dissertation, we must estimate the contrast evidence along the

target boundary. In an ideal image, a target appears as a compact region of constant intensity value on a uniform background of a different constant intensity value. The contrast between the target and background is the difference in the intensity values of the target and background. When moving from the inside of the target to the background or vice versa, the pixel intensity values along the path will show the pattern of a step edge. Therefore, the local contrast is defined to be the difference in the intensity values of the two sides of the step edge. In real applications, the intensity values of the target and the background are not constant. Contrast estimation in this situation requires some model fitting, which is studied extensively in the gradient estimation problem.

For a chosen model of the intensity change at the target boundary, which is the ideal sharp step edge model in this study, the gradient magnitude corresponds to the contrast via a one-to-one mapping. Due to the sampling, quantization, and noise in the digital images, gradient estimation for digital images is not a well defined problem. This fact along with the importance of change detection in early vision has resulted in a large number of studies on this issue reported in the literature [100, 51, 107, 19, 163, 124, 96, 28, 65, 45, 25, 149, 17, 150, 38, 133, 33, 145, 146, 98, 93]. One approach to handling the ill-posedness of the gradient estimation problem is via model fitting. We also take this approach in this dissertation. Local contrast and signal-to-noise ratio are estimated from the estimated gradient and noise variance.

4.1.1 A note on the dG operator

Currently, the most widely used image gradient estimator is the derivative of Gaussian (dG) operator used in Canny's edge operator [19]. In his widely-cited work, Canny studies the filter design problem in the 1-D case. He formulates an optimization problem in which the objective function is the output signal-to-noise ratio divided by the standard deviation of the location error. A penalty for multiple response from a single edge is introduced as a constraint. Canny uses numerical techniques to solve

the constrained optimization problem and obtains a filter that is very close to the first derivative of a Gaussian.

The particular form of the objective function and the constraint does not seem to bear more validity than an earlier not-so-successful attempt with an objective function that takes the simple product form of three terms for the output SNR, location error and multiple response. Also, from the numerous studies which establish the importance of the dG function in human vision and its optimality from many signal processing considerations, it is not without ground for our suspicion that the success in Canny’s formulation of the gradient operator design problem does not actually lie in the correctness of the formulation itself. Rather, it might well have resulted from the fact that the problem happens to adopt the dG as its solution, and the dG is a good operator for reasons other than (or more than) what is conveyed by Canny’s optimality criterion.

The 1-D formulation as used by Canny does not take into consideration the problem of gradient orientation estimation during the search for the optimal operator. Here we consider the problem in the 2-D setting. We determined that the gradient magnitude estimation performance and the orientation estimation performance are closely related via the operator output SNR, denoted by κ in this dissertation. κ^2 is the non-centrality parameter for the magnitude distribution and κ is the concentration parameter for the orientation distribution. We choose to measure the performance of a gradient operator by its worst-case mean squared error (MSE) in the orientation estimate. The optimal gradient operator design problem is then formulated as a min-max optimization problem. The cost function is the worst-case MSE over all possible edge orientations and displacement.

4.1.2 Other related works

The most closely related works on gradient operator performance characterization in the literature are those of Lyvers and Mitchell [96], and Zuniga and Haralick [163].

Lyvers and Mitchell conduct empirical experiments for a wide collection of gradient operators. The image intensity pattern studied is the square-aperture quantized sharp step edge of varying orientation and displacement. Experiments were conducted to observe the variance of the direction estimate for high SNR levels of 10dB, 20dB, 30dB and 40dB. The bias of the estimate is studied separately on the noise-free data. The conclusion from the experiment is that, for a good compromise between computational cost and overall estimation quality, the integrated directional derivative operator and the moment-based operator are approximately the best operators.

In addition to the empirical data, Lyvers and Mitchell also provide an analytic characterization of the magnitude and orientation estimates by giving the probability density functions (pdf) of their distributions. The pdf for the orientation estimate is

$$f_{\theta}(\theta) = e^{-\frac{\mu^2}{2\sigma^2}} \left\{ \frac{1}{2\pi} + \frac{\mu \cos(\theta - \phi)}{\sqrt{2\pi}\sigma} \exp \left[\frac{\mu^2 \cos^2(\theta - \phi)}{2\sigma^2} \right] \cdot \left[\frac{1}{2} + \frac{1}{2} \operatorname{erf} \left(\frac{\mu \cos(\theta - \phi)}{\sqrt{2}\sigma} \right) \right] \right\} \quad (4.1)$$

where ϕ is the noise-free estimate of the gradient orientation. We notify the readers of the Lyvers and Mitchell paper [96] that there are small but very significant derivation/typographical errors in the Equations (41) and (42) of their paper. The correct pdf is given by the equation above. Some characterizations of this pdf are given later in this chapter.

Zuniga and Haralick [163] introduced the integrated directional derivative gradient operator (IDDGO). This operator is based on the bi-variate cubic polynomial representation of the image intensity data. Gradient estimation is improved by integrating (averaging) the directional derivative over a small domain around the center of the neighborhood. This averaging operation is very effective in suppressing the effect of noise on the gradient orientation estimate. (As will become clear in this chapter, the gradient magnitude estimate is also improved at the same time.) Zuniga and Haralick carried out empirical experiments similar to those conducted by Lyvers and Mitchell [96] to compare the IDDGO with the Sobel operator, the Prewitt operator, and the

standard non-integrating cubic facet gradient operator. The tuning parameter for the IDDGO, i.e., the size of the integration domain, is also determined by empirical experiments.

In both of the works just mentioned, the authors just compare the performance measures of different operators. No effort is explicitly made to the search for the optimal operator that would achieve the upper bound of the performance according to their measures.

Iannino and Shapiro [65] consider this aspect of the problem, but only study the Sobel operators in the noise-free case. They propose the iterated Sobel operators in which the tuning parameters K_3 and K_5 in the 3×3 and 5×5 Sobel operators are iteratively determined, with the gradient orientation θ determined in the same iterative process. For the 3×3 case, the rule is

$$K_3 = \begin{cases} 2 & 0 \leq \theta \leq \tan^{-1} \frac{1}{3} \\ \frac{2.25 \tan^2 \theta - 1.25 \tan \theta - 1.25 + .25 / \tan \theta}{.25 / \tan \theta - 1.5 + 1.25 \tan \theta} & \tan^{-1} \frac{1}{3} \leq \theta \leq \frac{\pi}{4} \end{cases} \quad (4.2)$$

The process is initiated with $K_3 = 2$. The conditions for the convergence of the iterated operator are also given.

Elder and Zucker [38] use the dG operator for gradient estimation. They address the issue of scale selection (adaptively determining the value for the smoothing factor). The concept of minimum reliable scale is introduced. The smoothing factor has to be large enough for the magnitude of the first and second derivatives to be significant relative to the noise. The sub-pixel edge location is decided as the negatively sloped zero-crossing in the second derivative in the direction of the gradient, yet the estimation of gradient direction is not extensively discussed. The extension of that work to account for the gradient direction estimation is identified as a future research direction.

In this chapter we formulate a general minimax optimization problem for gradient operator design. The gradient direction estimation plays a central role in this formulation. By the characterization on the relationship between the quality of gradient

magnitude and direction estimation, the quality of the magnitude estimate will also be good when the orientation estimation performance is optimized.

4.1.3 Organization of the chapter

We start our discussion by considering the performance characterization of gradient operators, i.e., to answer the question: what are the desirable properties that an optimal gradient operator should possess? The question is answered in Section 4.2 within a general framework for the performance characterization of gradient operators. The resulting analytic expressions for the bias and variance of the gradient orientation estimate is used in Section 4.3 to formulate the optimal gradient operator design problem as a minimax optimization problem. Three classes of gradient operators are studied using this formulation in Sections 4.4, 4.5 and 4.6. Their performance is compared with the optimal operator obtained as the numerical solution to the minimax problem in Section 4.7. The effect of image blurring on the gradient operator performance is studied in Section 4.9.

4.2 Gradient operator performance characterization

The most popular gradient estimators are linear, shift-invariant operators that can finally reduce to a single mask or kernel. Convolving (or correlating) this mask with the image data gives the element of the gradient along the, say, row axis. The transpose of this mask is used for the column axis. This broad class of operators includes the most commonly used ones such as the the dG filter used in Canny's edge operator [19], the integrated directional derivative operator [163, 60], and the Sobel, Prewitt, Roberts operators [60]. Although different kernels are used by the different operators, the kernels share the same features: the elements sum to zero, and the mask is orthogonal to its transpose. Here we characterize this class of operators in terms of its output signal-to-noise ratio on the gradient magnitude estimate, and the accuracy

and precision and on the gradient direction estimate.

4.2.1 Notation

Let the observed image intensity values in the neighborhood being considered be denoted by J . It is a noisy observation of the underlying noise-free image intensity function J_f :

$$J = J_f + \eta \quad (4.3)$$

where η is the additive zero-mean white Gaussian noise with variance σ^2 , i.e., $\eta \sim n(0, \sigma^2 I^{N \times N})$ where $I^{N \times N}$ is the identity matrix, and N is the number of pixels in the neighborhood.

Let the row and column derivative masks be denoted by h_r and h_c , respectively. In the two-dimensional representation as convolution masks, they are the transpose of each other. The entries in h_r and h_c are constants that are independent of J_f and η . We assume that the sum of entries in h_r is zero, and h_r is orthogonal to h_c , i.e., the sum of the products of their corresponding elements is zero. This is true for virtually all linear, shift-invariant gradient estimators.

Let the values in J , J_f , η , h_r and h_c all be organized into column vectors following the same rule, say, the row-major order. Now h_r and h_c are no longer transpose of each other, but the values in one is just a permutation of those in the other. Also, the zero-sum and orthogonality conditions still holds.

$$\langle h_r, \mathbf{1} \rangle = \langle h_c, \mathbf{1} \rangle = 0 \quad (4.4)$$

$$\langle h_r, h_c \rangle = 0 \quad (4.5)$$

where $\mathbf{1}$ is an $N \times 1$ vector with all ones, and $\langle ., . \rangle$ stands for taking the inner product of two vectors. The sum of squared values in the row or column mask is denoted by λ^2 .

$$\lambda^2 = \langle h_r, h_r \rangle = \langle h_c, h_c \rangle \quad (4.6)$$

The noise-free and noisy estimates of the row and column derivatives are

$$\mu_1 = \langle h_r, J_f \rangle$$

$$\mu_2 = \langle h_c, J_f \rangle$$

$$D_1 = \langle h_r, J \rangle$$

$$D_2 = \langle h_c, J \rangle$$

4.2.2 Basic characterization

Given any row kernel h_r , the column kernel h_c is uniquely determined. h_r and h_c , and hence λ^2 as determined by Equation 4.6, are fully determined constants.

μ_1 and μ_2 depend only on the underlying noise-free image intensity J_f and the chosen gradient operator mask h_r , and are therefore not dependent on the observation noise.

D_1 and D_2 , which are noisy estimates of μ_1 and μ_2 based on the noisy observation J , are normal random variables. Since h_r and h_c are orthogonal to each other, D_1 and D_2 are independent normal random variables.

$$D_1 \sim n(\mu_1, \lambda^2 \sigma^2), D_2 \sim n(\mu_2, \lambda^2 \sigma^2)$$

4.2.3 Gradient estimator

The gradient estimate is given by the magnitude and orientation estimates

$$G = \sqrt{D_1^2 + D_2^2}$$

$$\theta = \text{atan2}(D_1, D_2)$$

where $\text{atan2}(D_1, D_2)$ computes the inverse tangent function for $\frac{D_1}{D_2}$ with the result in the full angular range of $(-\pi, \pi]$ so that $D_1 = G \sin \theta$, $D_2 = G \cos \theta$.

From the characterization on D_1 and D_2 , we know $G^2 = D_1^2 + D_2^2$ is related to the non-central chi-squared distribution with 2 degrees of freedom and non-centrality

parameter $\frac{\mu_1^2 + \mu_2^2}{\lambda^2 \sigma^2}$.

$$\frac{G^2}{\lambda^2 \sigma^2} \sim \chi^2_{2, \frac{\mu_1^2 + \mu_2^2}{\lambda^2 \sigma^2}} \quad (4.7)$$

Let the non-centrality parameter be denoted by κ^2 . It reflects the relative amount of signal and noise energy in the gradient operator output. We define as the signal-to-noise ratio of the gradient operator output

$$\kappa = \sqrt{\frac{\mu_1^2 + \mu_2^2}{\lambda^2 \sigma^2}}. \quad (4.8)$$

The bigger κ is, the better quality the estimate of the gradient magnitude is of. As will become clear later, a bigger κ also gives better precision measure for the gradient orientation estimate. Therefore, the quality of the magnitude estimate and the precision of the orientation estimate improve simultaneously when κ is increased.

Gradient magnitude and input SNR estimation

From the properties of the non-central chi-squared distribution, the mean value of G^2 is $(\kappa + 2)\lambda^2 \sigma^2$. In real applications, the true noise variance σ^2 is not known and needs to be estimated from image data. By the cubic facet model [60, 163], the estimate is given by

$$\hat{\sigma}^2 = \frac{\epsilon^2}{N - 10}, \quad (4.9)$$

where ϵ^2 is the sum of the squared fitting error of the cubic facet fit to the image neighborhood data and N is the number of pixels used in the fit. ($N = 25$ for the 5×5 neighborhood.) The reason for this noise variance estimator is that, under the assumed noise model, ϵ^2/σ^2 follows the chi-squared distribution with $N - 10$ degrees of freedom. Therefore, the mean value of ϵ^2 is $(N - 10)\sigma^2$.

Under the assumed noise and gradient estimation models, the estimates D_1 and D_2 are independent normal random variables. Furthermore, they are also independent of the noise variance estimate $\hat{\sigma}^2$. Then the estimate

$$\hat{\kappa}^2 = \frac{D_1^2 + D_2^2}{\lambda^2 \hat{\sigma}^2} \quad (4.10)$$

is related to the non-central F distribution [68]. This is because $(D_1^2 + D_2^2)/(\lambda^2\sigma^2)$ follows the non-central chi-squared distribution with 2 degrees of freedom and non-centrality parameter $(\mu_1^2 + \mu_2^2)/(\lambda^2\sigma^2)$; $(N - 10)\hat{\sigma}^2/\sigma^2 = \epsilon^2/\sigma^2$ follows the (central) chi-squared distribution with $N - 10$ degrees of freedom. Therefore,

$$\frac{\hat{\kappa}^2}{2} \sim F_{v_1, v_2}(\xi)$$

where $v_1 = 2$ and $v_2 = N - 10$ are the degrees of freedom, and $\xi = \frac{\mu_1^2 + \mu_2^2}{\lambda^2\sigma^2}$ is the non-centrality parameter. The mean value of the $F_{v_1, v_2}(\xi)$ distribution is $\frac{v_2(v_1 + \xi)}{v_1(v_2 - 2)}$ for $v_2 > 2$ [68].

In real applications, the non-centrality parameter ξ is not known. By equating $\frac{\hat{\kappa}^2}{2}$ with the mean value of $F_{v_1, v_2}(\xi)$, we obtain an estimate for it. In the 5×5 neighborhood, this is

$$\hat{\xi} = \frac{13}{15}\hat{\kappa}^2 - 2$$

for $\frac{13}{15}\hat{\kappa}^2 \geq 2$, and $\hat{\xi} = 0$ otherwise. Using the relationship $\xi = \frac{\mu_1^2 + \mu_2^2}{\lambda^2\sigma^2}$, the new estimate for the squared gradient magnitude is then

$$\hat{G}^2 = (\mu_1^2 + \mu_2^2) = \hat{\xi}\lambda^2\hat{\sigma}^2 = (\frac{13}{15}\hat{\kappa}^2 - 2)\lambda^2\hat{\sigma}^2, \quad (4.11)$$

where $\hat{\kappa}^2$ is estimated by Equation (4.10).

The input SNR, denoted by Ω , is defined to be the ratio of the true contrast of the sharp step edge over the standard deviation of the input additive noise. The edge contrast is defined to be the difference in the intensity values between the two sides of the sharp step edge. Borrowing the definition of κ_0 (a computable constant determined by the gradient operator kernel) in Equation (4.16) and the relationship between κ and κ_0 in Equation (4.17), we have

$$\hat{\Omega}_{in} = \sqrt{\frac{\hat{G}^2}{\kappa_0^2\lambda^2\hat{\sigma}^2}} = \sqrt{\frac{\frac{13}{15}\hat{\kappa}^2 - 2}{\kappa_0^2}} = \frac{1}{\kappa_0}\sqrt{\frac{13}{15} \cdot \frac{D_1^2 + D_2^2}{\lambda^2\hat{\sigma}^2} - 2}. \quad (4.12)$$

Notice that the estimate is 0 for $\frac{13}{15}\hat{\kappa}^2 < 2$.

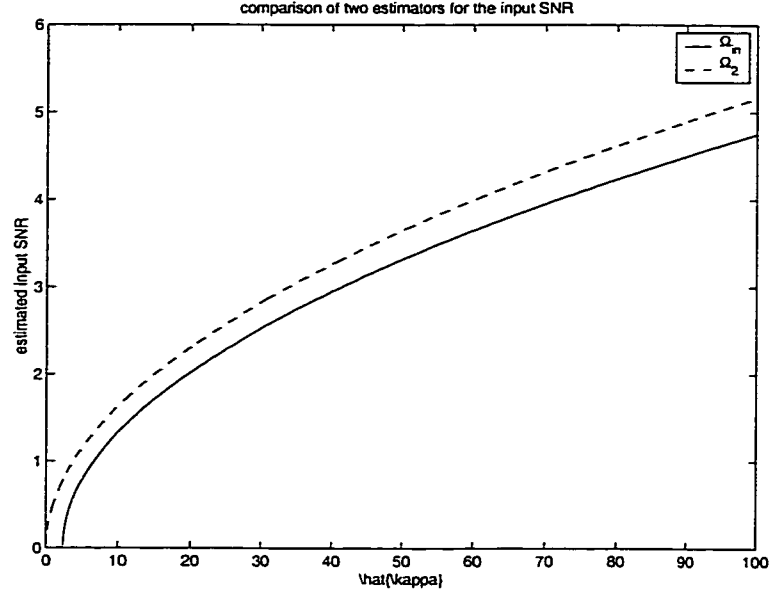


Figure 4.1: Comparison of two estimators for the input SNR. The biased estimate $\hat{\Omega}_2$ of Equation (4.13) is shown by the dashed curve which over-estimates the input SNR. The adjusted estimate $\hat{\Omega}_{in}$ of Equation (4.12) is shown in the solid curve. It is based on the properties of the non-central F distribution, and is a more accurate estimate.

If \hat{G}^2 were simply estimated by $D_1^2 + D_2^2 = \lambda^2 \hat{\sigma}^2 \hat{\kappa}^2$, the resulting estimate of the input SNR would be

$$\hat{\Omega}_2 = \sqrt{\frac{\lambda^2 \hat{\sigma}^2 \hat{\kappa}^2}{\lambda^2 \hat{\sigma}^2 \kappa_0^2}} = \frac{\hat{\kappa}}{\kappa_0} = \frac{1}{\kappa_0} \sqrt{\frac{D_1^2 + D_2^2}{\lambda^2 \hat{\sigma}^2}}, \quad (4.13)$$

which is an over-estimate. For the IDDGO (see Section 4.5) in the 5×5 neighborhood, $\kappa_0 = 1.95$. The two estimates given by the above two formulas are plotted in Figure 4.1.

Gradient orientation estimation

The distribution of the orientation estimate

$$\theta = \text{atan2}(D_1, D_2)$$

where $D_1 \sim n(\mu_1, \lambda^2 \sigma^2)$ and $D_2 \sim n(\mu_2, \lambda^2 \sigma^2)$ can be derived as follows [96].

The joint distribution for D_1 and D_2 is

$$f_{D_1, D_2}(d_1, d_2) = \frac{1}{2\pi\lambda^2\sigma^2} \exp\left(-\frac{(d_1 - \mu_1)^2 + (d_2 - \mu_2)^2}{2\lambda^2\sigma^2}\right).$$

By the transformation

$$D_1 = G \sin \theta, \quad D_2 = G \cos \theta,$$

the joint distribution for gradient magnitude G and orientation θ is obtained. The marginal distribution for θ is obtained by integrating out the gradient magnitude from the joint distribution. The result is the pdf

$$f_{\Theta}(\theta) = e^{-\frac{\mu_1^2 + \mu_2^2}{2\lambda^2\sigma^2}} \left\{ \frac{1}{2\pi} + \frac{\sqrt{\mu_1^2 + \mu_2^2} \cos(\theta - \theta_0)}{\sqrt{2\pi}\lambda\sigma} \exp\left[\frac{(\mu_1^2 + \mu_2^2) \cos^2(\theta - \theta_0)}{2\lambda^2\sigma^2}\right] \cdot \left[\frac{1}{2} + \frac{1}{2} \operatorname{erf}\left(\frac{\sqrt{\mu_1^2 + \mu_2^2} \cos(\theta - \theta_0)}{\sqrt{2}\lambda\sigma}\right) \right] \right\},$$

where $\theta_0 = \operatorname{atan2}(\mu_1, \mu_2)$ is the true gradient orientation. Using

$$\kappa = \sqrt{\frac{\mu_1^2 + \mu_2^2}{\lambda^2\sigma^2}},$$

the pdf for the orientation estimate is

$$f_{\Theta}(\theta|\theta_0, \kappa) = e^{-\frac{\kappa^2}{2}} \left\{ \frac{1}{2\pi} + \frac{\kappa \cos(\theta - \theta_0)}{\sqrt{2\pi}} \exp\left[\frac{\kappa^2 \cos^2(\theta - \theta_0)}{2}\right] \cdot \left[\frac{1}{2} + \frac{1}{2} \operatorname{erf}\left(\frac{\kappa \cos(\theta - \theta_0)}{\sqrt{2}}\right) \right] \right\} \quad (4.14)$$

This is a special case of the offset normal distribution [97], with equal variance and no correlation, i.e., in the notation of [97], $\sigma_1 = \sigma_2 = \sigma$ and $\rho = 0$.

It has been pointed out [97, 123, 122] that a conditional distribution of the orientation estimate is the von Mises distribution $M(\theta_0, \kappa^2)$ [97]. This condition is $D_1^2 + D_2^2 = \mu_1^2 + \mu_2^2$, which is in general not satisfied for low SNR, and approximately satisfied for very high SNR. Although the gradient magnitude estimation is closely coupled with the orientation estimation, our study here mainly concerns the marginal

distribution of the orientation estimate, not that conditional distribution. Therefore, the special case of the offset normal distribution in Equation (4.14), which is referred to as the orientation distribution in this dissertation, is the appropriate model for the orientation estimate. We are not further concerned with the von Mises distribution.

θ_0 is the mean value of $f_{\Theta}(\theta|\theta_0, \kappa)$, and κ is the concentration parameter, a term borrowed from the von Mises distribution. If we shift the domain of the distribution to be symmetric about θ_0 , the distribution is unimodal and is symmetrical about the mode, which is at θ_0 . The anti-mode is at $\theta = \theta_0 + \pi$. The distribution reduces to the uniform distribution $u(0, 2\pi)$ for $\kappa = 0$. When κ becomes larger, the pdf becomes taller and thinner around θ_0 . See Figure 4.2(a) for some sample plots of this pdf.

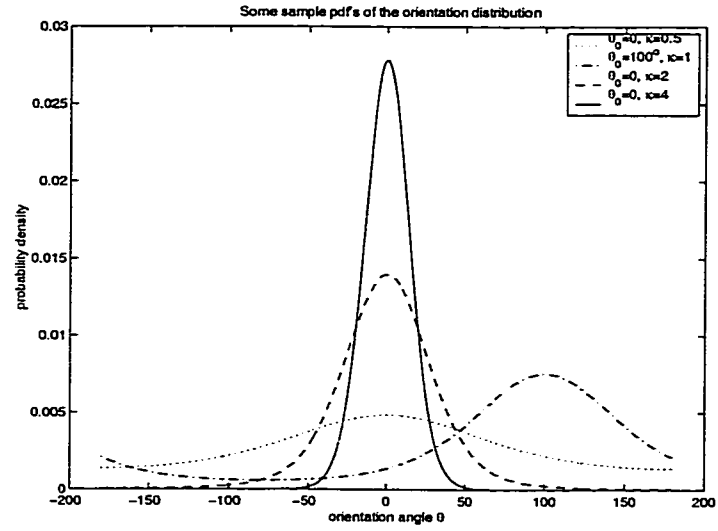
It is obvious that the mean of the distribution does not depend on the concentration parameter κ . Although not obvious from the expression of the pdf, it is observed that κ strongly affects the concentration of the distribution. As a general rule, the bigger κ is, the smaller the spread of the distribution is. The analytic expression for the standard deviation is difficult to obtain. However, for any given κ , the value of the variance defined as

$$\int_{\theta_0-\pi}^{\theta_0+\pi} f_{\Theta}(\theta|\theta_0, \kappa)(\theta - \theta_0)^2 d\theta = \int_{-\pi}^{\pi} f_{\Theta}(\theta|0, \kappa)\theta^2 d\theta \quad (4.15)$$

can be evaluated numerically. The values of the variance for a range of κ of interest are pre-computed and tabulated. A plot of the variance as a function of κ is shown in Figure 4.2(b). The plot shows clearly that, for κ greater than 3, the variance of the orientation distribution can be closely approximated by $1/\kappa^2$.

4.2.4 Performance characterization for gradient direction estimators

For a given underlying noise-free image intensity function of interest, the mean values μ_1 and μ_2 for D_1 and D_2 are fully determined by any gradient operator being considered. The values are, in general, different for different operators. For any given operator, the bias of the orientation estimate is defined to be the difference between



(a) examples of the pdf

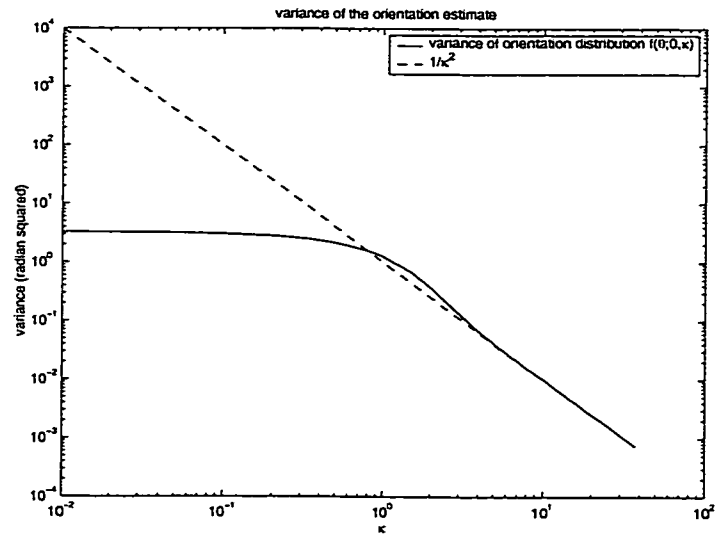
(b) variance as function of κ

Figure 4.2: Probability density and variance for the orientation distribution.

$\theta_0 = \text{atan2}(\mu_1, \mu_2)$ and the orientation of the true gradient of the intensity function. The standard deviation of the estimate characterizes the behavior of the operator in the presence of observation noise. The bias and standard deviation are the two components of the estimation error in the mean-squared sense. We will show that the standard deviation is inversely related to the concentration parameter $\kappa = \sqrt{\frac{\mu_1^2 + \mu_2^2}{\lambda^2 \sigma^2}}$.

Let the normalized output signal-to-noise ratio be denoted by κ_0 and defined by

$$\kappa_0 = \sqrt{\frac{\mu_{1,0}^2 + \mu_{2,0}^2}{\lambda^2}}, \quad (4.16)$$

where $\mu_{1,0}$ and $\mu_{2,0}$ are the values obtained by applying the row and column derivative kernels on the image data of the unit-contrast noise-free step edge (see below). This is a quantity fully determined by the chosen kernel and the noise-free image intensity values, and not affected by the noise level. It is a critical measure for the gradient operator performance. Under the general input SNR Ω , which is the ratio of the step edge contrast over the noise standard deviation, we have

$$\kappa = \kappa_0 \Omega \quad (4.17)$$

There are four major components in characterizing the performance of gradient direction estimators, namely, the operator as specified by its mask, the underlying noise-free image intensity function, the level of the observation noise, and the performance measures. From the first three, we compute λ^2 , μ_1 and μ_2 , and subsequently θ_0 and κ_0 . The final performance measures with respect to a certain noise level σ^2 , based on the mean and standard deviation of the orientation distribution with the parameters θ_0 and $\kappa = \kappa_0 \Omega$, can be easily computed numerically. Thus we characterize the gradient direction operator in terms of accuracy and precision with respect to the specified image intensity function and observation noise level. By the discussions leading to Equation (4.8), we know that the performance of the gradient magnitude estimate is monotonic with κ , hence is directly related to the precision of the direction estimate. Therefore we do not carry out a separate characterization for it.

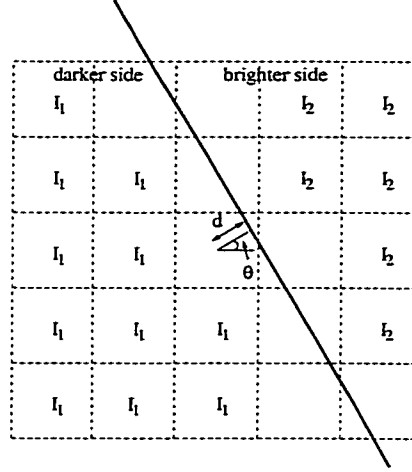


Figure 4.3: Illustration of the square-aperture sampling. The continuous step edge passes through the center pixel of the neighborhood. The orientation of the edge is θ and the displacement from the center of the center pixel is d . The darker and brighter sides of the edge have intensity values I_1 and I_2 , respectively.

In this work, we focus our attention on the image intensity data from the square-aperture sampling of the straight step edge. This intensity data was used in a number of studies in gradient estimation and edge detection [96, 163, 65]. This is well illustrated in Figure 4.3 which mimics the illustration in [96].

A step edge in any orientation θ passes through the center pixel of the neighborhood. It does not have to pass through the center of that pixel. The perpendicular distance from the center to the edge is the displacement d of the edge. The displacement d takes the value in the range $[-\frac{0.5}{\cos \theta}, \frac{0.5}{\cos \theta}]$.

When the step edge passes through a pixel and divides it into two parts having areas A_1 and A_2 with $A_1 + A_2 = 1$ (the unit being the area of a pixel), the intensity value for this pixel is $I = I_1 A_1 + I_2 A_2$ where I_1 and I_2 are the intensities on each side of the step edge. Pixels lying entirely on one side of the step edge have the intensity value for that side of the edge. (The contrast of the edge is therefore $|I_1 - I_2|$.) We only use the unit-contrast edge, i.e., intensity values $I_1 = -0.5$ and $I_2 = 0.5$ for the

two sides of the edge. Various signal to noise ratio is achieved by varying the noise variance. Here is an example of the noise-free unit-contrast step edge of 0° gradient orientation in the 5×5 neighborhood.

$$\begin{array}{ccccc}
 -0.5 & -0.5 & 0 & 0.5 & 0.5 \\
 -0.5 & -0.5 & 0 & 0.5 & 0.5 \\
 -0.5 & -0.5 & 0 & 0.5 & 0.5 \\
 -0.5 & -0.5 & 0 & 0.5 & 0.5 \\
 -0.5 & -0.5 & 0 & 0.5 & 0.5
 \end{array} \tag{4.18}$$

4.3 Optimal gradient operator design

In the previous analysis of gradient operator performance, we discussed the gradient orientation estimation error in two separate parts, i.e., the estimate bias and the random deviation. However, these two parts are not separately identifiable in real applications where the true gradient orientation is unknown. Therefore, the root-mean-square (RMS) error for the gradient orientation estimate is a more appropriate measure for the gradient operator performance in real applications. In the following, we formulate the optimal gradient operator design problem as a minimax problem where the worse-case RMS error needs to be minimized.

Let the row derivative mask of the gradient operator be denoted by h , which is obtained by arranging the elements in the mask in the row-major order into a column vector. Let the unit-contrast step edge with gradient orientation θ_0 and displacement d_0 be denoted by $e(\theta_0, d_0)$. $e_1(\theta_0, d_0)$ is a column vector with the elements of $e(\theta_0, d_0)$ in the row-major order, and $e_2(\theta_0, d_0)$ has them in the column-major order. Applying the row mask and column mask on $e(\theta_0, d_0)$ gives the two elements of the gradient estimate

$$\mu_1(h, \theta_0, d_0) = \langle h, e_1(\theta_0, d_0) \rangle \quad \mu_2(h, \theta_0, d_0) = \langle h, e_2(\theta_0, d_0) \rangle \tag{4.19}$$

where $\langle ., . \rangle$ stands for taking the inner product. Let the variance of the additive

white Gaussian noise be denoted by σ^2 . The input SNR defined as the ratio of the edge contrast over the noise standard deviation is $\Omega = \frac{1}{\sigma}$. The gradient orientation estimate given by the operator h is

$$\theta(h, \theta_0, d_0) = \text{atan2}(\mu_1(h, \theta_0, d_0), \mu_2(h, \theta_0, d_0)) \quad (4.20)$$

and the variance of the estimate is controlled by the concentration parameter

$$\kappa(h, \theta_0, d_0) = \sqrt{\frac{\mu_1(h, \theta_0, d_0)^2 + \mu_2(h, \theta_0, d_0)^2}{\langle h, h \rangle \sigma^2}} \quad (4.21)$$

$$= \sqrt{\frac{\mu_1(h, \theta_0, d_0)^2 + \mu_2(h, \theta_0, d_0)^2}{\langle h, h \rangle}} \Omega^2 \quad (4.22)$$

$$= \kappa_0(h, \theta_0, d_0) \Omega \quad (4.23)$$

where $\kappa_0(h, \theta_0, d_0) = \sqrt{\frac{\mu_1(h, \theta_0, d_0)^2 + \mu_2(h, \theta_0, d_0)^2}{\langle h, h \rangle}}$ is referred to as the SNR gain of the operator h for the step edge with orientation θ_0 and displacement d_0 .

The worst-case mean-squared error (MSE) in the gradient orientation estimate is

$$\epsilon(h) = \max_{\theta_0 \in [0, 2\pi]} \left\{ \max_{d_0 \in [-D(\theta_0), D(\theta_0)]} \left\{ (\theta(h, \theta_0, d_0) - \theta_0)^2 + \int_{-\pi}^{\pi} t^2 f(t; \kappa(h, \theta_0, d_0)) dt \right\} \right\} \quad (4.24)$$

where $D(\theta_0)$ specifies the range of the possible edge displacement, and

$$f(t; \kappa) = f_{\Theta}(t; 0, \kappa)$$

is the pdf of the orientation distribution (Equation (4.14)) with mean value of zero.

Note that the angular range of $[0, \frac{\pi}{4}]$ is enough for the range of θ_0 in Equation (4.24). This is due to the symmetry in the gradient orientation estimation. For this range, $D(\theta_0) = \frac{\sqrt{2}}{2} \cos(\frac{\pi}{4} - \theta_0) = 0.5(\cos \theta_0 + \sin \theta_0)$, which allows the step discontinuity to pass anywhere through the center pixel of the neighborhood.

4.3.1 Optimal gradient operator design — problem statement

For a given input signal-to-noise ratio Ω , determine the real-valued elements of h to minimize $\epsilon(h)$ in Equation (4.24).

4.3.2 Issues in solving the optimal gradient operator design problem

Trying to analytically solve the problem just stated is extremely difficult, if not impossible, by ordinary analytic methods of solving optimization problems. We consider solving the problem using numerical techniques.

The dimensionality of the variable in numerical optimization is a very important issue. For moderate neighborhood sizes, the number of elements in h is large. For example, for the 5×5 neighborhood, there are 25 numbers in h . However, we know they do not have to be independent variables in the optimization process. From practical experience, we know the row derivative mask of the gradient operators should be symmetric about the center column, and anti-symmetric about the center row. For the 5×5 neighborhood, the mask has the structure

$$\begin{pmatrix} -a_1 & -a_2 & -a_3 & -a_2 & -a_1 \\ -a_4 & -a_5 & -a_6 & -a_5 & -a_4 \\ 0 & 0 & 0 & 0 & 0 \\ a_4 & a_5 & a_6 & a_5 & a_4 \\ a_1 & a_2 & a_3 & a_2 & a_1 \end{pmatrix} \quad (4.25)$$

Therefore, we can reduce the number of free variables to 6. If we want three digits of fixed-point precision for the variables, they can take values in the set of integer numbers -1000 through 1000. The exhaustive brute force search of the parameter space needs more than 10^{18} computations of $\epsilon(h)$. Therefore, such computation is not feasible on current computers. Instead, some numerical search algorithm has to be used. In order to reduce the chance of being trapped in local minima, the initial masks should use good operators such as the IDDGO, extended Sobel, and dG. Also, multiple runs should be attempted with different initial masks.

Another way of reducing the search space is to impose some more structure on the mask, letting it to take some parametric form with a smaller number of free parameters to be determined. Examples for this are the IDDGO, dG, and Sobel-class operators, each of which has only one free parameter, namely the integration

domain size L for the IDDGO, the smoothing factor s for the dG operator, K_3 and K_5 for the 3×3 and 5×5 Sobel operator. Exhaustive search over their possible ranges yields the best operators within their respective family of operators. Since most readers are more familiar with the simpler Sobel-class gradient operators, we discuss them first in Section 4.4. Some general observations made there also applies to the discussions on the IDDGO and dG operators, which are presented in Sections 4.5 and 4.6, respectively.

4.4 Optimizing the Sobel-class gradient operator

The row derivative masks of the 3×3 and 5×5 Sobel-class gradient operators have the form [65]

$$\begin{pmatrix} -1 & -K_3 & -1 \\ 0 & 0 & 0 \\ 1 & K_3 & 1 \end{pmatrix} \quad (4.26)$$

and

$$\begin{pmatrix} -0.5 & -0.8 & -1 & -0.8 & -0.5 \\ -0.4 & -1 & -K_5 & -1 & -0.4 \\ 0 & 0 & 0 & 0 & 0 \\ 0.4 & 1 & K_5 & 1 & 0.4 \\ 0.5 & 0.8 & 1 & 0.8 & 0.5 \end{pmatrix} \quad (4.27)$$

In the following, we decide the optimal values for K_3 and K_5 . As expected, these values change with the input SNR.

Notice that the general form of the optimal gradient operator in the 3×3 neighborhood has the form

$$\begin{pmatrix} -a_1 & -a_2 & -a_1 \\ 0 & 0 & 0 \\ a_1 & a_2 & a_1 \end{pmatrix} \quad (4.28)$$

Since a positive multiplicative constant does not matter in the gradient orientation estimation, this form is equivalent to the 3×3 Sobel-class operator with $a_1 = 1$ and

$a_2/a_1 = K_3$. Therefore, by any optimality criterion, the optimal member of the 3×3 Sobel-class operators is exactly the optimal 3×3 gradient operator.

In our experiment, to achieve the numerical precision of two to three digits, we choose $a_1 = 100$ and vary a_2 in the range of integer numbers from -1000 to 1000. The optimal value is that which gives the smallest value for the cost function given in Equation (4.24). For each input SNR value of interest, the optimal values for K_3 in the Sobel-class operators are obtained and shown in Figure 4.4(a). Shown in the same figure are the curves for the SNR gain κ_0 and the absolute value of the corresponding bias component of the error for the resulting optimal operator. As can be seen clearly, with the increase of input SNR, the optimal operator reacts with a decreasing κ_0 . This is the reasonable behavior — when the SNR is high, less capability in noise suppression is needed and more emphasis should be put on reducing the bias. The achieved minimax RMS orientation estimation error is shown as the dashed curve in Figure 4.4(b).

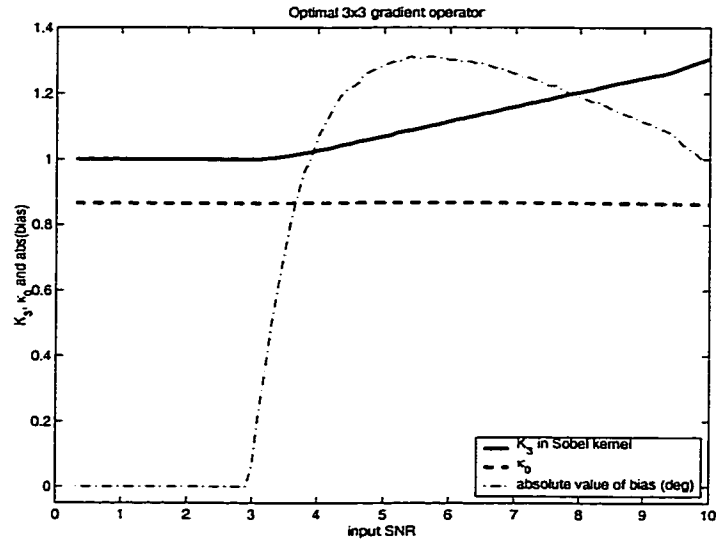
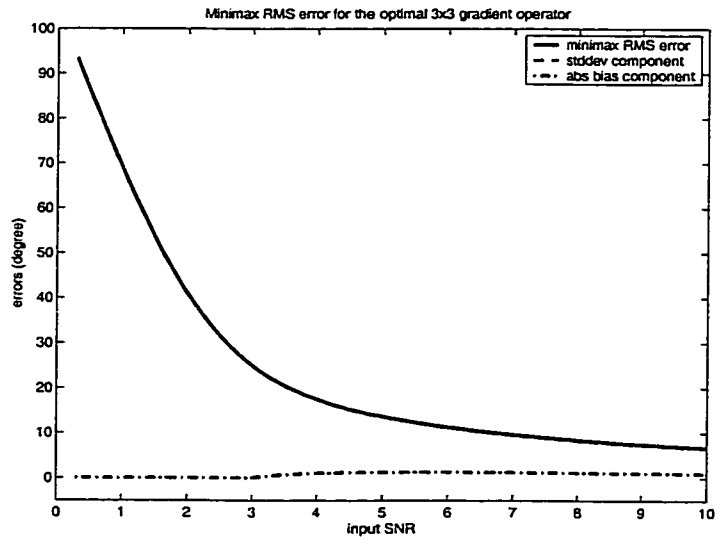
There are, however, several interesting behaviors in the plots that deserve some more discussion. First, we observe that, for a low SNR range of less than 3, the optimal value for K_3 remains a constant value of about 1.0. This is a desirable robustness property of a gradient operator with respect to the input SNR, because it eliminates the need for automatic parameter selection, which is important but often difficult. However, this robustness is weakened for higher input SNR values, as the optimal K_3 starts to change with the SNR. This pattern of behavior is also observed for the 5×5 Sobel-class operator and the 5×5 dG-class operator.

Secondly, we observe that the optimally chosen member of the Sobel-class operator has zero bias component in the minimized cost function (the minimax error) for the input SNR less than 3. For higher SNR values, the optimal operator no longer keeps a zero bias component. However, it achieves a larger value for the SNR gain κ_0 , therefore, having more power in noise reduction. The reason for such sudden presence of a bias component is related to the change in the relative importance of

the two terms in the cost function, i.e., the bias term and the random deviation term in Equation (4.24). Remember that the random deviation term is a monotonically decreasing function of $\kappa_0\Omega^2$ where Ω is the input SNR. At a given low SNR value, the random deviation term dominates the cost function, and it is very sensitive to any change in κ_0 . The edge pattern for the 45° orientation and $\sqrt{2}/2$ translation (see last section) has the smallest κ_0 among all orientations and translations considered. This is because that the effective signal energy in the 3×3 neighborhood for this edge pattern is the smallest. This smallest κ_0 yields the largest value for the random deviation term in the cost function. For edge patterns with other orientations and translations, although the bias term might be non-zero, the increase in κ_0 is enough to reduce the random deviation term by a magnitude that is more than the magnitude of the bias term. Therefore, the worst-case error occurs at 45° which has zero bias.

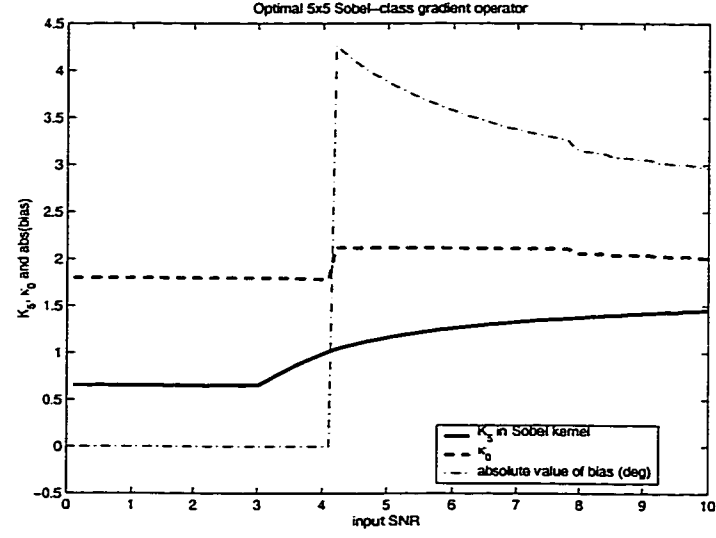
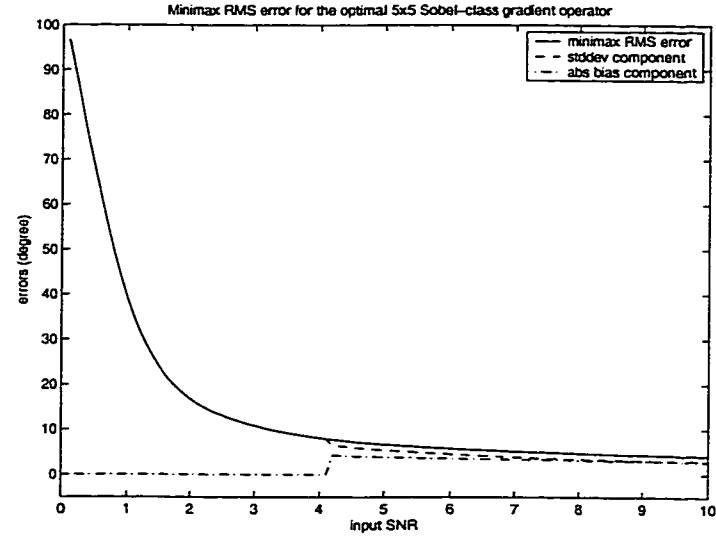
As the input SNR increases, for any given edge pattern, the magnitude of the random deviation term reduces quickly, while the bias term remains constant. Therefore, the bias term quickly gains importance. When the bias term is greater than the difference in the random deviation term due to two different κ_0 values from two different edge patterns, the total error for the 45° edge pattern will become smaller than that for some other edge pattern which has a non-zero bias term. At this time, the worst-case error occurs for an edge pattern with a non-zero bias and a larger κ_0 , causing the sudden presence of bias and the associated sudden increase in κ_0 in Figure 4.4. Notice that this behavior is common to all other gradient operator classes studied in this chapter. The same explanation applies to those operators, too.

The behavior of the optimal member of the 5×5 Sobel-class operators are shown in Figure 4.5. Notice also the behavior of a constant K_5 for SNR less than 3, and the absence and presence of a bias component in the minimax error for SNR ranges less than and greater than 4.1, respectively. From this, we conclude that it is just a coincidence for the optimal 3×3 Sobel-class operator to have the same input SNR of 3 for the two behaviors discussed above.

(a) K_3 in Sobel operator and SNR gain κ_0 

(b) achieved minimax RMS error

Figure 4.4: K_3 and corresponding SNR gain κ_0 for the optimal 3×3 Sobel-class operator, and the achieved minimax RMS error with its two components.

(a) K_5 in Sobel operator and SNR gain κ_0 

(b) achieved minimax RMS error

Figure 4.5: K_5 and corresponding SNR gain κ_0 for the optimal 5×5 Sobel-class operator, and the achieved minimax RMS error with its two components.

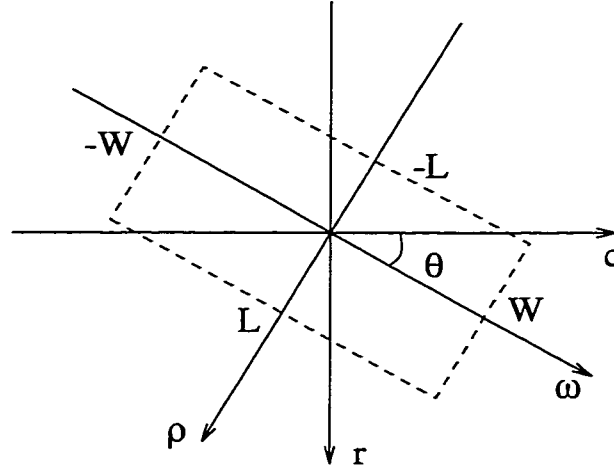


Figure 4.6: Illustration of the IDDGO. The integration of the directional derivative is done within the dashed line denoted rectangular area.

4.5 Optimizing the IDDGO

4.5.1 The integrated directional derivative gradient operator

The IDDGO was introduced by Zuniga and Haralick [163] to improve the gradient orientation estimation quality of the facet model based operator. The idea is illustrated in Figure 4.6. When estimating the gradient vector from the least-squares fit by a bi-variate cubic polynomial to the observed intensity function, the average (integration) of the directional derivative over a rectangular neighborhood is used. For a direction θ of interest, the boundaries of this rectangular neighborhood are parallel and normal to θ . The directional derivative of the estimated intensity function along direction θ is averaged (integrated) over the neighborhood of size $2W \times 2L$. The direction along which this average is maximized is the estimate for the gradient direction, and is denoted by θ_{MAX} .

In most computer vision applications involving gradient estimation, square neighborhoods are used. Here we are only interested in square neighborhoods and square

integration domain ($L = W$) for the IDDGO.

The integral of the directional derivative in direction θ is given by [163]

$$F_\theta = \frac{1}{4L^2} \int_{-L}^L \int_{-L}^L f'_\theta(\rho \cos \theta + \omega \sin \theta, -\rho \sin \theta + \omega \cos \theta) d\rho d\omega \quad (4.29)$$

where $f'_\theta(\cdot)$ is the first directional derivative of the image intensity function $f(\cdot)$ in the direction θ . L specifies the size of the integration domain.

The gradient estimate is [163]

$$G = F_{\theta_{MAX}} u_{\theta_{MAX}} \quad (4.30)$$

where $u_{\theta_{MAX}}$ is the unit vector in the direction along which the integration of the directional derivative achieves its maximum value $F_{\theta_{MAX}}$.

The solution is found to be

$$\theta_{MAX} = \tan^{-1} \frac{D_1}{D_2} \quad (4.31)$$

$$F_{\theta_{MAX}} = \sqrt{D_1^2 + D_2^2} \quad (4.32)$$

where

$$D_1 = L^2 K_7 + \frac{1}{3} L^2 K_9 + K_2 \quad (4.33)$$

$$D_2 = L^2 K_{10} + \frac{1}{3} L^2 K_8 + K_3 \quad (4.34)$$

The $\{K_i | i = 2, 3, 7, 8, 9, 10\}$ coefficients are from the cubic facet model parameters [163, 60]. In this model, the underlying intensity function is assumed be a bivariate cubic polynomial

$$f(r, c) = K_1 + K_2 r + K_3 c + K_4 r^2 + K_5 r c + K_6 c^2 + K_7 r^3 + K_8 r^2 c + K_9 r c^2 + K_{10} c^3 \quad (4.35)$$

The coefficients (K 's) can be estimated by a least-square-error surface fit through the discrete orthogonal polynomials (DOP) [60]. For any given neighborhood size and true gradient direction θ^* , the values of the coefficients can be computed exactly.

Alternatively, the row derivative kernel can be obtained from the kernels for the cubic facet model parameters. For the 5×5 neighborhood, the row derivative kernel is

$$\frac{1}{420} \begin{pmatrix} -11L^2 + 31 & -5L^2 - 5 & -3L^2 - 17 & -5L^2 - 5 & -11L^2 + 31 \\ 12L^2 - 44 & 15L^2 - 62 & 16L^2 - 68 & 15L^2 - 62 & 12L^2 - 44 \\ 0 & 0 & 0 & 0 & 0 \\ -12L^2 + 44 & -15L^2 + 62 & -16L^2 + 68 & -15L^2 + 62 & -12L^2 + 44 \\ 11L^2 - 31 & 5L^2 + 5 & 3L^2 + 17 & 5L^2 + 5 & 11L^2 - 31 \end{pmatrix}$$

The column derivative kernel is just the transpose. The sum of squared entries is

$$\lambda^2 = \frac{1}{2520}(37L^4 - 262L^2 + 527) \quad (4.36)$$

The determinant of the right-hand side of the above equation as a quadratic function of L^2 is less than 0. Therefore, λ^2 is always positive for all values of L .

4.5.2 Mean and variance of the IDDGO direction estimate

When noise is present in the observed intensity function, the facet model coefficients are random variables. The gradient direction estimate as given in Equation (4.31) is a function of the coefficients, and hence is also a random variable. We are interested in the mean-squared error of its deviation from the true direction value.

When the noise can be modeled as additive white Gaussian, the facet model coefficients are multivariate normal random variables. The mean values depend on the true underlying intensity function, and the covariance matrix can be estimated. In particular, $\{K_2, K_7, K_9\}$ and $\{K_3, K_{10}, K_8\}$ are two correlation groups, and there is no correlation between the two groups. The two groups have the same covariance matrix.

$$\begin{pmatrix} \sigma_{K_2}^2 & \sigma_{K_2, K_7}^2 & \sigma_{K_2, K_9}^2 \\ \sigma_{K_7, K_2}^2 & \sigma_{K_7}^2 & \sigma_{K_7, K_9}^2 \\ \sigma_{K_9, K_2}^2 & \sigma_{K_9, K_7}^2 & \sigma_{K_9}^2 \end{pmatrix} = \begin{pmatrix} \sigma_{K_3}^2 & \sigma_{K_3, K_{10}}^2 & \sigma_{K_3, K_8}^2 \\ \sigma_{K_{10}, K_3}^2 & \sigma_{K_{10}}^2 & \sigma_{K_{10}, K_8}^2 \\ \sigma_{K_8, K_3}^2 & \sigma_{K_8, K_{10}}^2 & \sigma_{K_8}^2 \end{pmatrix} \quad (4.37)$$

For the 5×5 neighborhood, the values are

$$\begin{pmatrix} 0.2091 & -0.0472 & -0.0143 \\ -0.0472 & 0.0139 & 0 \\ -0.0143 & 0 & 0.0071 \end{pmatrix} \sigma^2 \quad (4.38)$$

where σ^2 is the variance of the additive noise, and can be estimated from the residual fitting error (See Section 4.2). As two groups of multivariate normal random variables, $\{K_2, K_7, K_9\}$ and $\{K_3, K_8, K_{10}\}$ are independent since they are uncorrelated. Therefore D_1 and D_2 as given in Equation (4.33) and (4.34) are independent normal random variables.

The mean values of D_1 and D_2 depend on the underlying noise-free intensity function and the neighborhood size over which the cubic facet parameters are calculated. D_1 and D_2 have the same variance $\lambda^2 \sigma^2$, where λ^2 is a constant depending only on the neighborhood size and L . As a matter of fact, λ^2 is the sum of the squared elements in the equivalent kernel for computing D_1 or D_2 .

$$\lambda^2 \sigma^2 = \left(\sigma_{K_7}^2 + \frac{1}{9} \sigma_{K_9}^2 + \frac{2}{3} \sigma_{K_7, K_9}^2 \right) L^4 + \left(2 \sigma_{K_2, K_7}^2 + \frac{2}{3} \sigma_{K_2, K_9}^2 \right) L^2 + \sigma_{K_2}^2 \quad (4.39)$$

For the 5×5 neighborhood, $\lambda^2 = 0.0147L^4 - 0.1039L^2 + 0.2091$ — the same result as Equation (4.36) obtained by using the IDDGO row derivative kernel at the end of Section 4.5.1. For large neighborhood sizes, finding the analytic expression, such as the one in Equation (4.36), of the IDDGO row derivative kernel in terms of L can be rather tedious. On the other hand, the covariance matrix of the facet parameters can easily be computed numerically. Therefore, it is preferable to use the Equation (4.39) for computing λ^2 .

4.5.3 Finding the optimal integration domain size

Now that the expressions for all terms involved in Equation (4.24) are determined for the IDDGO, the optimal gradient operator design problem constrained in the IDDGO

class operators reduces to searching for the best value of L which yields the smallest value for Equation (4.24).

For the neighborhood size of 5×5 , the optimal value for L and the corresponding SNR gain κ_0 are plotted versus the input SNR level in Figure 4.7(a). We observe a nice property of the IDDGO in that over a large range of input SNR levels, the optimal integration domain size L only changes very slightly. This property provides the robustness of the operator since in real applications the input SNR cannot be estimated unreliably. The achieved minimax RMS error at the examined input SNR levels with its bias and standard deviation components is shown in Figure 4.7(b).

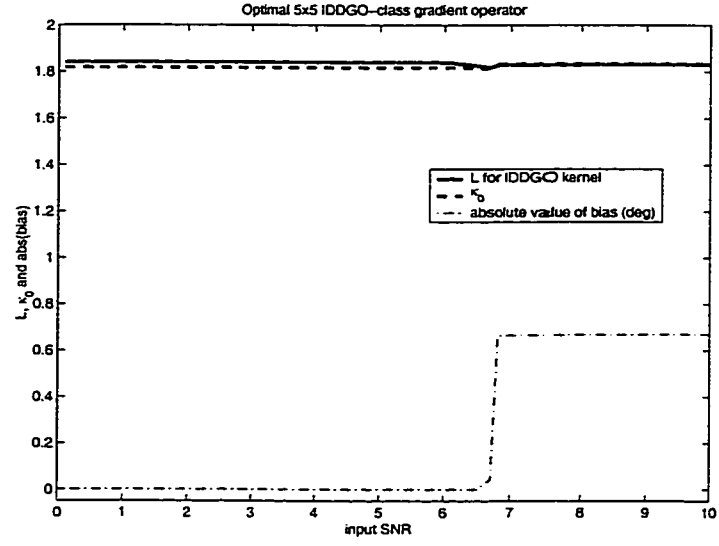
4.6 Optimizing the dG gradient operator

In his edge detector, Canny [19] uses what has become known as the dG operator for gradient estimation. This operator is the tensor product of a smoothing Gaussian along one direction with the derivative of that Gaussian along the perpendicular direction. The row-derivative kernel is expressed as

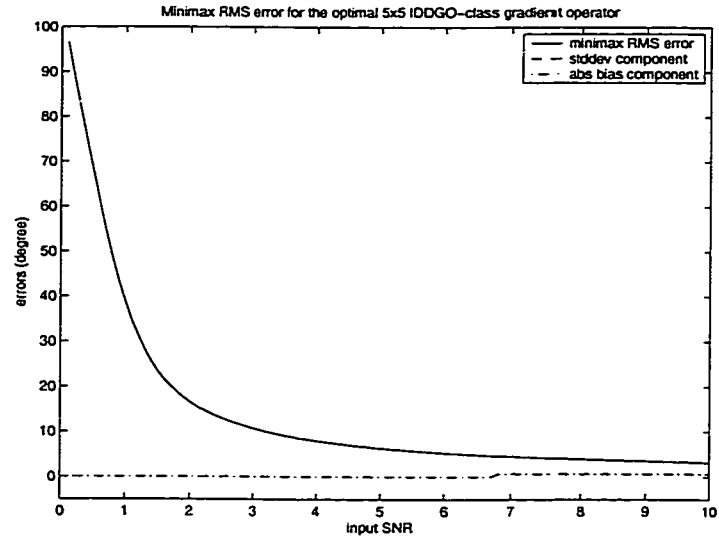
$$h_r = \frac{r}{2\pi s^4} e^{-\frac{r^2 + c^2}{2s^2}} \quad (4.40)$$

where $s > 0$ is the standard deviation of the smoothing Gaussian. The bigger s is, the greater the smoothing effect is.

In the 1-D formulation, the derivative of Gaussian enjoys many good properties. It approximates very closely to the numerically obtained optimal digital filter that maximizes Canny's edge criterion function. It is the exact solution to the alternative edge location criterion function proposed by Tagare *et al* [149]. However, there has not been extensive study as to the choice of the smoothing factor s of the smoothing Gaussian relative to the filter size. From the plot of the optimal digital filter in [19], the filter size seems to be big enough to hold most of the mass of the Gaussian. In practice, standard implementations of the Canny's edge detector make their local neighborhood cover four to six times the standard deviation of the center part of the



(a) integration domain size L and SNR gain κ_0 for the IDDGO



(b) achieved minimax RMS error

Figure 4.7: Integration domain size L and corresponding SNR gain κ_0 for the optimal 5×5 IDDGO, and the achieved minimax RMS error with its two components.

Gaussian. (In the continuous case, six times the standard deviation at the center part covers over 99.7% of the mass.)

The kernel for the dG operator is easily obtained by sampling the continuous filter in Equation (4.40). For the 5×5 neighborhood, this is

$$\frac{1}{2\pi s^4} \begin{pmatrix} -2u^8 & -2u^5 & -2u^4 & -2u^5 & -2u^8 \\ -u^5 & -u^2 & -u & -u^2 & -u^5 \\ 0 & 0 & 0 & 0 & 0 \\ u^5 & u^2 & u & u^2 & u^5 \\ 2u^8 & 2u^5 & 2u^4 & 2u^5 & 2u^8 \end{pmatrix} \quad (4.41)$$

where $u = e^{-\frac{1}{2s^2}}$. The sum of the squared elements is

$$\lambda^2 = \frac{2}{4\pi^2 s^8} (8u^{16} + 10u^{10} + 4u^8 + 2u^4 + u^2) \quad (4.42)$$

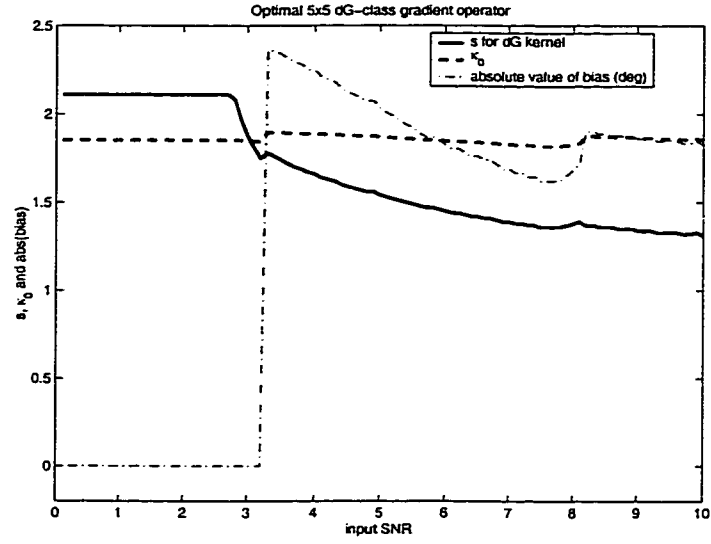
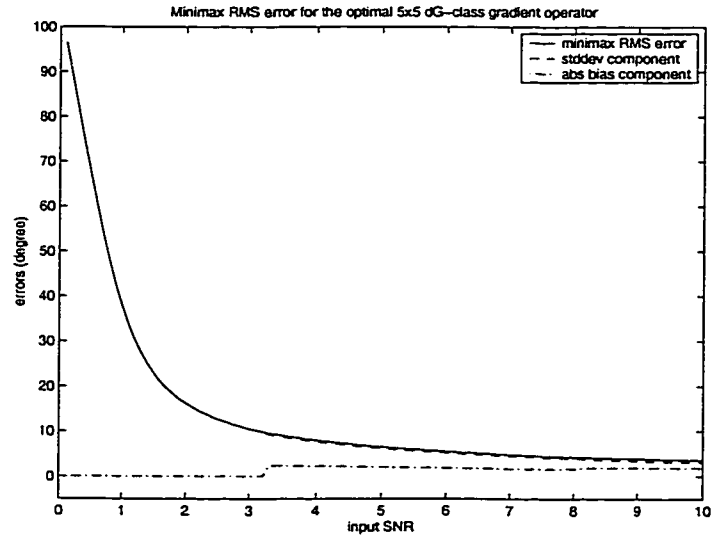
For each given input SNR, the optimal value for s is found which minimizes the worst-case RMS error of Equation (4.24).

For the neighborhood size of 5×5 , the optimal smoothing factor s and the corresponding SNR gain κ_0 are plotted versus the input SNR level in Figure 4.8(a). The achieved minimax RMS error at the examined input SNR levels with its bias and standard deviation components is shown in Figure 4.8(b). The computation of the optimal value for s in neighborhood sizes other than 5×5 can be carried out in exactly the same fashion as is done here.

4.7 Discussions

4.7.1 Comparison of optimally tuned operators

We have applied the theory for deriving the optimal gradient operator on three classes of 5×5 operators, namely Sobel-class, IDDGO-class, and dG-class. The numerical search procedure outlined in Section 4.3.2 for the full 6-parameter 5×5 gradient operator is also carried out. The resulting operator is referred to as the fourth class. It

(a) smoothing factor s for dG and SNR gain κ_0 

(b) achieved minimax RMS error

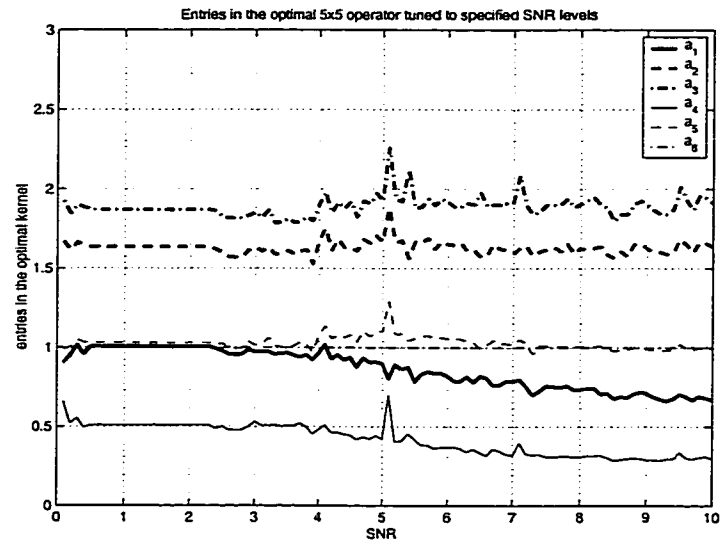
Figure 4.8: Smoothing factor s and corresponding SNR gain κ_0 for the 5×5 dG operator, and the achieved minimax RMS error with its two components.

is supposed to be the optimal gradient operator for the given input SNR. The coefficients in the row-derivative kernel and the resulting gradient orientation estimation performance are shown in Figure 4.9. Now for each input SNR in the range of 0 to 10 with the interval of 0.1, we have four operators, each being the optimal member of its class tuned to the specified SNR.

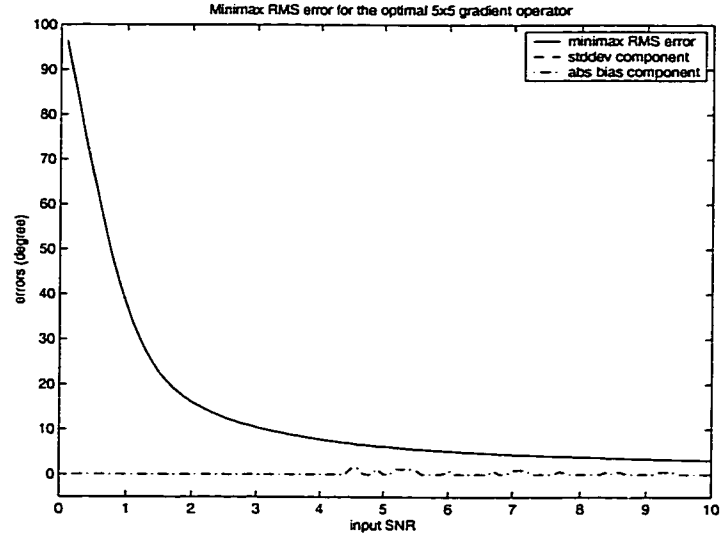
The achieved RMS error plots for these operators are compared in Figure 4.10. For the purpose of comparison, the curves are also shown for two commonly used operators, namely, the dG operator with fixed $s = 1.0$ and the Sobel operator with fixed $K_5 = 2$. It can be seen from Figure 4.10(a) that the optimal operators from the four classes are fairly close in their performance, which is quite noticeably better than the dG with fixed $s = 1.0$ and Sobel with fixed $K_5 = 2$. To allow the study of the detailed differences between the operator classes, two plots of their relative values are shown in Figures 4.10(b) and 4.10(c). In Figure 4.10(b), the achieved minimax RMS error of the fourth class is subtracted from the errors for all classes, and the remaining part of the error is plotted. The unit is the angular degree. In Figure 4.10(c), this remaining part is represented as a percentage of the value that is subtracted, i.e., the achieved minimax RMS error of the fourth class operator.

From these plots, we observe that the fourth class operator is indeed the best performing operator through out the input SNR range of 0 to 10. None of the three other classes can achieve the same low level of error. For these three classes, we observe that, for high input SNR values greater than 4.5, the IDDGO class is the best and its performance is very close to that of the fourth class. For SNR less than 4.5, the dG class is the best among the three. The Sobel class is never the best among the three.

The dG operator with fixed $s = 1$ and Sobel with fixed $K_5 = 2$ both compare very unfavorably with the tuned operators from the four classes for the low SNR range. These tuning parameter values are appropriate only for some very high input SNR levels.



(a) six independent coefficients in the row-derivative kernel of the fourth class operator



(b) achieved minimax RMS error

Figure 4.9: Coefficients in the mask of the optimal fourth class operator, and the achieved minimax RMS error with its two components.

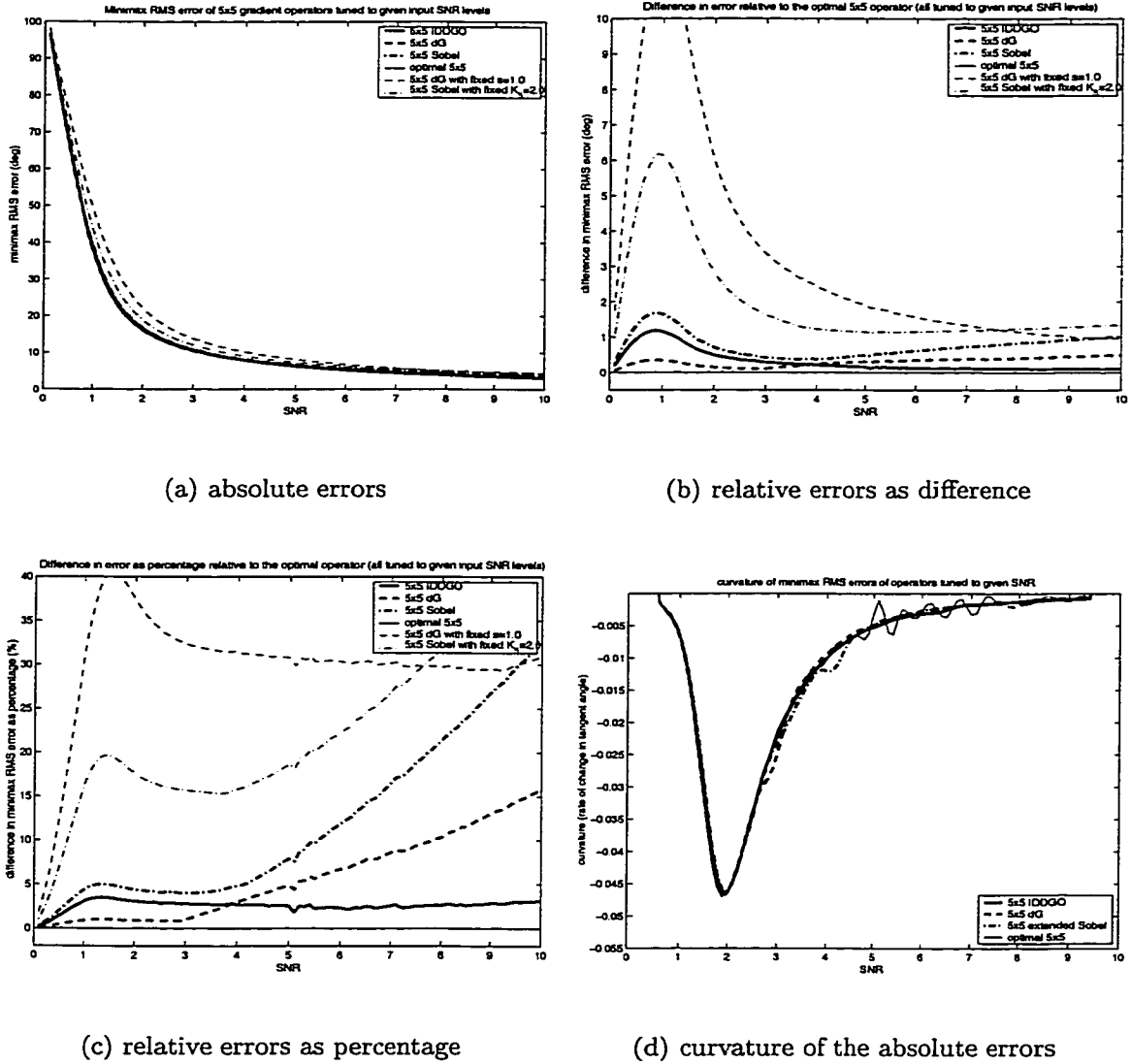


Figure 4.10: Comparison of gradient operator classes optimally tuned to given input SNR levels. (a) achieved minimax RMS errors in degrees; (b) difference in error relative to the that of the optimal 5×5 (fourth class) operator; (c) difference represented as percentage relative to the error of the optimal 5×5 (fourth class) operator; (d) the maximum in the absolute value of the curvature identifies the “knee” of the error curves in (a).

4.7.2 *Comparison of fixed members of operator classes*

The optimal operators we obtained in the experiments just described are tuned to the given input SNR levels. However, in real applications the true SNR is not known. Although it can be estimated, the estimated value is expected to be quite noisy. Iterative procedures can be attempted to improve the estimation quality. One related example is the adaptive bias correction technique for the orientation estimation. This type of technique is quite likely to fail for low SNR situations [96].

For robustness of the operation, ideally we wish to use a single value for the tuning parameter in the gradient operator that would work reasonably well for all SNR levels. Alternatively, we can break the entire range of SNR levels of interest into a very small number of intervals and identify a value for the tuning parameter for each of the intervals. The more intervals we break the entire SNR range into, the better performance we can have for the operator to work in its designated interval. The down side is that it is more likely that the estimated SNR level is out of the interval which contains the true SNR level. This would result in the use of the operator tuned for the wrong SNR and hence the degradation in performance. We need to strike a balance between the operator performance within its designated SNR interval and the probability that the correct SNR interval is identified by the estimated SNR.

We observe that the achieved minimax RMS errors of the gradient operators as functions of the input SNR roughly take the shape of an exponentially decreasing function. The “knee” of the curve is a good point to break the curve into a fast decreasing part and a slowly decreasing part. We identify the knee to be the point with the maximum curvature [60, Section 11.5.7]. The curvature of the smoothed minimax RMS error curves is shown in Figure 4.10(d). We see the knee occurs at around input SNR of 1.9 to 2. Therefore, we roughly break the entire input SNR range of 0 to 10 into two pieces, namely the low SNR range of 0 to 2 and the high SNR range of 2 to 10. We identify two operators from each class for operating on the

$$\begin{array}{cc}
\begin{pmatrix} -0.5 & -0.8 & -1 & -0.8 & -0.5 \\ -0.4 & -1 & -0.654 & -1 & -0.4 \\ 0 & 0 & 0 & 0 & 0 \\ 0.4 & 1 & 0.654 & 1 & 0.4 \\ 0.5 & 0.8 & 1 & 0.8 & 0.5 \end{pmatrix} & \begin{pmatrix} -0.5 & -0.8 & -1 & -0.8 & -0.5 \\ -0.4 & -1 & -1.262 & -1 & -0.4 \\ 0 & 0 & 0 & 0 & 0 \\ 0.4 & 1 & 1.262 & 1 & 0.4 \\ 0.5 & 0.8 & 1 & 0.8 & 0.5 \end{pmatrix} \\
\text{(a) } K_5 = 0.654 \text{ for low SNR} & \text{(b) } K_5 = 1.262 \text{ for high SNR}
\end{array}$$

Figure 4.11: Row derivative kernels for the 5×5 Sobel operator

$$\begin{pmatrix} -0.4562 & -1.5936 & -1.9728 & -1.5936 & -0.4562 \\ -0.2417 & -0.8104 & -1 & -0.8104 & -0.2417 \\ 0 & 0 & 0 & 0 & 0 \\ 0.2417 & 0.8104 & 1 & 0.8104 & 0.2417 \\ 0.4562 & 1.5936 & 1.9728 & 1.5936 & 0.4562 \end{pmatrix}$$

Figure 4.12: Row derivative kernel for the 5×5 IDDGO with $L = 1.841$ for both low and high SNR ranges

two SNR ranges. The operator tuned to SNR=1 is used for the low SNR ranges and the operator tuned to SNR=6 is used for the high SNR range. The row derivative kernels for these operators are shown in Figures 4.11, 4.12, 4.13 and 4.14. Notice that, for the IDDGO, since the single value of $L = 1.841$ is optimal for the entire SNR range of 0 to 10, there is only one kernel.

From the dG and the fourth class kernels in Figures 4.13 and 4.14, we see that for the lower SNR range, the mass of the operators concentrates less to the center column. This increases the averaging effect of the kernel and results in better noise suppression. For the higher SNR range, less noise suppression capability is needed,

$$\begin{pmatrix} -0.9112 & -1.2762 & -1.4278 & -1.2762 & -0.9112 \\ -0.6381 & -0.8937 & -1 & -0.8937 & -0.6381 \\ 0 & 0 & 0 & 0 & 0 \\ 0.6381 & 0.8937 & 1 & 0.8937 & 0.6381 \\ 0.9112 & 1.2762 & 1.4278 & 1.2762 & 0.9112 \end{pmatrix}$$

(a) $s = 2.11$ for low SNR

$$\begin{pmatrix} -0.3785 & -0.7725 & -0.9798 & -0.7725 & -0.3785 \\ -0.3862 & -0.7883 & -1.0000 & -0.7883 & -0.3862 \\ 0 & 0 & 0 & 0 & 0 \\ 0.3862 & 0.7883 & 1.0000 & 0.7883 & 0.3862 \\ 0.3785 & 0.7725 & 0.9798 & 0.7725 & 0.3785 \end{pmatrix}$$

(b) $s = 1.45$ for high SNRFigure 4.13: Row derivative kernels for the 5×5 dG operator

$$\begin{pmatrix} -1.0061 & -1.6349 & -1.8691 & -1.6349 & -1.0061 \\ -0.5090 & -1.0290 & -1 & -1.0290 & -0.5090 \\ 0 & 0 & 0 & 0 & 0 \\ 0.5090 & 1.0290 & 1 & 1.0290 & 0.5090 \\ 1.0061 & 1.6349 & 1.8691 & 1.6349 & 1.0061 \end{pmatrix}$$

(a) for low SNR

$$\begin{pmatrix} -0.8182 & -1.6497 & -1.9174 & -1.6497 & -0.8182 \\ -0.3687 & -1.0534 & -1 & -1.0534 & -0.3687 \\ 0 & 0 & 0 & 0 & 0 \\ 0.3687 & 1.0534 & 1 & 1.0534 & 0.3687 \\ 0.8182 & 1.6497 & 1.9174 & 1.6497 & 0.8182 \end{pmatrix}$$

(b) for high SNR

Figure 4.14: Row derivative kernels for the optimal 5×5 operator obtained by numerical search (the fourth class operator).

and the mass concentrates more to the center column.

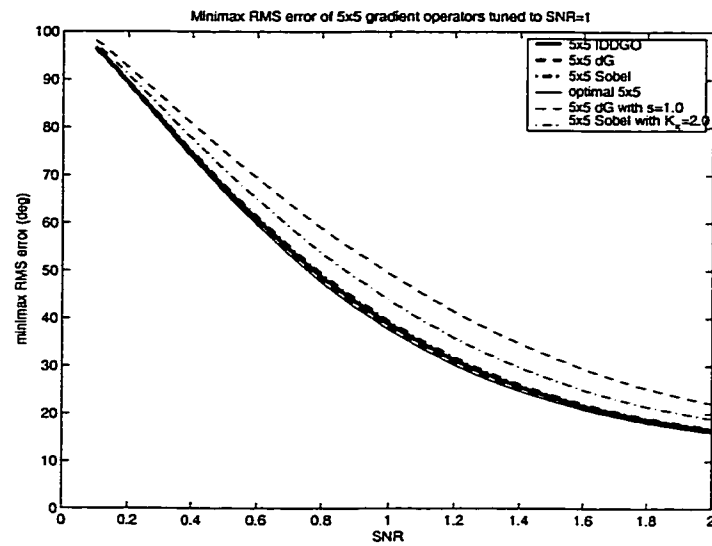
The performance of these operators is shown in Figures 4.15 and 4.15 for the low and high SNR range, respectively. The performance of these operators are again very close to each other, with generally the same pattern as observed in Figure 4.10. One particular observation is that, when the input SNR is greater than 8.5, the fixed IDDGO actually out-performs the optimal 5×5 operator obtained by numerical search. This is simply because that operator is only tuned to the input SNR of 5. The generalization capability of the optimal operator to cover other input SNR values is not as good as that of the IDDGO. If the input SNR is expected to be at this high value range, the optimal operator tuned for, say, SNR=10 can be used to achieve better performance. The row derivative kernel for this operator is

$$\begin{pmatrix} -0.6595 & -1.5865 & -1.8473 & -1.5865 & -0.6595 \\ -0.3736 & -0.8892 & -1 & -0.8892 & -0.3736 \\ 0 & 0 & 0 & 0 & 0 \\ 0.3736 & 0.8892 & 1 & 0.8892 & 0.3736 \\ 0.6595 & 1.5865 & 1.8473 & 1.5865 & 0.6595 \end{pmatrix}$$

The performance of all operators studied here depends heavily on the value of the tuning parameters. We make a special note on the dG operator which is used in the popular Canny's edge detector. In most common implementations of the operator, the value 0.6 to 1 is use for the smoothing factor s when the neighborhood size of 5×5 is used. As is observed in the plots, its performance compares very unfavorably with the values chosen here, i.e., $s = 1.85$ for the low SNR range and $s = 1.41$ for the high SNR range.

4.8 Validation experiment for the MSE computation

In the previous experiments that search for optimal tuning parameters of gradient operators, the mean-squared error (MSE) in the criterion function (Equation (4.24))



(a) absolute errors

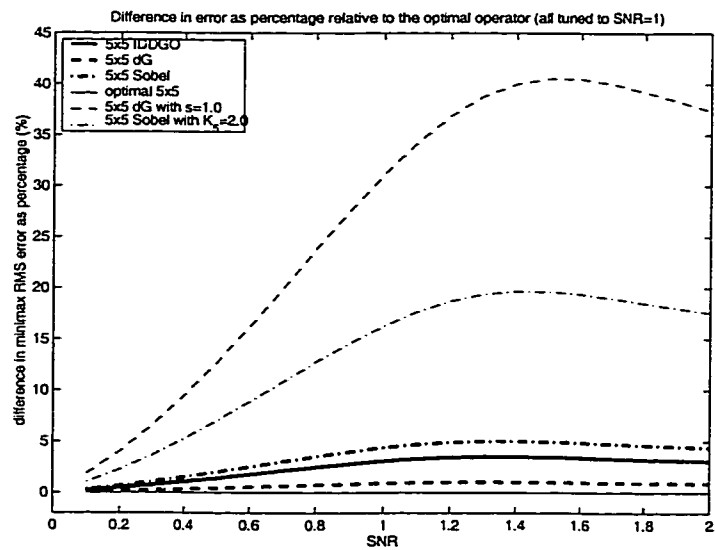
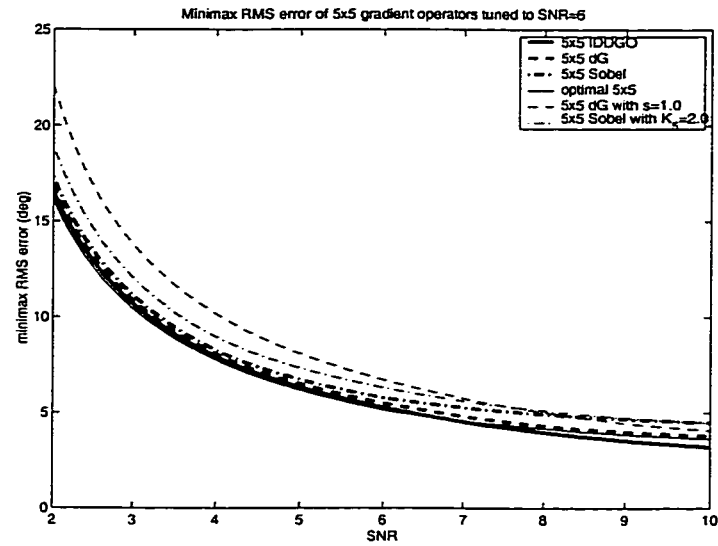
(b) difference in error relative to the that of the optimal 5×5 operator, represented as percentage

Figure 4.15: Comparison of gradient operators for the low SNR range.



(a) absolute errors

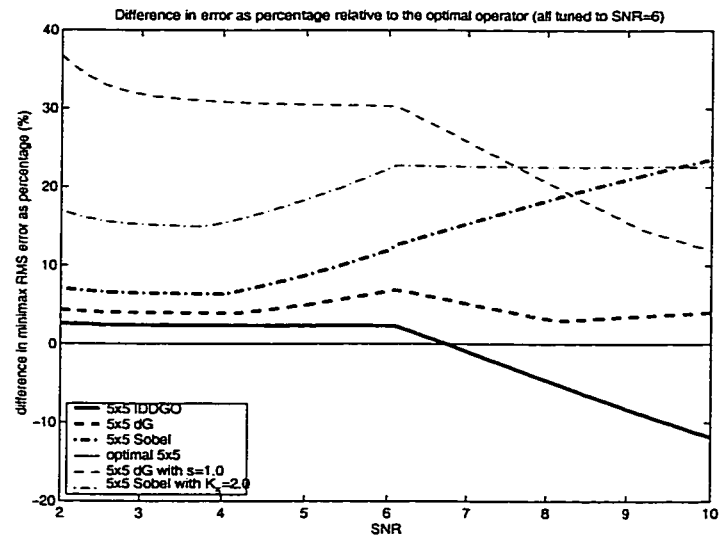
(b) difference in error relative to the that of the optimal
5 × 5 operator, represented as percentage

Figure 4.16: Comparison of gradient operators for the high SNR range.

is computed through numerical evaluation of the analytical expression, which is based on the distribution model Equation (4.14) for the orientation estimate. Although the derivations that lead to Equation (4.14) are rigorous, we still wish to experimentally validate the results from these computations by comparing them with those obtained from Monte Carlo simulations. This is conducted by repeating the gradient operator optimization experiment with a new module for computing the MSE. In this new module, noisy observations of image neighborhood intensity values are simulated by adding white Gaussian noise to the noise-free edge patterns. A large number of such noisy observations are generated, on which the gradient operator is applied. The gradient orientation estimates made by the operator are used to compute the MSE, which is just the average of the squared difference between the estimate and the true orientation.

In this Monte Carlo simulation, variation in the observed values due to limited sample size is inevitable. A balance point needs to be determined between the sampling variation and the amount of computation involved. For a chosen sample size, the expected sampling variation can be estimated. The values observed in the previous experiments with numerical evaluation of the distribution model should fall within the range of the expected sampling variation.

In our experiment, we decided on a sample size of 10000. This means that, for each combination of the value of the gradient operator tuning parameter, the sampled value of the true gradient orientation, and the sampled value of the true edge translation, 10000 noisy observations are simulated and used to compute the MSE. This requires a large amount of computation. For example, with 46 steps to cover the $[0^\circ, 45^\circ]$ orientation range and 21 steps to cover the entire range of edgel translation, the search for the optimal 5×5 Sobel operator takes more than four hours to run on a 500MHz Pentium III CPU, whereas it takes less than a second for the numerical evaluation method without the Monte Carlo simulations.

Due to the large amount of computation involved, the validation experiment is

only carried out for the 5×5 Sobel operator at ten sparsely sampled SNR values: $1, 2, \dots, 10$.

Figure 4.17(a) compares the optimal values for the tuning parameter K_5 in the 5×5 Sobel operator obtained in the simulation based and in the numerical evaluation based optimization experiments. Figure 4.17(b) compares the achieved minimax RMS error. Figure 4.17(c) is the bar-plot for studying the significance of the difference in the achieved MSE between the two methods. At each input SNR value, the center of the vertical bar is located at the MSE from the Monte Carlo simulations. The full length of the bar is six times the estimated standard deviation of the MSE due to sampling variation. As can be seen from the bar-plot, the difference between the Monte Carlo simulations and numerical evaluation is within the sample variation.

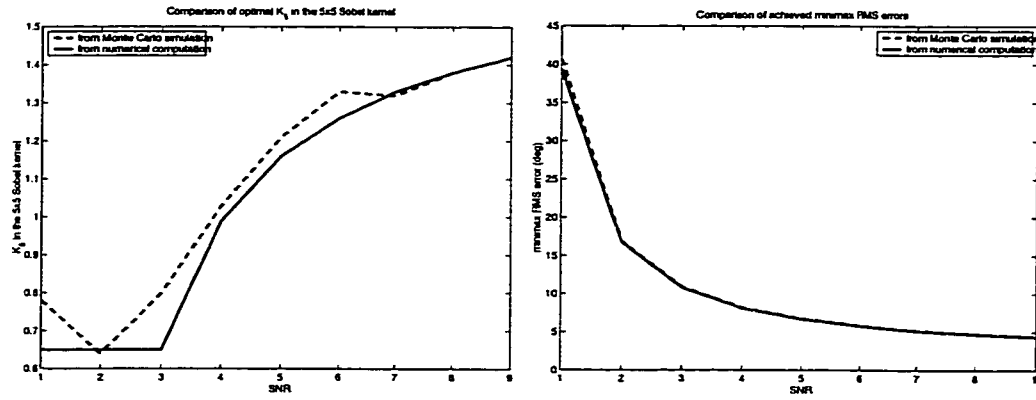
4.9 Experiment on the smoothed edge

Although the formulation developed in this chapter is general and applicable to any image intensity function of interest, the previous discussions mainly concern the sharp step edge in the examples and experiments. The reason for doing that is to use the simplest example to illustrate the main idea of the formulation. Now that we have completed the study of the performance of the various gradient operators on the simplest sharp step edge, we are ready to study the more realistic case of the smoothed edge.

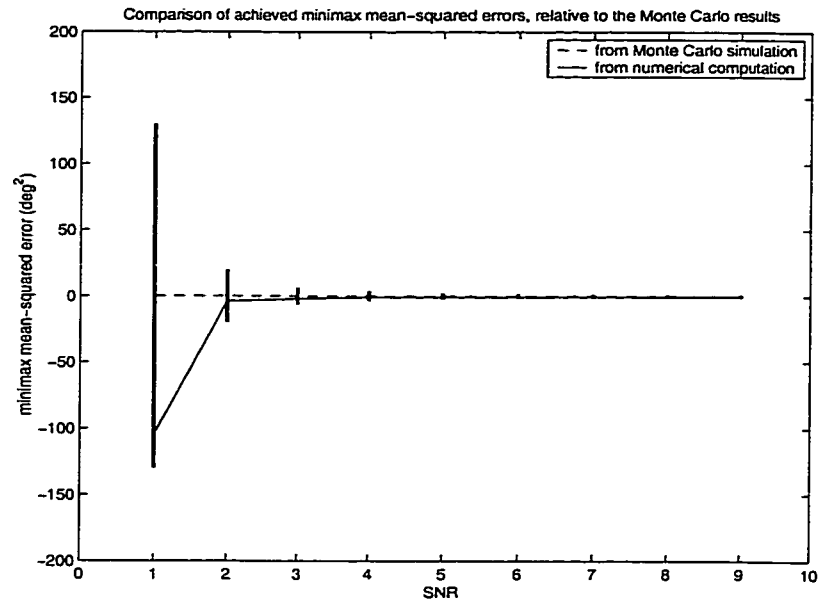
Due to the effects of imaging sensor optics, shading and atmospheric scattering, sharp intensity changes often appear as smoothed or blurred intensity transition in the image. This blurred intensity transition can be modeled by a sigmoid function [38]

$$I(x) = \begin{cases} -\frac{1}{2} & x < -r \\ \frac{1}{2} - \frac{1}{\pi}(\arccos \frac{x}{r} - \frac{x}{r}\sqrt{1 - \frac{x^2}{r^2}}) & -r \leq x \leq r \\ \frac{1}{2} & x > r \end{cases} \quad (4.43)$$

where the blur parameter r controls the spatial extent $([-r, r])$ over which the intensity

(a) optimal K_5

(b) achieved minimax RMS error



(c) bar-plot for studying difference

Figure 4.17: Comparison of gradient operator optimization results by Monte Carlo simulation and by numerical evaluation of analytical expressions.

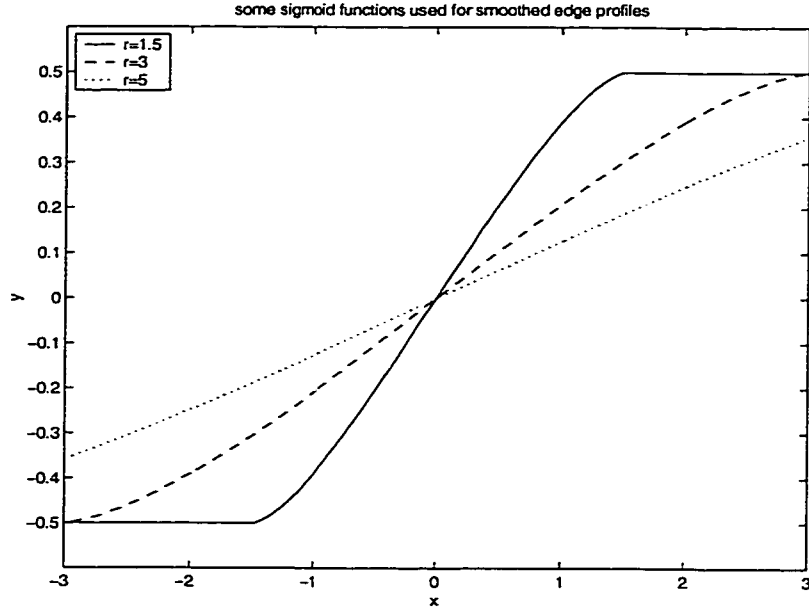


Figure 4.18: Plots of sigmoid function with different blur parameters.

changes smoothly from the saturated minimum value to the saturated maximum value. Some examples of this sigmoid function with particular values for r are shown in Figure 4.18.

To understand the impact of the blur on the gradient estimation performance, we conduct a set of experiment similar to that conducted in Section 4.7.1. The difference here is that the SNR level is held at constant and the blur parameter is varied. For each value of the blur parameter, the noise-free intensity values of the blurred transition are first determined, to which independent noise values are then added. There is no further smoothing after adding the noise. The four classes of gradient operators, i.e., the Sobel-class, the IDDGO-class, the dG-class, and the fourth class (see Section 4.3.2), are individually tuned to their optimal performance (smallest worst-case RMS orientation estimation error) on the blurred image.

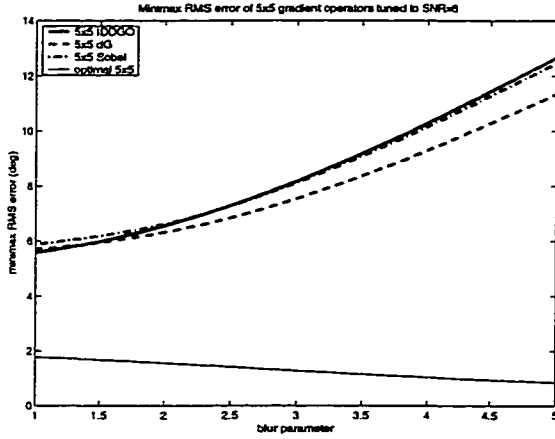
The experiment is conducted at the SNR level of 5. The blur parameter r is varied

over the range $[1, 5]$. The achieved minimax RMS errors in the gradient orientation estimate are used as the measures of operator performance. Also, the values for the optimal tuning parameters of the operators are recorded to show the sensitivity of the operators to the change in the amount of blur in the image. These values are plotted against the blur parameter r in Figure 4.19.

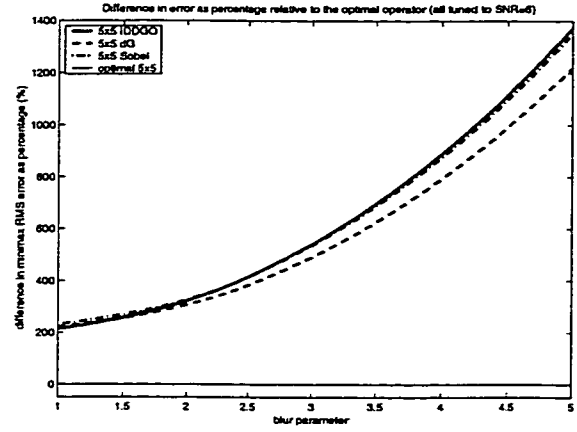
From the plots, we observe that the optimal 5×5 operator from the numerical search procedure can again be made to adapt well to the image blur and yield the best performance. The next best performance comes from the dG operator, which performs noticeably better than the Sobel and IDDGO operators. The IDDGO operator performs better than the Sobel operator for up to medium (r around 2) amount of blur, and performs not as good for larger amount of blur. This experiment result suggests that the IDDGO operator does not adjust very well to significant amount of image blur of this type.

Similar to the situation in the experiment with varying SNR, the IDDGO operator is again most insensitive to the amount of change in the image, in this case the amount of blur. This is a preferable property since in real applications the amount of blur is often unknown and difficult to estimate. The insensitivity of the IDDGO operator to the amount of blur renders the robustness to unexpected situations.

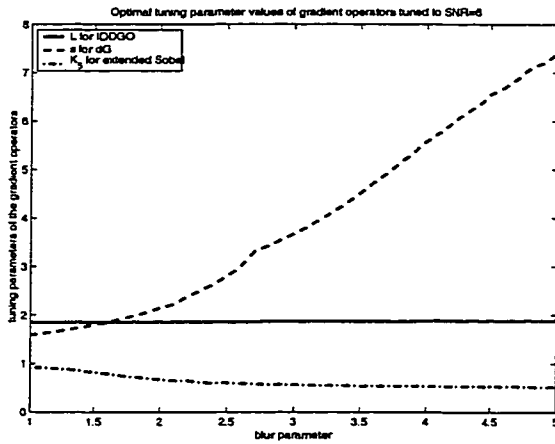
When there is very significant amount of blur ($r = 5$), the noise-free image intensity profile becomes very flat, as shown in Figure 4.20(a). The optimal 5×5 operator converges to the Prewitt operator [163, 60], whose row derivative mask is shown in Figure 4.20(b). The explanation is that, as the amount of blur increases, the effective SNR in the observed local image neighborhood decreases. To compensate for the decreased SNR, the optimal operator changes towards the matched filter, which maximizes the output SNR.



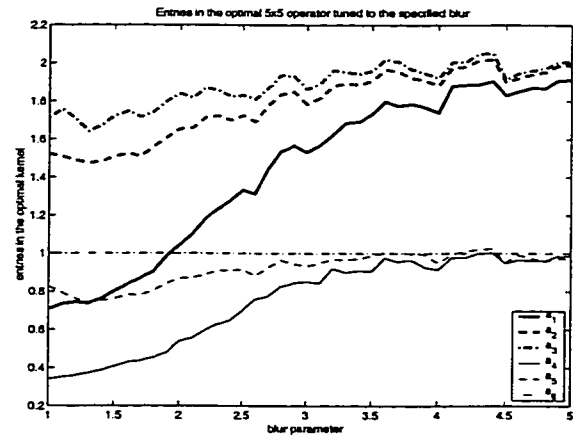
(a) absolute errors



(b) relative errors as percentage



(c) tuning parameters of the Sobel, ID-DGO, dG class operators



(d) tuning parameters of the fourth class operator

Figure 4.19: Comparison of gradient operator classes optimally tuned to given blur parameters. (a) achieved minimax RMS errors in degrees; (b) difference represented as percentage relative to the error of the optimal 5×5 operator; (c) the optimal tuning parameters for the Sobel, IDDGO, and dG class operators; (d) the optimal tuning parameters for the optimal 5×5 operator.

-0.2477	-0.2477	-0.2477	-0.2477	-0.2477	-2	-2	-2	-2	-2
-0.1265	-0.1265	-0.1265	-0.1265	-0.1265	-1	-1	-1	-1	-1
0	0	0	0	0	0	0	0	0	0
0.1265	0.1265	0.1265	0.1265	0.1265	1	1	1	1	1
0.2477	0.2477	0.2477	0.2477	0.2477	2	2	2	2	2
(a)					(b)				

Figure 4.20: Intensity values of very significantly blurred edge and the Prewitt gradient operator mask. (a) Intensity values of a blurred edge with the blur parameter $r = 5$ and true gradient orientation of 0° ; (b) The column mask of the Prewitt operator.

4.10 Conclusion

In this chapter, we address the optimal gradient operator design problem. The discussion is based on a general framework for characterizing the performance of gradient direction estimation. The quality of gradient magnitude estimation is discovered to be linked directly to the precision of the direction estimate. This framework can be used for a wide class of linear, shift-invariant gradient operators to study their performance. Analytical expressions for the bias, standard deviation and mean-squared error for the gradient direction estimate are obtained. The operator characterization results using these expressions agree very closely with the empirical simulation based studies reported in the literature. Full characterization of any gradient operator under any conditions of interest can now be carried out in this framework without the need to conduct empirical simulations, which intrinsically involves uncertainty from the use of random samples.

We formulate a minimax optimization problem for the design of gradient operators. The worst-case mean-squared error in the gradient direction estimate is minimized in the search for the optimal operator. The problem is solved numerically for neighborhood sizes 3×3 and 5×5 . Optimal operators for larger neighborhood sizes, for

any SNR level of interest and for any other edge types can be obtained by following the same procedure. As an example, the smoothed edge image modeled by a sigmoid function is also used in the experiment. For the 5×5 neighborhood, three classes of popular gradient operators are optimized. As expected, we observe that image blur causes gradient estimation performance to degrade significantly for all operator classes.

Chapter 5

EDGEL LOCATION UNCERTAINTY CHARACTERIZATION

5.1 *Introduction and background*

In our FLIR ATR application, boundary contrast is the primary visual cue for target detection and recognition. In Chapter 3 we describe how a weighted average type of scheme can be used for accumulation of the contrast evidence. This scheme makes use of the fact that high contrast boundary points can also be well localized spatially. In this chapter, we quantitatively study the relationship between boundary point contrast and location uncertainty. The study is carried out in the edge detection framework. The result of the derivation is used in the numerical evaluation of the boundary point location uncertainty as a function of the contrast.

5.1.1 *Location estimation*

Local neighborhood gradient based edge operators, e.g., [51, 19], search the maximum in the derivative of the image intensity surface along the direction of the gradient, or, equivalently, the zero-crossing in the second directional derivative in that direction. Although it has been argued that the gradient orientation estimate is usually quite noisy [96, 163], the analysis of the localization error of edge detectors has yet remained a one-dimensional formulation. The image intensity profile along the cutting plane in the direction of the gradient estimate has been studied and the radial localization error along that direction has been examined [19, 149, 17, 150]. The error in the orientation of the cutting plane has not been incorporated in the error analysis of edge localization

until recently. Marimont and Rubner [98] were the first to combine orientation and radial localization distributions to compute the edge location probability for edge pixel detection.

To distinguish single edge pixels from edge features which usually consist of multiple edge pixels, we use the term *edgel* to refer to edge elements which are just single edge pixels. The orientation of the edgel is the gradient orientation, and the location of the edgel is the zero-crossing in the second directional derivative.

In this chapter, we incorporate the gradient orientation estimation error into the study of the edgel location estimation. The edgel localization error is then a two-dimensional vector as opposed to the scalar used in previous analyses [19, 149, 17, 150]. This results in a more accurate characterization of the relationship between contrast and location uncertainty.

5.1.2 Background

We now restate some technical details concerning edgel measurements in gradient based methods. The underlying assumption is that the noise can be modeled as zero-mean white additive Gaussian.

- By the discussions in Section 4.2, the distribution of the orientation estimate $\hat{\theta}$ has the pdf [96]

$$f_{\Theta}(\theta|\theta_0, \kappa) = \exp\left(-\frac{\kappa^2}{2}\right) \left\{ \frac{1}{2\pi} + \frac{\kappa \cos(\theta - \theta_0)}{\sqrt{2\pi}} \exp\left(\frac{\kappa^2 \cos^2(\theta - \theta_0)}{2}\right) \cdot \left[\frac{1}{2} + \frac{1}{2} \operatorname{erf}\left(\frac{\kappa \cos(\theta - \theta_0)}{\sqrt{2}}\right) \right] \right\} \quad (5.1)$$

where θ_0 is the true underlying gradient orientation and κ is the gradient operator output SNR. This distribution is symmetric about θ_0 . The output SNR κ is also the concentration parameter, which controls how much the distribution is concentrated around θ_0 . The bigger κ is, the more concentrated the distribution is.

For any given gradient operator, κ is given by

$$\kappa = \kappa_0 \Omega \quad (5.2)$$

where Ω is the input SNR defined as ratio of the true step edge contrast over the standard deviation of the noise. In practice, Ω is estimated by Equation (4.12).

κ_0 is a constant determined by the gradient operator itself

$$\kappa_0 = \sqrt{\frac{\mu_{1,0}^2 + \mu_{2,0}^2}{\lambda^2}} \quad (5.3)$$

where λ^2 is the sum of squared elements of the gradient operator kernel; $\mu_{1,0}$ and $\mu_{2,0}$ are the two components of the gradient computed by the operator on the noise-free unit-contrast step edge. See Section 4.2 for details on κ_0 and λ^2 . In the 5×5 neighborhood with a true gradient orientation of 0° , κ_0 is 1.28 for the standard cubic facet model estimate and 1.95 for the integrated directional derivative gradient operator (IDDGO) [163, 60, 96].

- The estimated radial edgel location \hat{v} , taken along a line in the gradient direction, is determined in each neighborhood for the center pixel in that neighborhood. It is determined to be the location of the zero-crossing of the second directional derivative in the direction of the gradient. The derivatives are computed by the IDDGO which is based on the cubic facet model [51, 60].

In the edge detection application, only the negatively sloped zero-crossing is of interest. Positively sloped ones do not correspond to authentic edges and are called “phantom” edges [25] which most often result from stair-case like image intensity patterns. The impact of the sign of the slope is discussed in Section 5.5.

When the image is noisy, even along the true gradient orientation, the estimated radial edgel location \hat{v} is different from the true edgel location. The localization error from reasonable detectors has a distribution that is symmetric about 0.

- Using the cubic facet model, \hat{v} is determined to be the inflection point of the cubic polynomial along the direction of the gradient orientation.

$$\hat{v} = -\frac{\hat{c}_2}{3\hat{c}_3} \quad (5.4)$$

where \hat{c}_2 and \hat{c}_3 are the estimated coefficients for the second and third order term of the cubic polynomial. Due to the observation noise,

$$\hat{c}_2 \sim n(c_2^* \Omega \sigma, \kappa_2^2 \sigma^2) \quad \hat{c}_3 \sim n(c_3^* \Omega \sigma, \kappa_3^2 \sigma^2)$$

where c_2^* and c_3^* are the true value of the cubic polynomial coefficients; κ_2 and κ_3 are constants from the cubic facet model; and σ^2 is the variance of the observation noise.

Using the cubic facet model, σ^2 is estimated by $\epsilon^2/(N - 10)$ where ϵ^2 is the sum of the squared facet fitting error, and N is the number of pixels used in the fit. (In the 5×5 neighborhood, $N = 25$.) Ω is estimated by Equation (4.12). Other constants involved in the distribution of \hat{c}_2 and \hat{c}_3 for the unit-contrast step edge in the 5×5 neighborhood with the true gradient orientation of 0° are

$$c_2^* = 0, c_3^* = -\frac{1}{12}, \kappa_2 = 0.1196, \kappa_3 = 0.1179 \quad (5.5)$$

Under the assumed additive white Gaussian noise model and using the discrete orthogonal polynomial (DOP) [60] for estimating the facet parameters, \hat{c}_2 and \hat{c}_3 are independent of each other.

It is easily seen that

$$\hat{v} = -\frac{\kappa_2}{3\kappa_3} Z \quad (5.6)$$

where Z is distributed as

$$\frac{n(\frac{c_2^*}{\kappa_2} \Omega, 1)}{n(\frac{c_3^*}{\kappa_3} \Omega, 1)} \quad (5.7)$$

The term *normal quotient* is used to refer to the family of distributions for the ratio of two independent normal random variables with unit variance. The probability density function (pdf) for this family has the form [113]

$$f_Z(z \mid \mu_1, \mu_2) = \frac{1}{\pi} \frac{1}{z^2 + 1} \exp \left(-\frac{1}{2} \cdot \frac{(\mu_1 - z\mu_2)^2}{z^2 + 1} \right) \cdot \left\{ \exp \left(-\frac{(z\mu_1 + \mu_2)^2}{2(z^2 + 1)} \right) + \sqrt{\pi} \frac{z\mu_1 + \mu_2}{\sqrt{2(z^2 + 1)}} \operatorname{erf} \left(\frac{z\mu_1 + \mu_2}{\sqrt{2(z^2 + 1)}} \right) \right\}$$

For the sharp step edge, $\mu_1 = 0$ and $\mu_2 = \frac{c_3^*}{\kappa_3} \Omega$.

$$f_Z(z \mid 0, \mu_2) = \frac{1}{\pi(z^2 + 1)} e^{-\frac{\mu_2^2}{2}} \cdot \left\{ 1 + e^{\frac{\mu_2^2}{2(z^2 + 1)}} \frac{\sqrt{\pi} \mu_2}{\sqrt{2(z^2 + 1)}} \operatorname{erf} \left(\frac{\mu_2}{\sqrt{2(z^2 + 1)}} \right) \right\} \quad (5.8)$$

It is clear that in this situation the distribution is symmetric about 0.

For gradient estimation, we were tempted to use the fourth class gradient operator described in Chapter 4, i.e., the operator obtained from numerically optimizing the worst-case error criterion function over the full parameter space. However, this operator as developed there is not appropriate for the situation here. The reasons are as follows. The entire formulation of Chapter 4 is based on the consideration of developing the best operator for a specific and known SNR level. In the ATR application, the true SNR is not known. A good method for SNR estimation is required before the operator can be effectively applied. This method has not yet been developed, and is a topic of future research. On the other hand, the IDDGO has a natural method for noise variance estimation from the local neighborhood, which is necessary in estimating the SNR. Also, the optimal value for the tuning parameter of the IDDGO remains constant over a very wide range of SNR levels, which makes it very robust to changing SNR levels. In addition, there are a number of studies in the literature [163, 96, 123] that report the superior performance of the IDDGO. Therefore, the IDDGO is used for gradient estimation in this dissertation unless otherwise stated.

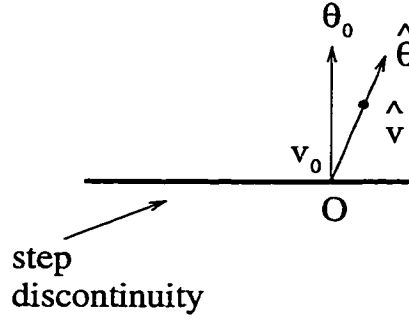


Figure 5.1: Step edge detection scenario.

5.2 Edgel location estimation

The edgel location estimation scenario is illustrated in Figure 5.1. The polar coordinate system is used. The true and unobserved underlying gradient orientation is θ_0 . The step edge discontinuity occurs at a radial distance of v_0 from the origin along the direction θ_0 . The true input SNR is Ω_0 . The chosen gradient operator gives the estimate for these as $\hat{\theta}$, \hat{v} and $\hat{\Omega}$, respectively. The estimate of the two-dimensional edgel location is $(\hat{v} \sin \hat{\theta}, \hat{v} \cos \hat{\theta})$ in the row-column coordinate system. These estimates are random variables. Their statistical properties characterize the relationship between the edgel location uncertainty and the step edge SNR. This relationship is to be used along with the polygon centroid uncertainty model to combine target boundary contrast into a single target saliency measure.

5.2.1 Distribution of 2-D location estimate

Using the definition of conditional probability density, we have

$$f(\hat{\theta}, \hat{v} \mid \theta_0, v_0) = f(\hat{v} \mid \theta_0, v_0, \hat{\theta}) f(\hat{\theta} \mid \theta_0, v_0) \quad (5.9)$$

Using the IDDGO for gradient estimation and the zero-crossing of the second directional derivative along the direction of the gradient for edgel location, the two terms

on the right-hand side of the equation can be approximated by the pdf's $f_Z(\cdot)$ and $f_\Theta(\cdot)$, respectively. Specifically,

$$\begin{aligned} f(\hat{\theta} \mid \theta_0, v_0) &= e^{-\frac{\kappa^2}{2}} \left\{ \frac{1}{2\pi} + \frac{\kappa \cos(\hat{\theta} - \theta_0)}{\sqrt{2\pi}} e^{\frac{\kappa^2 \cos^2(\hat{\theta} - \theta_0)}{2}} \left[\frac{1}{2} + \frac{1}{2} \operatorname{erf} \left(\frac{\kappa \cos(\hat{\theta} - \theta_0)}{\sqrt{2}} \right) \right] \right\}, \\ f(\hat{v} \mid \theta_0, v_0, \hat{\theta}) &\approx f(\hat{v} \mid \theta_0, v_0) \\ &= f_Z\left(-\frac{3\kappa_3}{\kappa_2}\hat{v} \mid 0, \mu_2\right) \cdot \frac{3\kappa_3}{\kappa_2} \\ &= \frac{1}{\pi(z^2 + 1)} e^{-\frac{\mu_2^2}{2}} \left\{ 1 + e^{\frac{\mu_2^2}{2(z^2 + 1)}} \frac{\sqrt{\pi}\mu_2}{\sqrt{2(z^2 + 1)}} \operatorname{erf} \left(\frac{\mu_2}{\sqrt{2(z^2 + 1)}} \right) \right\} \cdot \frac{3\kappa_3}{\kappa_2}, \end{aligned}$$

where

$$z = -\frac{3\kappa_3}{\kappa_2}\hat{v}, \quad \kappa = \kappa_0\Omega, \quad \mu_2 = \frac{c_3^*}{\kappa_3}\Omega.$$

Ω is estimated by $\hat{\Omega}$ of Equation (4.12). κ_0 , c_3^* , κ_2 and κ_3 are constants from the IDDGO. For the 5×5 neighborhood, $\kappa_0 = 1.9460$, $c_3^* = -\frac{1}{12}$, $\kappa_2 = 0.1196$, and $\kappa_3 = 0.1179$.

The probability density function of the estimate for the edgel position in the 2-D Cartesian space can be obtained from the above joint density function via standard random variable transformation [21]. Notice that the Jacobian for this transformation is $1/\sqrt{x^2 + y^2}$ and causes a singularity at the origin.

5.2.2 Study of approximation error

Since \hat{v} and $\hat{\theta}$ are both computed from the cubic facet parameters estimated from the noisy image intensity values, \hat{v} is not independent of $\hat{\theta}$. Specifically, not only does the random variable \hat{v} depend on $\hat{\theta}$, the parameters μ_2 , κ_2 and κ_3 its pdf $f(\hat{v} \mid \theta_0, v_0)$ are also functions of $\hat{\theta}$, instead of the constants given earlier. Therefore, the approximation of

$$f(\hat{v} \mid \theta_0, v_0, \hat{\theta}) \approx f(\hat{v} \mid \theta_0, v_0)$$

that we make in computing $f(\hat{\theta}, \hat{v} \mid \theta_0, v_0)$ is likely to cause error. The error is expected to be most observable for low SNR levels, when $\hat{\theta}$ becomes very noisy and has large deviation from θ_0 .

The approximation error in this situation is studied in a simple Monte Carlo experiment. \hat{v} and $\hat{\theta}$ are computed from simulated noisy intensity values with a low SNR of 0.5. The distribution of the observed orientation estimate $\hat{\theta}$ is shown in Figure 5.2(a). Due to the low SNR, $\hat{\theta}$ is very noisy and the distribution is spread all over the full angular range.

We examine the distribution of the 1-D radial location estimate \hat{v} conditioned on the orientation estimate $\hat{\theta}$ being in four angular intervals: $[-\pi/4, \pi/4)$, $[\pi/4, 3\pi/4)$, $[3\pi/4, 5\pi/4)$, $[-3\pi/4, -\pi/4)$. These distributions are shown in Figure 5.2(b). As can be seen clearly, the conditional distribution of \hat{v} for $\hat{\theta}$ being in different ranges are very similar to each other, albeit noticeable difference exist. Specifically, the spread of \hat{v} is larger, and the mode is lower, when $\hat{\theta}$ is away from the true orientation $\theta_0 = 0$. The mode for $\hat{\theta}$ being in the first interval is about 15% taller than that for the third interval. The sample standard deviation of \hat{v} conditioned on $\hat{\theta}$ being in the intervals are 0.7904, 0.8496, 0.8709, and 0.8457, respectively. Comparing these with the value of 0.7162 predicted by our formula for the case $\hat{\theta} = \theta_0 = 0$, we expect that at this SNR level the real location distribution will be more spread out than dictated by our formula with approximations.

Although there are noticeable differences, the pdf curves are still very close to each other for many practical purposes. At high SNR levels, the orientation estimate $\hat{\theta}$ is distributed in a very small range around θ_0 . The error from the approximation becomes hardly noticeable. From these observations, we conclude that, with the approximation error being well tolerable, this approximation gives a much simpler theoretical characterization of the 2-D edgel location distribution. Therefore, it is adopted for this dissertation study.

5.2.3 Preliminary characterization of 2-D location distribution

The probability density function for the 2-D Cartesian location estimate from the IDDGO based estimator is shown in Figure 5.3. (The pdf approaches infinity at the

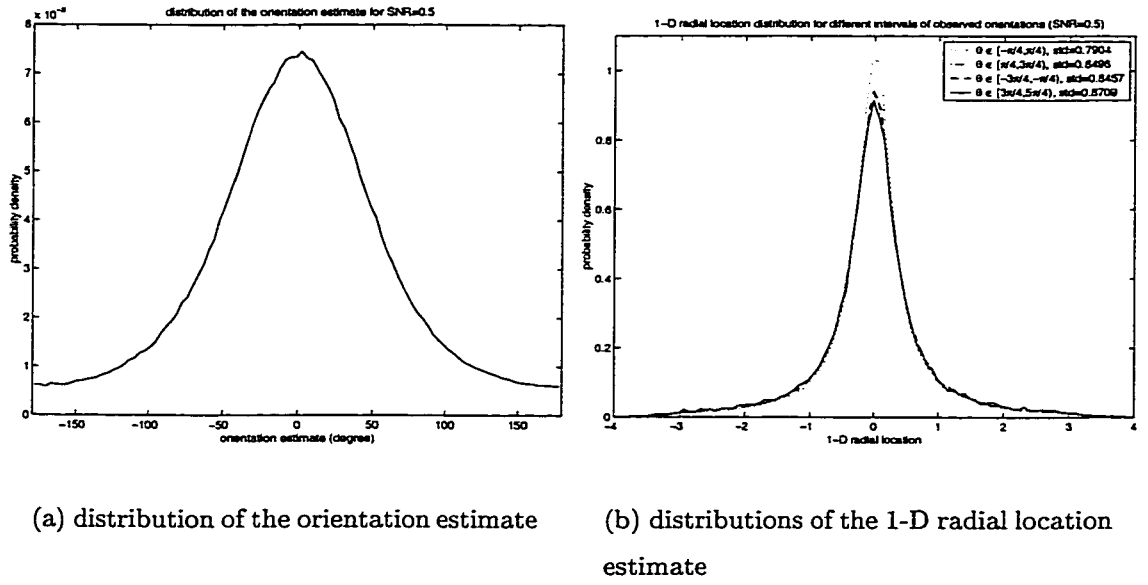


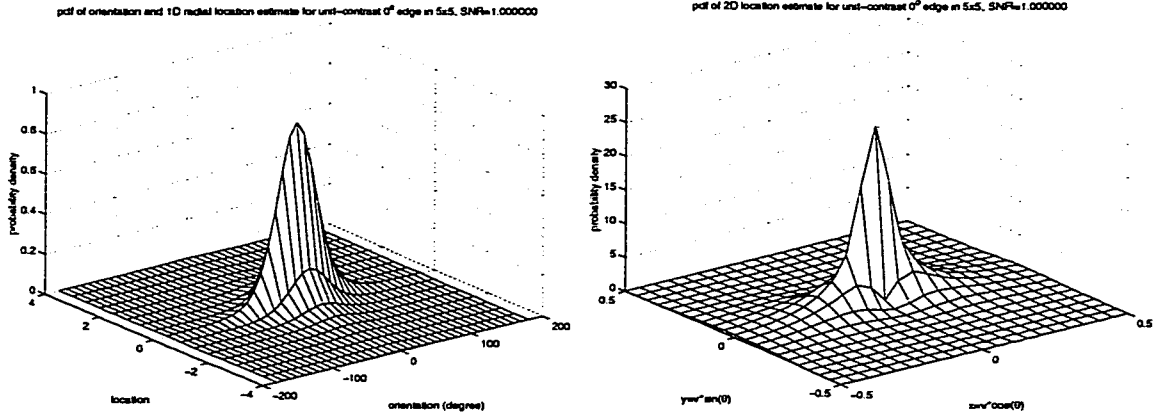
Figure 5.2: Illustration of the approximation error in assuming the independence of \hat{v} from $\hat{\theta}$. Data collected from Monte Carlo experiments at SNR=0.5. (a) wide spread of the orientation estimate $\hat{\theta}$ over the entire angular range; (b) distribution of the 1-D radial location estimate \hat{v} is different for different ranges of $\hat{\theta}$.

origin due to the singularity caused by the Jacobian in the transformation from the polar coordinate system to the Cartesian coordinate system. The plot at the origin shows the ratio of the probability divided by the area of a small circular neighborhood around the origin.) The underlying image data is a straight unit-contrast step edge in the 5×5 neighborhood, with $\theta_0 = 0^\circ$, $v_0 = 0$, and with an input SNR of 1.

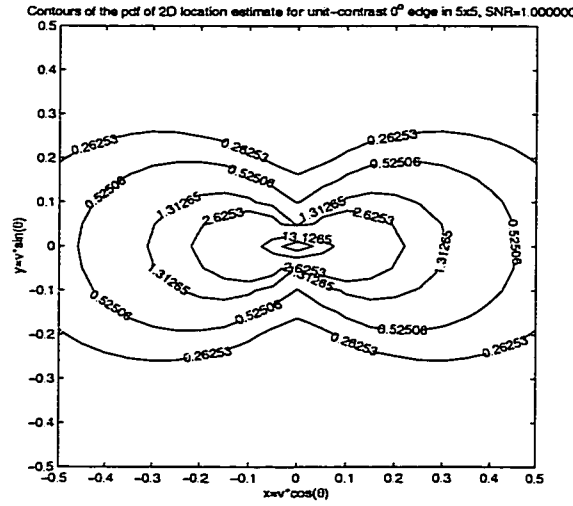
It can be seen from Figure 5.3 that the distribution of the edgel location estimate is symmetric about the origin, which is the true edgel location. The distribution is not rotationally symmetric and is more concentrated along the true orientation of the gradient. As the SNR increases, the concentration is increased and the entire distribution shrinks toward the origin. For low SNR, as is the case shown in the figures, the variation in the orientation estimate is very large.

We note that the level contour of the pdf of the 2-D edgel location as shown in Figure 5.3 has two very significant concavities in the direction perpendicular to the gradient orientation. This is due to the fact that the orientation estimate is more concentrated around the true orientation. This behavior should be common to all reasonable edgel operators whose errors in orientation and 1-D radial location estimate are symmetric about the true values.

Since the 2-D Cartesian location distribution is center-symmetric about the true edge location, the Cartesian coordinates computed from $\hat{\theta}$ and \hat{v} are unbiased estimates. Analytical derivation of the covariance matrix of this distribution is difficult, but the numerical values can be computed easily from the spatial spread of the pdf. The trace of this covariance matrix is a good measure of the edgel localization performance. When the SNR increases, the spread of the pdf shrinks and results in better edge localization performance. When there is no underlying edge structure, i.e., when the SNR reduces to 0, then there is no preference of any orientation for the non-existent edgel. In this case, the 1-D radial location estimate follows the Cauchy distribution. The pdf of the 2-D Cartesian edgel location estimate is then rotationally symmetric. Although theoretically the Cauchy distribution does not have finite mo-

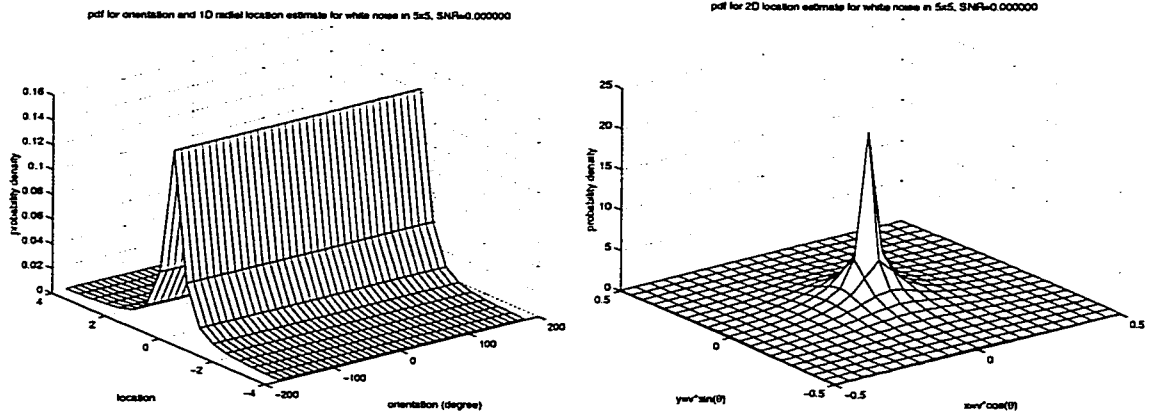
(a) pdf for joint distribution of θ and v

(b) pdf for 2-D location

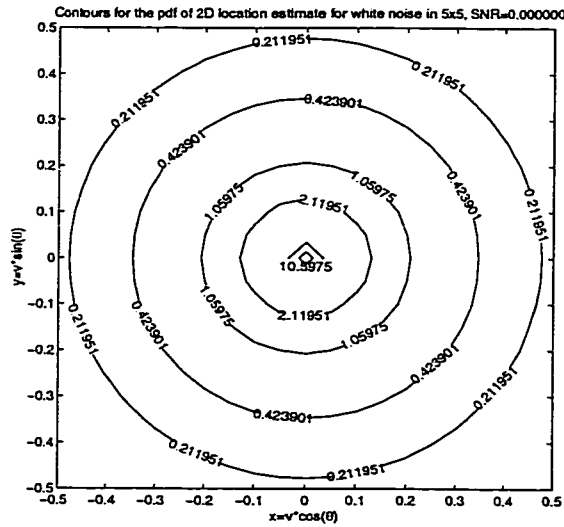


(c) level contour of the surface in (b)

Figure 5.3: The pdf of the joint distribution of orientation and 1-D radial location, and probability of the 2-D Cartesian location. Estimated by the IDDGO. Underlying image data is the unit-contrast step edge in the 5×5 neighborhood, with 0° gradient orientation and $\Omega = 1$.

(a) pdf for joint distribution of θ and v

(b) pdf for 2-D location



(c) level contour of the surface in (b)

Figure 5.4: The pdf of the joint distribution of orientation and 1-D radial location, and probability of the 2-D Cartesian location as estimated by the IDDGO. Underlying image data is unit variance white Gaussian noise ($\Omega = 0$).

ments, we are only interested in the truncated version of it. Then it has all moments. The pdf for the 2-D location estimate in this case is shown in Figure 5.4.

5.3 Tabulating edgel location uncertainty

At a given input SNR, the pdf of the edgel location is used in the numerical evaluation of the edgel location covariance matrix. The edgel location uncertainty can be characterized by the trace or the larger of the two eigenvalues of the covariance matrix. These as functions of the input SNR are shown in Figure 5.5(a).

It is interesting to notice that, in the very low SNR range, as the input SNR grows, the larger eigenvalue does not monotonically decrease as one might have expected. Rather, it increases first! However, there is an explanation to this behavior. The most basic reason is that the variance of the orientation estimate responds to the increase in the input SNR earlier than the variance of the radial location estimate.

Consider the situation illustrated in Figure 5.5(b). When a point moves from P_1 to P_2 with its radial distance from the origin held constant, i.e., $|OP_1| = |OP_2|$, the projection on the x -axis increases with the move. At the same time the projection on the y -axis decreases. This behavior has a central role in the explanation for the “bump” in the curve for the variance along the x -axis.

As shown in Figure 5.5(c), when the input SNR initially starts to grow, the variance of the radial edgel location remains largely unchanged. This roughly corresponds to a constant radial distance from the origin. At this time, the variance of the orientation estimate starts to decrease quickly, i.e., the orientation estimate starts to concentrate toward the true gradient orientation which is along the x -axis. This roughly corresponds to a reduction in the angle from XOP_1 to XOP_2 . Therefore, there is an increase in the projection along the x -axis and a decrease in the projection along the y -axis. Since the variance along an axis is a weighted average of the square of the length of the projection, this results in an increase in the variance along the x -axis,

and a decrease in the variance along the y -axis.

As input SNR further increases, the variance of the radial location also starts to decrease quickly. This roughly corresponds a decrease in the length of OP_2 , which in turn causes the projection on the x -axis to decrease. This is when the variance along the x -axis starts to decrease. In the entire process, the projection along the y -axis decreases monotonically, which results in a monotonically decreasing variance along the y -axis.

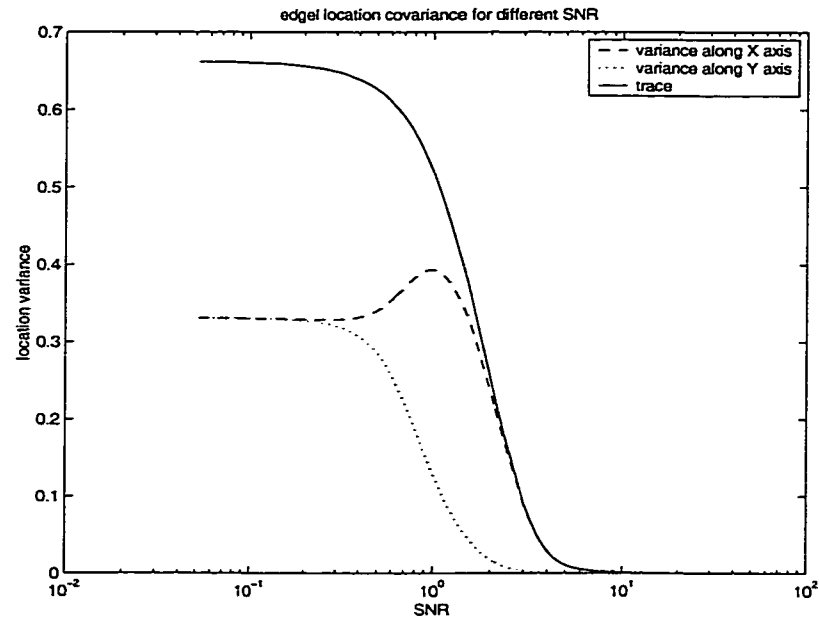
This explanation is made more concrete by quantitatively considering the following simplified situation. Consider a uniform distribution in the shaded area of a circular disk shown in Figure 5.6(a). The area of the shaded area is

$$\frac{1}{2}\rho^2 \times (2\pi - 4\theta) = (\pi - 2\theta)\rho^2$$

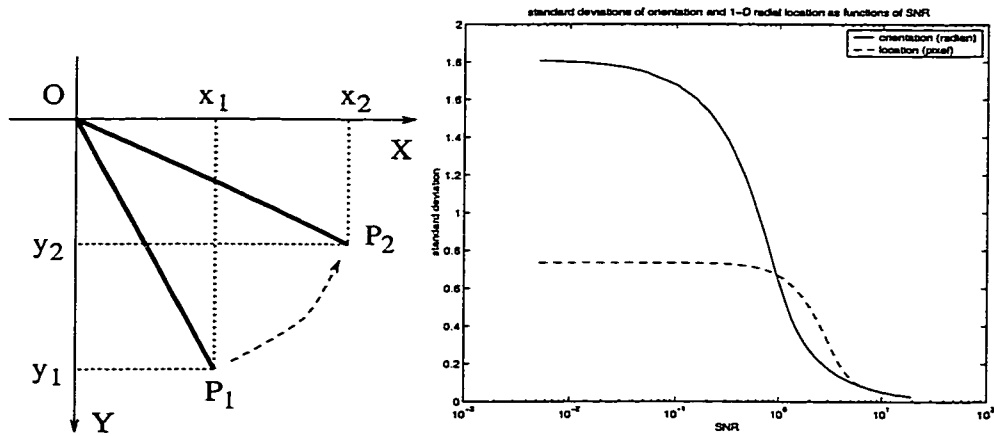
with the unit of θ being the radian. The variance along the x -axis is

$$\begin{aligned} V_x^2 &= \int_{-\frac{\pi}{2}+\theta}^{\frac{\pi}{2}-\theta} \int_0^\rho \frac{(r \cos t)^2}{(\pi - 2\theta)\rho^2} r dr dt + \int_{\frac{\pi}{2}+\theta}^{\frac{3\pi}{2}-\theta} \int_0^\rho \frac{(r \cos t)^2}{(\pi - 2\theta)\rho^2} r dr dt \\ &= 2 \int_{-\frac{\pi}{2}+\theta}^{\frac{\pi}{2}-\theta} \int_0^\rho \frac{r^3 \cos^2 t}{(\pi - 2\theta)\rho^2} dr dt \\ &= \frac{2}{(\pi - 2\theta)\rho^2} \cdot \frac{\rho^4}{4} \cdot \int_{-\frac{\pi}{2}+\theta}^{\frac{\pi}{2}-\theta} \frac{1 + \cos 2t}{2} dt \\ &= \frac{\rho^2}{4(\pi - 2\theta)} (\pi - 2\theta + \sin 2\theta) \\ &= \frac{\rho^2}{4} \left(1 + \frac{\sin 2\theta}{\pi - 2\theta} \right) \end{aligned}$$

This as a function of ρ and θ is plotted in Figure 5.6(b). For a fixed ρ , V_x^2 increases as θ increases. However, V_x^2 is more affected by the change in ρ and decreases quickly as ρ decreases. In the 2-D edgel location uncertainty case, as shown in Figure 5.5(c), when the SNR initially increases, the orientation estimate immediately starts to concentrate toward the true orientation, which roughly corresponds to a increase in the θ here. The variance of the radial location estimate stays at the same level, which roughly corresponds to a fixed ρ here. Hence the variance along the x -axis starts to grow.



(a)



(b)

(c)

Figure 5.5: Uncertainty of the edgel location estimate. Orientation estimated by the 5×5 IDDGO. (a) 2-D edgel location uncertainty as functions of input SNR; solid – trace of location covariance matrix, dashed – variance along the x -axis, dotted – variance along the y -axis; (b) illustration for explaining the “bump” in the variance along the x -axis; (c) standard deviations of orientation (solid curve) and radial location (dashed curve) estimates as functions of input SNR.

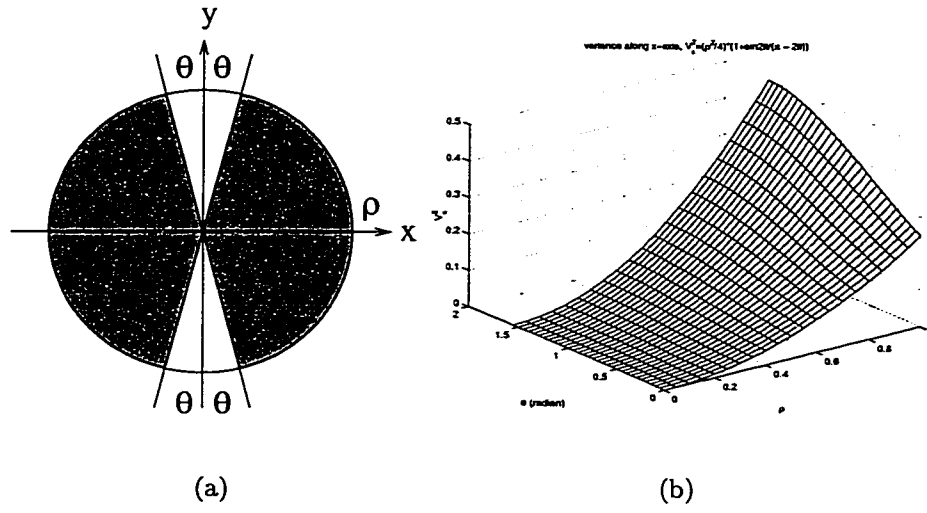


Figure 5.6: Simplified situation that explains the increase in edgel location uncertainty along the true gradient orientation. (a) Consider the variance V_x^2 along the x -axis of a uniform distribution over the shaded region. (b) V_x^2 as a function of ρ and θ .

When the SNR further increases, the variance of the radial location estimate starts to decrease quickly, roughly corresponding to a decreasing ρ . This causes V_x^2 to decrease.

The correlation between the x and y elements of the edgel location is zero for the 0° step edge. For each SNR value of interest, we numerically compute the covariance matrix and tabulate the variance in the x and y directions.

Notice that, since the Cauchy distribution does not have finite moments, the covariance matrix for the 2-D location does not exist for $\Omega = 0$. Practically, although the distribution is long tailed, we are mostly interested in its behavior in a finite neighborhood around the origin. The distribution truncated in this finite neighborhood is used for the covariance matrix evaluation. In our experiments, the circular neighborhood of radius $3 * \frac{3\kappa_3}{\kappa_2} = 8.8721$ is used, which corresponds to the 1-D radial location being within the range $[-3, 3]$.

The tabulated covariance matrices are used for computing the quantities $\text{Cov}(\Delta\mathbf{x}, \Delta\mathbf{x})$,

$\text{Cov}(\Delta\mathbf{y}, \Delta\mathbf{y})$ and $\text{Cov}(\Delta\mathbf{x}, \Delta\mathbf{y})$ in Equations (3.15), (3.17) and (3.18). The orientation θ and SNR at each boundary point are first estimated by the IDDGO. The SNR is used to index into the table made here and retrieve the two diagonal elements of the covariance matrix for that SNR level. This diagonal matrix is then pre- and post-multiplied by the rotation matrix for the orientation θ . The resulting matrix is the covariance matrix for the boundary point in question, and is used subsequently in the computation of the target centroid uncertainty.

5.4 Validation experiment and correlated noise

In the previous parts of this chapter, we theoretically derived the distribution of the edgel location estimate which is subject to error due to the noise in the observed image. The development studies the location estimate of the sharp step edge and assumes the additive white Gaussian noise. The result of the development is the relationship between the edgel location covariance matrix and the input SNR. In this section, we describe the experiments for verifying the validity of that relationship, and for examining the accuracy of that relationship when the white noise assumption is violated.

5.4.1 Validation experiment for the white noise

We first want to verify the relationship between the edgel location covariance and the input SNR when all assumptions for the development are satisfied. This is done by conducting simulation experiments and comparing the observed edgel location covariance with the values given by the relationship that appears in the previous section.

The experiment is conducted for the sharp step edge with the 0° orientation in the 5×5 neighborhood. Noisy observations are simulated by adding white Gaussian noise, the variance of which is varied to give different input SNR levels in different

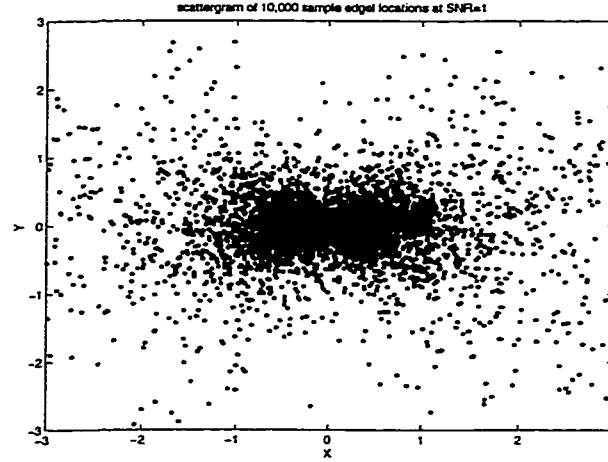
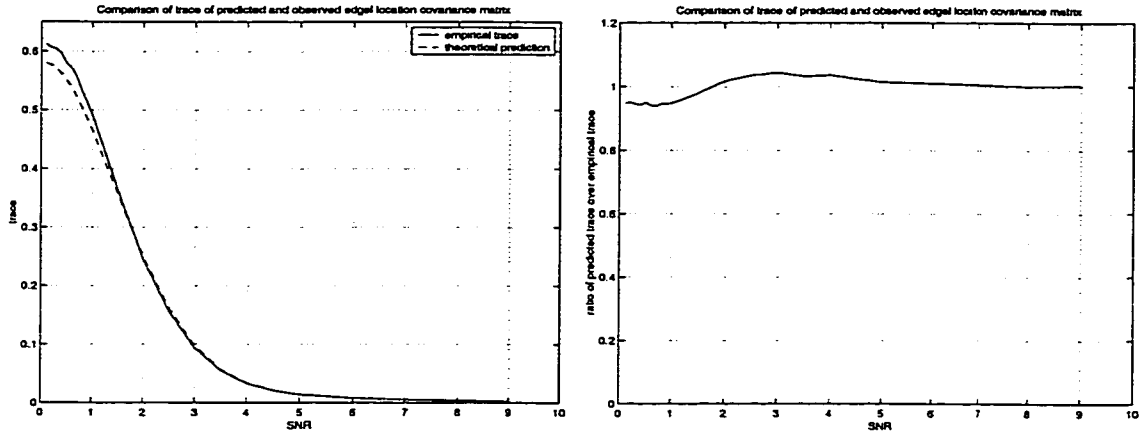


Figure 5.7: Empirically observed scatter of edgel location estimate.

runs. For each noisy observation, the IDDGO based edgel location estimator is used to obtain a noisy estimate of the edgel location. For each input SNR level, a large number (10,000,000) of noisy estimates of the 2-D edgel location are obtained. The estimates are constrained to be within the circular neighborhood of radius 3 around the center of the 5×5 neighborhood. Figure 5.7 shows an example of the scattergram of the noisy estimates for SNR=1. We notice the similarity between the spatial distribution of the estimated edgel locations and the theoretically derived probability shown in Figure 5.3. The sample covariance matrix of the estimates are computed. These are repeated for a number of input SNR levels. The result is the empirically observed relationship between the edgel location covariance and the input SNR. Since the sample size (10,000,000) in the experiment is very large, the sample variation in the observed location covariance matrix is fairly small, typically less than 1%.

Due to the fact that the edgel location distribution is very different from the bivariate normal distribution, the statistical tests [72] specially designed for comparing multi-variate normal distributions cannot be used to access the closeness of the empirically observed covariance to the predicted. Therefore, we subjectively compare



(a) trace of empirical and predicted covariance matrix

(b) ratio of the trace

Figure 5.8: Comparison of empirical and predicted edge location uncertainty.

the empirical relationship with the predicted by the plots of the trace (the sum of the two diagonal elements) of the covariance matrix.

The plots are shown in Figure 5.8. It is clear from the plots that the theoretically predicted relationship between the location uncertainty and the input SNR is very close to the empirically observed curve. However, there is noticeable difference between the two. Specifically, for SNR less than 2, the theoretically predicted value is an under-estimate of the true edge location uncertainty, with maximum deviation of about 6% of the empirically observed value. This maximum deviation occurs at SNR=0.5. For SNR between 2 and 6, the predicted value is an over-estimate, with maximum deviation of about 4% of the empirical value, occurring at SNR=3. The difference between the curves comes from the approximation error that is discussed in Section 5.2.2. That approximation essentially assumes the independence of the 1-D radial location estimate \hat{v} from the orientation estimate $\hat{\theta}$. Since at higher SNR $\hat{\theta}$

concentrates more and more around the true orientation θ_0 , the approximation error becomes less and less noticeable. (We observe some very small differences between the empirical and predicted values at very high SNR levels. These are due to the quantization error in the computation of the theoretical values. It is not caused by any approximation or error in the modeling.)

In order to reduce the approximation error at the low SNR range, more careful analysis is needed in the computation of $f(\hat{\theta}, \hat{v}|\theta_0, v_0)$, which needs to deal with the dependence of \hat{v} on $\hat{\theta}$. This is a difficult task. Also, we consider that level of accuracy is not very necessary in our FLIR ATR algorithm framework. Here the relationship between the location uncertainty and the input SNR is simply used for converting the local estimate of the input SNR into a form suitable for use in the weighted boundary contrast combination. Therefore, it is mainly the curve's general trend that is of more importance. The small amount of prediction error does not seem to cause much harm in the performance of the entire FLIR ATR algorithm as observed in our experiments.

5.4.2 *Effect of the correlated noise*

Although the independent noise model is assumed in the theoretical development in the earlier parts of this chapter, violation of this assumption has the potential to cause very significant error in the theoretical prediction. Here we give the experiment results that studies the impact of the correlation in the noise on the relationship between the edgel location uncertainty and the input SNR.

The experiment is conducted similar to that in the previous section. However, correlated noise instead of white noise is used in simulating the noisy observation. We assume the correlation between the noise values at different pixel locations decays exponentially. The rate of decay is controlled by the parameter d_0 . If two pixels are of distance d apart, the correlation coefficient between the noise values at these two pixel locations is $c = e^{-d/d_0}$. A number of values are chosen for d_0 so that the correlation between adjacent pixels, which is computed as $c = e^{-1/d_0}$, takes the values

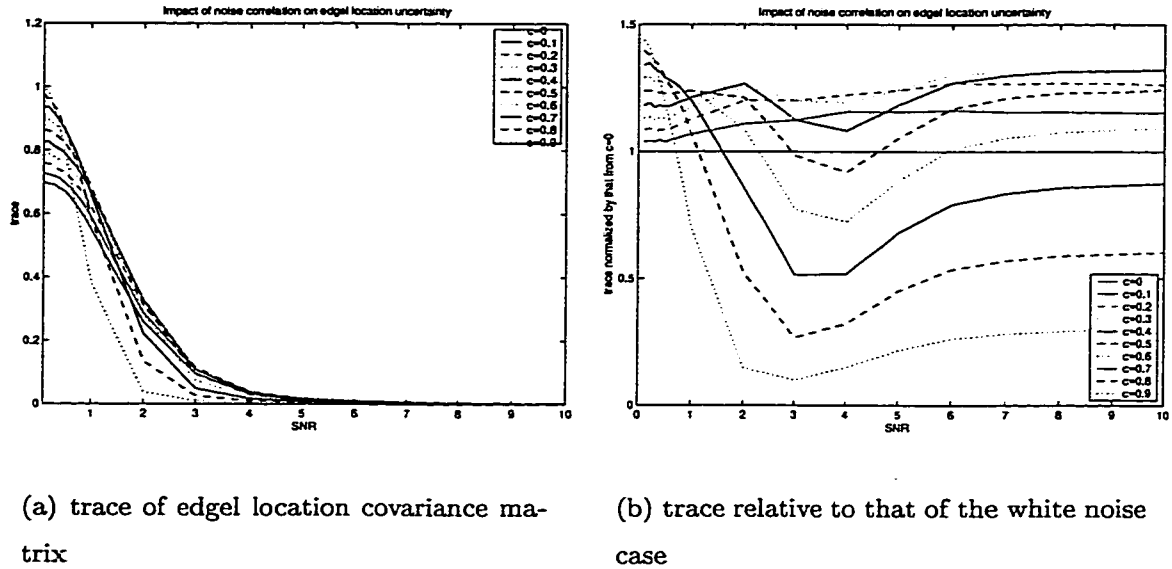


Figure 5.9: Impact of noise correlation on the relationship between the edgel location uncertainty and the input SNR.

0.1, 0.2, ..., 0.9.

For each value of the correlation coefficient c , the sample covariance matrices for the estimated edgel location are obtained for a series of input SNR levels. We plot for each c a curve showing the empirical relationship between the observed edgel location uncertainty and the input SNR. These plots are shown in Figure 5.9.

From these plots, we see that for various levels of correlation, the curves more or less share the similar general trend. For the correlation coefficient up to about 0.5, the general effect of the correlation in the noise is to increase the edgel location uncertainty. For stronger correlation, the edgel location uncertainty is increased for the lower SNR levels, and decreased for the higher SNR levels. The cross-over point moves toward lower SNR values as the correlation becomes stronger. Through out the range of the correlation, the slope of the curves in general becomes more steep as the correlation becomes stronger. For very strong correlation ($c > 0.6$), the location

uncertainty stays lower than that with the uncorrelated case after the curves cross each other at some moderate SNR levels (around 1 to 3).

From this experiment, we see that correlation in the noise does in general have a quite noticeable influence on the relationship between the edgel location uncertainty and the input SNR. This influence can be significant for applications where the accuracy of the location uncertainty prediction is required. The understanding of effect of noise correlation is in general a very difficult task. It is also the case here. This will be a topic of the future research. Before further understanding is achieved, we can use the empirical curves in Figure 5.9 to roughly predict the resulting edgel location uncertainty from some properly estimated correlation coefficient. Also, due to the particular way in which the location uncertainty and SNR relationship is used in our FLIR ATR algorithm and due to the similar general trend shared by the curves, we suspect noise correlation up to 0.3 may not cause very significant degradation of the performance of the entire FLIR ATR algorithm.

5.5 Effect of slope and displacement of the zero-crossing

5.5.1 The requirement of negatively sloped zero-crossing

We have identified the edgel location to be the zero-crossing of the second directional derivative in the direction of the gradient. In the edge detection application, it has been pointed out that the authentic step edge structures have negatively sloped zero-crossings [60, 25]. Due to image noise, the observed slope might become positive even if the underlying noise-free image intensity function does represent an authentic step edge structure.

In this section, we show that this requirement on the zero-crossing being negatively sloped can cause problems when the input SNR is low. Furthermore, for the purpose of target boundary contrast combination, this requirement is not necessary and edgel location uncertainty formulation need not concern the slope of the zero-crossing.

In the cubic facet model, the sign of the slope at the zero-crossing of the second derivative is determined by the sign of \hat{c}_3 in Equation 5.4. As already mentioned, for the authentic step edge structure, \hat{c}_3 is distributed as $n(c_3^*\Omega\sigma, \kappa_3^2\sigma^2)$. Let X be random variable distributed as $n(0, 1)$. Then \hat{c}_3 can be written as $\kappa_3\sigma X + c_3^*\Omega\sigma$. The probability that the slope is positive can be computed as

$$\begin{aligned} P(\hat{c}_3 > 0) &= P(\kappa_3\sigma X + c_3^*\Omega\sigma > 0) \\ &= P(X > -\frac{c_3^*}{\kappa_3}\Omega) \end{aligned}$$

In the 5×5 neighborhood, $c_3^* = -0.0833$ and $\kappa_3 = 0.1179$, and the probability is computed as

$$P(\hat{c}_3 > 0) = P(X > 0.7068\Omega) = \Phi(-0.7068\Omega)$$

where $\Phi(\cdot)$ is the cumulative distribution function (cdf) of the standard normal distribution. This probability is plotted in Figure 5.10. From the plot, we see that for the low SNR range, there is very significant probability that the observation noise can make the zero-crossing of an authentic step edge into positively sloped. If requirement for the negative slope is enforced in collecting the target boundary contrast evidence, a significant portion of the evidence will be rejected due to the wrong sign of the slope. Since the imagery in the FLIR ATR application is typically of low SNR, the enforcement of the negative slope requirement is problematic. In this situation, hard decisions performed on the pixel level are intrinsically unreliable and should be avoided altogether. Therefore, in the ATR application, we do not perform target boundary point detection. Instead, we use the centroid uncertainty model to combine evidence along the hypothesized target boundary. Not enforcing the negative slope requirement increases the noise robustness and reduces misdetection probability in the evidence collection at true target locations. This, however, may tend to increase the probability of false alarm at non-target locations.

On the other hand, the motivation for the negative slope in the edge detection application is to avoid detecting certain high-contrast details in the image which do

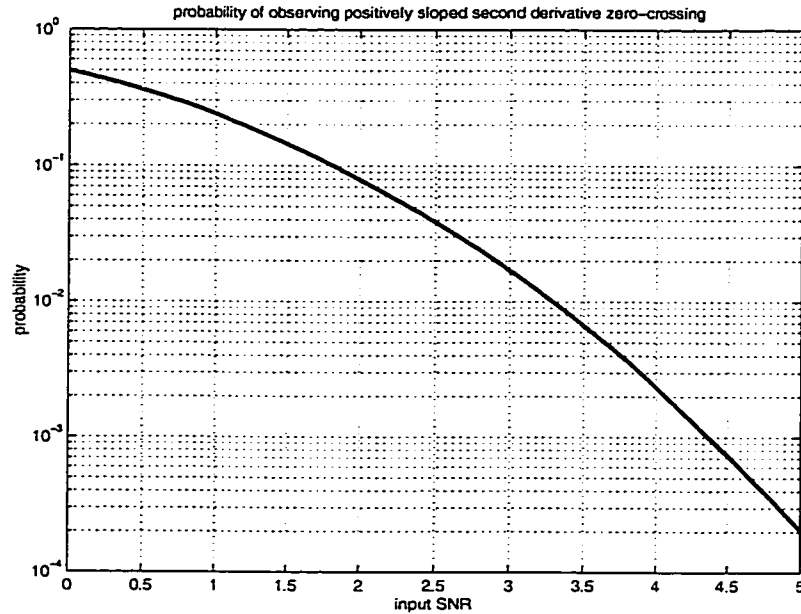


Figure 5.10: Probability for observing positively sloped zero-crossing for the authentic step edge.

not possess the correct intensity pattern. However, this is not as important in the FLIR ATR application. In this application, the existence and location of high contrast parts are far more important than their intensity patterns. High-contrast image neighborhoods with positively sloped zero-crossings also serve well as the evidence for target boundary contrast. Since the location uncertainty is the same for both positively and negatively sloped zero-crossings, the amount of evidence for the target boundary contrast is also the same. Therefore, the sign of the slope is not important in the edgel location uncertainty model.

5.5.2 Increased location uncertainty for the shifted edgel location

The development and computations in the previous parts of this chapter assumes the true 1-D radial location of the edgel is at the center of the neighborhood. Will the

location uncertainty be affected if the edgel is moved away from the center? This question is answered by the following experiment.

In the 5-point 1-D neighborhood, the noise-free unit-contrast sharp step edge is represented by

$$-\frac{1}{12}(x - x_0)^3 + \frac{7}{12}(x - x_0) = -\frac{1}{12}x^3 + \frac{x_0}{4}x^2 + \left(\frac{7}{12} - \frac{x_0^2}{4}\right)x + \frac{x_0^3 - 7x_0}{12}$$

where x_0 is the true location of the edge. Now c_2^* is $x_0/4$ instead of 0. The noisy estimate of the 1-D radial location of the edgel is

$$\hat{v} = -\frac{\kappa_2}{3\kappa_3}Z = -\frac{0.1196}{3 \times 0.1179}Z = -0.3381Z$$

where Z is distributed as

$$\frac{n(\frac{c_2^*}{\kappa_2}\Omega, 1)}{n(\frac{c_3^*}{\kappa_3}\Omega, 1)} = \frac{n(\frac{x_0}{4 \times 0.1196}\Omega, 1)}{n(-\frac{1}{12 \times 0.1179}\Omega, 1)} = \frac{n(2.0903x_0\Omega, 1)}{n(-0.7068x_0\Omega, 1)}$$

The variance of \hat{v} as a function of x_0 for $\Omega = 1$ is shown in Figure 5.11. It is clear that as x_0 moves away from 0, the variance of the 1-D radial edgel location estimate increases monotonically. In the 2-D case, when the edgel moves away from the center of the neighborhood, the variance of the orientation estimate also increases. Therefore, the 2-D edgel location uncertainty increases as the true location moves away from the center of the neighborhood.

5.6 Summary

We studied the estimation error in the gradient direction and the zero-crossing of the second directional derivative. Their joint distribution is used to characterize the behavior of the edgel location estimate in the 2-D Cartesian space. An approximation in the joint distribution allows simple analytical expressions to be written for the probability density function. The error from the approximation is well tolerable for our application. The covariance matrix of the distribution describes the uncertainty

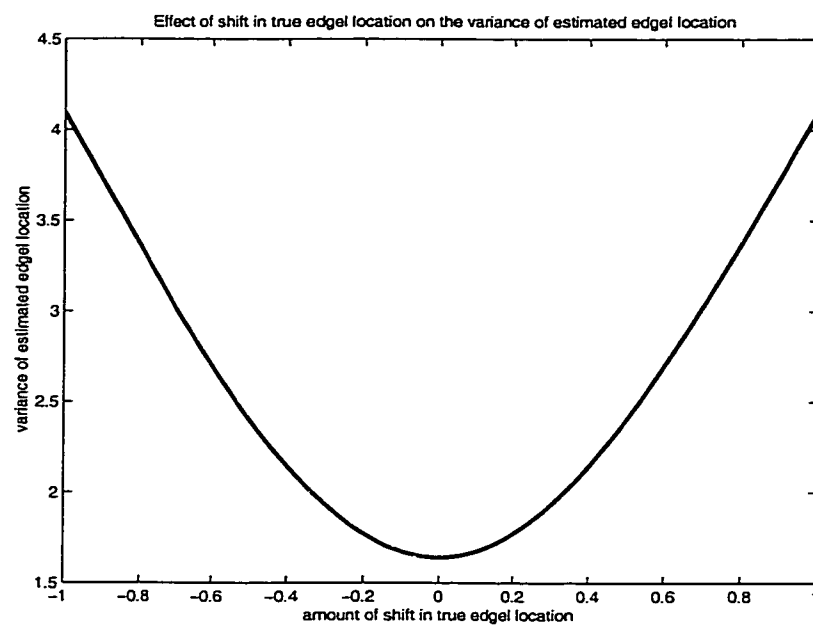


Figure 5.11: Edgel location uncertainty increases as the true edgel location moves away from the center of the neighborhood.

in the edgel location estimate and is strongly affected by the input signal-to-noise ratio. The covariance matrices for a range of input SNR of interest are obtained via the numerical evaluation. They are tabulated for convenient use in the centroid uncertainty based FLIR ATR algorithm for computing the boundary point location uncertainty.

The validity of theoretically computed location uncertainty is verified in the experiment, where good agreement is observed between the empirically observed and theoretically computed values. The effect of noise correlation is found to increase location uncertainty at all SNR levels when the correlation is not very strong. Strong correlation is observed to increase location uncertainty at low SNR levels and decrease it at high SNR levels.

Different from the edge detection application, the sign of the slope of the zero-crossing in the second directional derivative in the gradient direction need not be negative for the purpose of contrast evidence accumulation in the FLIR ATR. The displacement of edgel from the center of the neighborhood is found to increase the location uncertainty.

The main purpose of this chapter is to find the relationship between the edgel contrast and location uncertainty. However, the resulting characterization on the edgel location uncertainty is quite useful in other basic topics in computer vision as well. Not only are accuracy and precision of edge location important in themselves in measurement oriented applications [6, 115], the edge localization plays important roles in computer vision algorithms such as the Hough transform [7, 46, 83, 74] and the edge linking process.

The derivations in this chapter closely parallel the line of reasoning of Marimont and Rubner [98]. We became aware of their work only after we finished our derivation and experiments. Also, our work is different from [98] in that we do not use the resulting edgel location uncertainty in low level edge pixel detection. As has been stated, this is highly prone to error due to the noisy nature of local neighborhood

operations. Instead, we explicitly compute the edgel location uncertainty as indexed by the signal-to-noise ratio (SNR). The location uncertainty along the target boundary is combined in the polygon centroid uncertainty model, hence we avoid making hard decisions at the pixel level. The edgel location distribution plays more of the role of the medium for conveying the contrast into another form which is more appropriate for evidence accumulation.

In applying the result in edge detection, we believe the edge detection problem should not be formulated as a classification problem on the pixel level. Instead, a Bayesian formulation should be used which optimally combines the information from higher level edge structure and local image measurements, including contrast, edgel orientation and location as well as their expected uncertainty. This is a direction of future research.

Chapter 6

CONSTRAINED MATCHING PROBLEM IN PERFORMANCE EVALUATION

6.1 *Introduction*

Performance evaluation in computer vision is an important and active research area [54, 57, 122, 15, 24, 62, 16, 142, 75]. Both theoretical and empirical approaches are being taken by different research groups. In empirical performance evaluation [15, 24, 62, 16, 142, 75], one major approach is to get ground-truth for some test image data and compute objective performance measures by comparing the algorithm output with the ground-truth. This appear to be the only accepted approach for practical ATR system performance evaluation.

In characterizing ATR algorithm performance [32, 147, 76, 12, 27, 119, 102, 117, 14, 36, 29], it is customary to determine the detection rate, false alarm rate, as well as the recognition rate for the correctly detected targets. These values need to be determined by properly comparing the algorithm output against the ground-truth. Often there are multiple ground-truth target instances in a single test image, and an ATR algorithm may also produce multiple target declarations on that image. The declared targets may or may not be close enough to ground-truth targets, and the ones close enough may or may not have the right identification of the target type. Also, algorithms may claim more than one target around a single ground-truth target, and may also claim targets around clustered ground-truth targets where it is not obvious which ground-truth target(s) should be associated with the declarations.

Faced with this complicated situation, we need to develop a concrete and con-

sistent technique that unambiguously and reasonably classifies ground-truth targets into detected and misdetected targets, and classifies ATR algorithm target declarations into correct declarations and false alarms. Also, for the detected ground-truth targets, it discriminates those that are correctly recognized and those that are not.

In this chapter, we use the optimal matching problem to model the situation. A simpler scenario is first considered which is the performance evaluation of edge detection. The main development of how an optimal matching problem is set up, transformed into an optimal assignment problem, and solved by the efficient Hungarian algorithm is carried out in a discussion of the edge detection performance evaluation. This enables a simpler and clearer description of the essence of the problem. After this is accomplished, the procedure for ATR performance evaluation is described and an example is presented. The performance evaluation result using this procedure is seen to be precise, consistent and conceptually appealing.

6.2 Performance evaluation problems in edge detection

In assessing the performance of edge detectors [19, 51, 107, 146, 62, 75], we care about the detection rate, false alarm rate and the average localization error for the correctly detected edge pixels. A performance assessment technique needs to classify the edge pixels in the ground-truth into two distinct classes, namely detected and misdetected edge pixels, and to classify the edge pixels in the declared edge map into two distinct classes, namely correct declarations and false alarms. To tolerate a certain amount of localization error, ground-truth edge pixels and declared edge pixels do not have to be at exactly the same pixel location for them to be declared as detected ground-truth edge pixels and correct declarations, respectively. However, each detected ground-truth edge pixel needs to be associated with at least one of the correct declarations, and vice versa. Because of the relaxed requirement on the localization, the classification and association between ground-truth edge pixels and

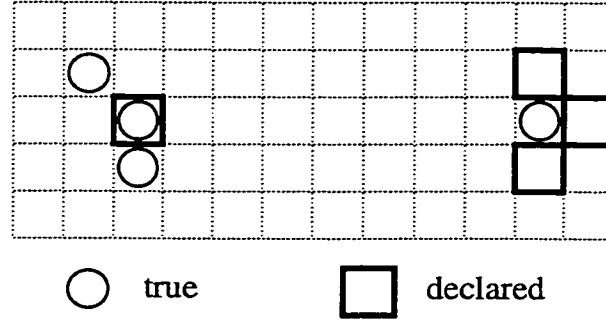


Figure 6.1: An example where distance transform based method fail to give accurate performance measures.

declared edge pixels are not trivial problems. Different ways of doing these leads to different performance measures.

A distance transform based technique [62, 16] has been customarily used for this purpose. A distance map is first obtained where each pixel location is assigned a value which is the distance from this pixel location to the closest ground-truth edge pixel. A threshold on this distance is chosen. The correct declarations are all those edge pixels in the declared edge map which have a distance value not larger than the threshold. The detected ground-truth pixels are those which have declared edge pixels within their neighborhoods. The size of the neighborhood is also determined by the threshold on the distance. The problem with this technique is that it allows multiple-to-one and one-to-multiple correspondence between ground-truth and declared edge pixels. Consider the example shown in Figure 6.1, where ground-truth edge pixels are marked by circles, and declared edge pixels are marked by solid squares. According to distance transform based method, all ground-truth pixels are detected and there is no false alarm declarations.

Intuitively, one would want to have an exact one-to-one correspondence between detected ground-truth edge pixels and correct declarations. If this is established, the numbers of detected and misdetected ground-truth edge pixels and the numbers of

correct declarations and false alarms will be more informative of the edge detection performance. According to this consideration, for the example in Figure 6.1, only two ground-truth pixels, the one in the middle on the left half and the one on the right are regarded as being detected, and the other two on the left half are regarded as being misdetected. Similarly, only two of the declared pixels, the one on the left half and one of the three on the right (which one does not matter in this example), are regarded as correct declarations. The rest two on the right half are regarded as false alarm declarations. Therefore, the more appropriate performance evaluation method should declare two misdetections and two false alarms.

Here we propose to establish the desired one-to-one correspondence by solving a combinatorial optimal matching problem. This problem is transformed into an unconstrained optimal assignment problem, which can be efficiently solved by the Hungarian algorithm [130, 77]. The solution yields a maximal one-to-one correspondence between ground-truth edge pixels and declared edge pixels. Performance measures based on this correspondence follow the intuition and are more informative of the edge detection performance.

In our experiment, we choose to use a synthetic image to show the difference between two evaluation techniques. Although the use of synthetic imagery in comparing the performance of edge detectors (using a chosen evaluation technique) is questionable [62], it is appropriate in comparing particular performance evaluation techniques. This is because it allows a clear and unambiguous judgment of edge detection quality by human observers with minimal inter- and intra-observer variations. This judgment acts as a gold standard to which the output of different performance evaluation techniques should be compared. The best performance evaluation technique is the one whose output is closest to that judgment.

6.2.1 A note on the relevance of performance characterization for edge detection modules

The following development of the edge detector performance evaluation procedure not only serves as an example for how to establish a one-to-one correspondence in the general performance evaluation framework, it is also very important in its own right. This is due to the wide presence of edge detection modules in various computer vision systems, and to the numerous edge detection techniques proposed in the literature.

Since edge detection is usually at earlier stages of computer vision algorithms and is almost never the final goal of any practical system, there is legitimate concern about its performance characterization without the context of a particular vision system. We agree that, due to the many different ways in which edge detection output is used by subsequent vision algorithm modules, the impact of edge detection quality on the entire vision system performance will vary among different vision systems. However, we still insist on the need for a general framework for precise characterization of edge detection quality by a set of quality measures general enough to be used in as many different applications as possible, yet still carrying enough specific information to be useful. For most applications, we believe that the detection rate, false alarm rate, and average location error constitute a good set of measures for this purpose.

The theory for the computation of such measures can either be applied directly or used as the guideline for developing more specific performance characterization schemes for particular applications. To this end, we propose the use of a one-to-one correspondence and develop a procedure for establishing it. The set of quality measures computed from the theory is essential in studying the sensitivity of the performance of the entire vision system to the edge detection module. Therefore, besides their being relevant to the particular vision system under consideration, the preciseness of the measures is highly desirable. The preciseness here refers to the sensitivity of the values of the measures to changes in the edge detector output. As

will be pointed out in the following experiments, the performance measures computed from the methodology proposed here is much more precise than the distance-transform based technique being customarily used. Although this level of preciseness might be so high that it is not even necessary for certain computer vision systems, due to the relative insensitivity of the vision system performance to the edge detection module quality, it is comforting to know that a very high precision, the cost of which is very much tolerable, has been achieved.

6.3 *Unconstrained optimal assignment problem*

In the assignment problem, one is concerned with establishing a full one-to-one correspondence between two discrete sets K and L , both of which have N elements. An assignment is a one-to-one mapping $a : K \rightarrow L$.

Let the cost for associating $k \in K$ with $l \in L$ be $q(k, l)$. The total cost of an assignment a is

$$\sum_{k \in K} q(k, a(k)) \quad (6.1)$$

The optimal assignment is a mapping $a : K \rightarrow L$ which yields the smallest cost. The assignment problem is to find such an optimal assignment. If there is some constraint between the k 's and the l 's, a constraint which prohibits certain pairings, then the problem becomes the constrained optimal assignment problem. Otherwise, it is the unconstrained problem.

If $q(k, l)$ is finite for all $k \in K$ and $l \in L$, the optimal solution to the unconstrained problem always exists, but might not be unique. However, from the optimization point of view, all optimal solutions are equivalent.

For an integral or rational valued cost function $q(\cdot, \cdot)$, the optimal assignment can be found by the Hungarian algorithm [130, 77]. This is a numerical search algorithm that guarantees to arrive at one optimal solution. The computational complexity is $O(N^3)$ [77].

In the context of edge detection performance evaluation, K is the set of ground-truth edge pixels, and L is the set of declared edge pixels. The cost $q(k, l)$ is a meaningful measure of the distance between a ground-truth edge pixel k and a declared edge pixel l . Different forms of the distance measure can be used, which are usually reasonable variations of certain distance measures, e.g., the L_1 norm (city-block distance), L_2 norm (Euclidean distance), and L_∞ norm. In our experiments, the squared Euclidean distance is used.

Most likely, the numbers of ground-truth edge pixels and declared edge pixels are not equal. This does not cause a problem. Conceptually, we can think of adding “ghost” pixels to the set which has smaller number of elements. The distance from any ghost pixel to all pixels in the other set is assigned a finite but very large value. This way, the ghost pixels will not compete with any of the original pixels in the assignment. In the end, all ghost pixels will also be assigned to pixels in the other set. All those pixels that are the counterparts of the ghost pixels are considered unmatched, and therefore are either misdetected ground-truth edge pixels (if the ghost pixels are added to the declared edge pixel set) or false alarms in the declared edge map (if the ghost pixels are added to the ground-truth edge pixel set.)

6.4 The Hungarian algorithm

The unconstrained optimal assignment problem is essentially determined by the cost function $q(\cdot, \cdot)$. Exhaustive search for the optimal assignment would require $O(N!)$ computations of the total cost in (6.1). Fortunately, this need not be done.

Let the cost function be represented in a matrix form. Adding a constant to any row of the matrix does not change the the optimal assignment. (Although the total cost of the resulting optimal assignment gets changed, it is an optimal assignment, not its associated total cost, that is of interest to us.) Similarly, adding a constant to any column of the matrix does not change the solution, either. The Hungarian

algorithm [80, 130, 77] works by adding suitable constants to some properly chosen rows and columns of the matrix, and finding the maximum number of independent zeros in all equivalent forms of the cost matrix. Due to the König-Egerváry Theorem [78, 37, 130], this is equivalent to finding the smallest number of lines (rows and columns) to cover all zeros.

In this research, we take a black-box approach to the Hungarian algorithm. It is simply used as an efficient tool to solve the assignment problem. In performance evaluation, it is sufficient to know the existence of such a solution and to have available its software implementation. The details of the algorithm are not of major interest to this research and, therefore, are omitted here. Interested readers can find the details of the algorithm in [80, 130, 77] and the references therein.

In the situation where every element in K is allowed to be paired with every element in L , the optimal solution always exists. If all entries in the cost matrix are integral or rational numbers, the Hungarian algorithm is guaranteed to arrive at one optimal solution with finite iterations. The Knuth implementation [77] of the algorithm handles the integral cost matrix. The number of elements in K and L need not be equal. The computational complexity is $O(m^2n)$ where m is the smaller of the cardinalities $|K|$ and $|L|$, and n is the larger.

6.5 Constrained case — optimal matching problem

In edge detection performance evaluation, we know that corresponding ground-truth edge pixels and declared edge pixels cannot be spatially very far apart. It makes sense to set up a threshold for the maximum tolerable localization error, i.e., the maximum tolerable distance between a ground-truth edge pixel and a declared edge pixel that are to be paired together. The threshold on the distance between them can be conveniently set by the neighborhood size used in the edge detection. We want to

put a constraint on the solution of the optimal assignment problem:

$$q(k, a(k)) \leq \tau \quad \forall k \in K \quad (6.2)$$

where τ denotes the maximum tolerable distance. The constrained optimal assignment problem is then to find a mapping $a : K \rightarrow L$ which minimizes the cost in (6.1) and satisfies the constraint in (6.2).

Now the finiteness of $q(k, l)$ does not guarantee the existence of a solution to the constrained problem. However, in order to determine misdetections and false alarms in performance evaluation, we have to find some reasonable association, maybe compromised in some way, between the k 's and the l 's.

The constrained optimal assignment problem in its original form is not an appropriate model for our problem at hand. Our problem, however, can be well modeled by an *optimal matching problem*.

The matching problem is similar to the assignment problem in that it tries to establish a one-to-one pairing between elements in K and L . Here the sizes of K and L need not be equal. There is a compatibility relationship $H \subset K \times L$. Any $k \in K$ and $l \in L$ for which $(k, l) \notin H$ are not allowed to be paired together. Only a subset of K needs to be paired up with distinct elements in L . The matching problem is to determine a match with the largest number of pairs.

Naturally, for our problem at hand, the compatibility relationship is

$$H = \{(k, l) \in K \times L \mid q(k, l) \leq \tau\}.$$

For any particular match $m : K \rightarrow L$, let $K_s(m)$ be a subset of K containing all the members for which the constraint is satisfied.

$$K_s(m) = \{k \in K \mid q(k, m(k)) \leq \tau\}. \quad (6.3)$$

The rank of a match m is defined as the cardinality (number of elements) of $K_s(m)$.

$$r(m) = |K_s(m)|. \quad (6.4)$$

In edge detection evaluation, we are only interested in the largest ranking matches, which are called the *maximal matches*. Let M be the set of all possible matches between K and L subject to the compatibility relationship H . Then the set of maximal matches can be formally written as

$$M_m = \{m \mid r(m) = \max_{n \in M} r(n)\}. \quad (6.5)$$

The cost for such a maximal match m is defined as

$$\sum_{k \in K_s(m)} q(k, m(k)) \quad (6.6)$$

The optimal matching problem is to minimize this cost over all maximal matches in the set M_m of Equation (6.5).

In the edge detection evaluation application, the solution to the optimal matching problem gives the largest possible number of associated pairs between ground-truth and declared edge pixels. Among all the different ways for making such a match, it picks the one with the smallest localization error. In a sense, it is trying to give the most positive interpretation of the declared edge map. This is the right attitude, since all computer vision algorithms using the edge map need to try their best to make the most positive use of it.

6.5.1 Solving the optimal matching problem

We now apply the idea of “ghost pair” to transform the optimal matching problem back to the unconstrained optimal assignment problem, solve the optimal assignment problem using the Hungarian algorithm, and apply some simple post-processing to enforce the constraint and obtain the solution to the optimal matching problem.

Let d denote a finite and very large value. For example,

$$d = N \times \tau \quad (6.7)$$

We selectively modify the cost by

$$q(k, l) = \begin{cases} d & \text{if } q(k, l) > \tau \\ q(k, l) & \text{otherwise} \end{cases} \quad (6.8)$$

This modified cost function is used to form an unconstrained optimal assignment problem. Each pair of elements whose cost gets changed to d is called a “ghost pair.” Since d is such a large value, this particular pair does not compete for each other against other elements in the assignment process. Notice that, normally each ground-truth and declared edge pixel is involved in many ghost pairs since there almost always exist edge pixels in the other class that are faraway from it. However, detected ground-truth edge pixels and correctly declared edge pixels are also involved in some pairs that are not ghost pairs. In the association process, it is these non-ghost pairs that are of interest to us. However, the provision for the ghost pairs is needed to change the hard constraint on the compatibility relationship into penalty terms in the total cost. This is necessary for the Hungarian algorithm to be used to find the solution.

The optimal solution to this new unconstrained problem exists due to the finiteness of the cost function. It can be found by the Hungarian algorithm. In the resulting solution, we examine again the cost between the assigned pairs, to enforce the compatibility relationship. The pairs whose cost are not larger than τ are good associations. They give correspondence between the involved detected ground-truth edge pixels and correct declarations. The cost of each of the rest of the pairs is larger than τ , and these must all be ghost pairs.

Remember that the original distances between the two elements in a ghost pair is larger than the threshold. In edge detection performance evaluation, we do not allow pixels that are farther apart than the threshold to be associated with each other. The two elements in a resulting ghost pair are then determined to be a misdetected ground-truth edge pixel and a false alarm in the declared edge map.

6.6 Association procedure

It is now clear that the association of ground-truth edge pixels with the declared edge pixels can be obtained with the procedure described below. The result of the association also determines the misdetected ground-truth edge pixels and the false alarms in the declared edge map.

Note that although the original formulation of the optimal assignment problem assumes the numbers of elements in K and L are equal, the Hungarian algorithm can be implemented to handle unequal cases as well [77]. Therefore, we do not need to make them the same size by adding ghost elements. Let N_K and N_L denote the number of elements in K and L , respectively.

The following is an outline of the edge detection result classification and association procedure.

1. Resolving the simple cases:

- The ground-truth edge pixels which have declared edge pixels right at the same locations are automatically determined as detected ground-truth edge pixels. They are associated with those declared edge pixels at the same locations, hence they have no localization error. The declared edge pixels that these ground-truth pixels are associated with are automatically determined as correct declarations.
- The ground-truth edge pixels which do not have any declared edge pixels within a distance of the chosen threshold τ are automatically determined as misdetected ground-truth edge pixels.
- Similarly, the declared edge pixels which do not have any ground-truth edge pixels within a distance of the chosen threshold τ are automatically determined as false alarms.

The rest of the procedure only deals with the ground-truth and declared edge pixels that are left undetermined.

2. Let K be the set of ground-truth edge pixel locations, L be the set of declared edge pixel locations. Create an $N_K \times N_L$ matrix Q of distance values between each pair in $K \times L$.
3. Selectively modify Q by applying the rule in (6.8), with N_K as N in (6.7).
4. Apply the Hungarian algorithm on the modified Q .
5. Examine the resulting assignment. (Implementation issue: remember to recompute the distance for the assigned pairs if the content of the distance matrix Q is modified by the particular implementation of the Hungarian algorithm.)
 - For each assigned pair
 - if the value in the distance map Q is not larger than τ , this is a valid association. The involved ground-truth and declared edge pixels are paired up and to be counted as a detected ground-truth pixel and a correct declaration;
 - otherwise, this is a ghost pair. If this pair is denoted by (k, l) , k is counted as a misdetected ground-truth edge pixel, and l is counted as a false alarm.
 - The left-over elements in K (if $N_K > N_L$) or L (if $N_K < N_L$) that are not paired up are all misdetected ground-truth edge pixels (if $N_K > N_L$) or false alarms (if $N_K < N_L$).

6.7 Experiment

We use the synthetic test image used in [62] to compare the proposed method with a distance transform based method. This test image is 64×64 , and has a brighter disk of constant gray value against a darker constant background. White Gaussian noise is added to obtain a noisy version of the image with $\text{SNR}=4$. Figure 6.2(a) shows the ground-truth edge map obtained by following the boundary (using 8-connectivity) of the disk on the noise-free image. There are a total of 132 ground-truth edge pixels and 3964 background pixels. Notice that this ground-truth is different from that used in [62], where a three-label (true-positive, don't-care, and false-positive) ground-truth is used. Here we want to give sharper performance measures and do not specify the “don't-care” region in the ground-truth. Our methodology, however, can easily accomodate the provision for the “don't-care” zone in situations where it is desired. This is done by re-examining the false alarms and omitting the ones falling in that zone.

We apply the implementation of Canny's edge detector used in [62] to the noisy image with different tuning parameters. The tuning parameters of (1.05,0.77,0.88) given in [62] produces the edge map shown in Figure 6.2(b). Another two sets of parameters are chosen to give more obvious misdetections and false alarms, respectively. These are also shown in Figure 6.2.

The proposed optimal matching based method and the distance transform based method are used to compare the declared edge maps with the ground-truth edge map. The distance transform based method used in [62] has a circular search radius of three pixels for edge (true-positive) pixels. Our implementation of this method used in the experiment reported here uses a square search region of 5×5 pixels with the origin at the center. The Euclidean distance squared is used to form the distance matrix of the assignment based method. The threshold τ in inequality (6.2) is set to 8, corresponding to a circular search radius of $2\sqrt{2}$. The performance measures of

Table 6.1: Performance measures for the declared edge maps by two performance evaluation methods.

edge map	optimal matching			distance transform		
	# MD	# FA	RMS loc error	# MD	# FA	RMS loc error
1	0	42	0.5	0	0	1.0
2	9	28	1.39	0	0	1.27
3	0	74	1.0	0	28	1.0

number of misdetections ($\#MD$), number of false alarms ($\#FA$), and the root-mean-squared (RMS) localization error for the detected ground-truth edge pixels from these two methods are given in Table 6.1.

It should be noted that the purpose of the experiment here is to demonstrate the appropriateness of using the optimal matching model for counting misdetections and false alarms and calculating localization error for correct detections. It is not our purpose here to carry out a full empirical evaluation of any edge detector. No effort is made to tune the detector to its best performance according to the performance measures.

When examining the data in Table 6.1, the emphasis should be on comparing the same performance measures given by the two different methods.

In visually comparing edge map 1 with the ground-truth, we see no misdetection. However, the thickness of the edge is not even, with some parts being 8-connected and some parts being densely 4-connected. According to our 8-connected thin ground-truth edge map, these thicker parts of the declared edge (where edge pixels have more than two 8-neighbors) contains false alarms. The proposed method precisely picks out these declared pixels as false alarms, which are shown in Figure 6.3(a). The distance transform based method leniently accepts all declared edge pixels as being appropriate. If, however, this level of detail is not of interest, a “don’t care” zone can

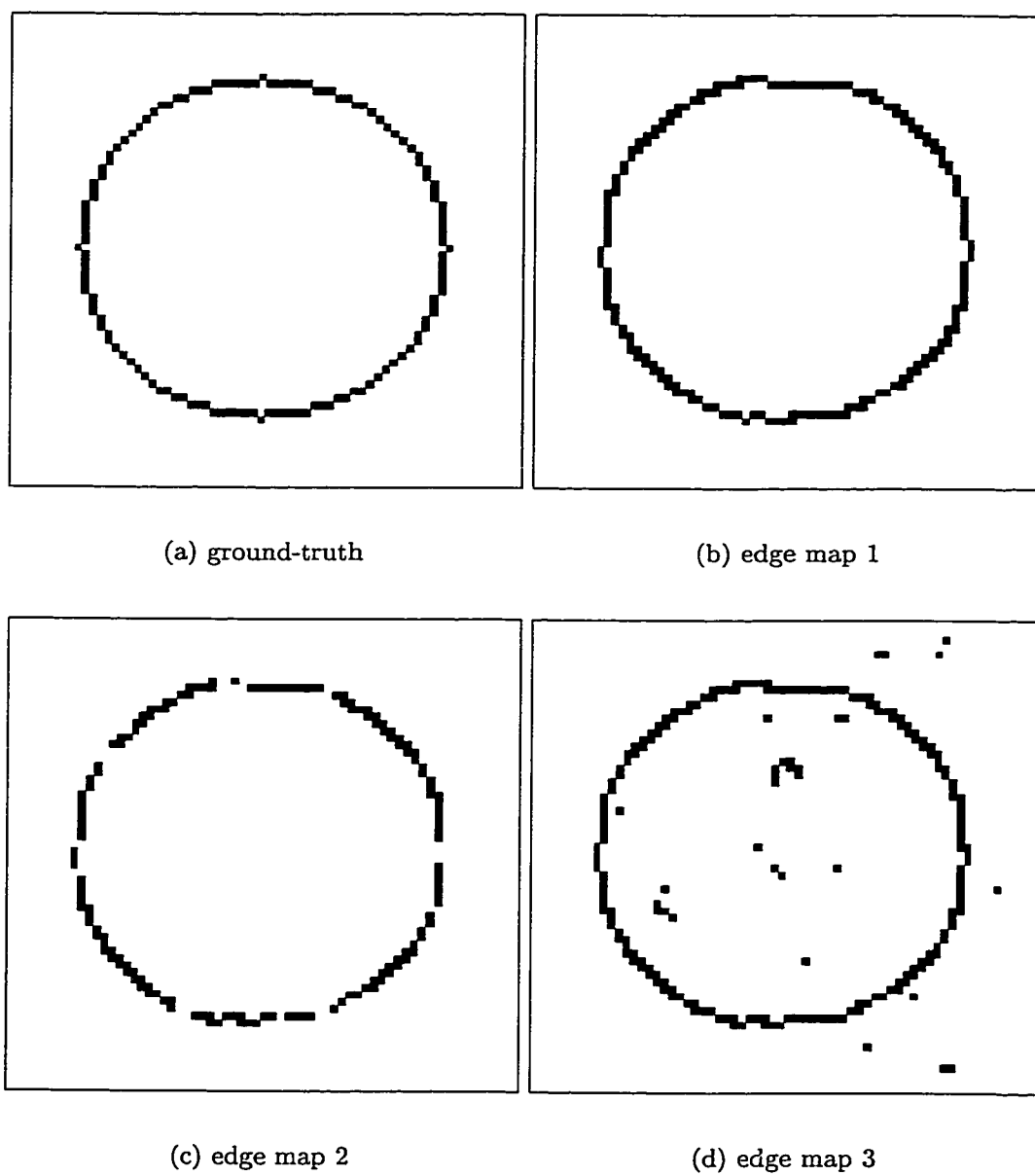


Figure 6.2: Edge maps on which the performance measures are calculated.

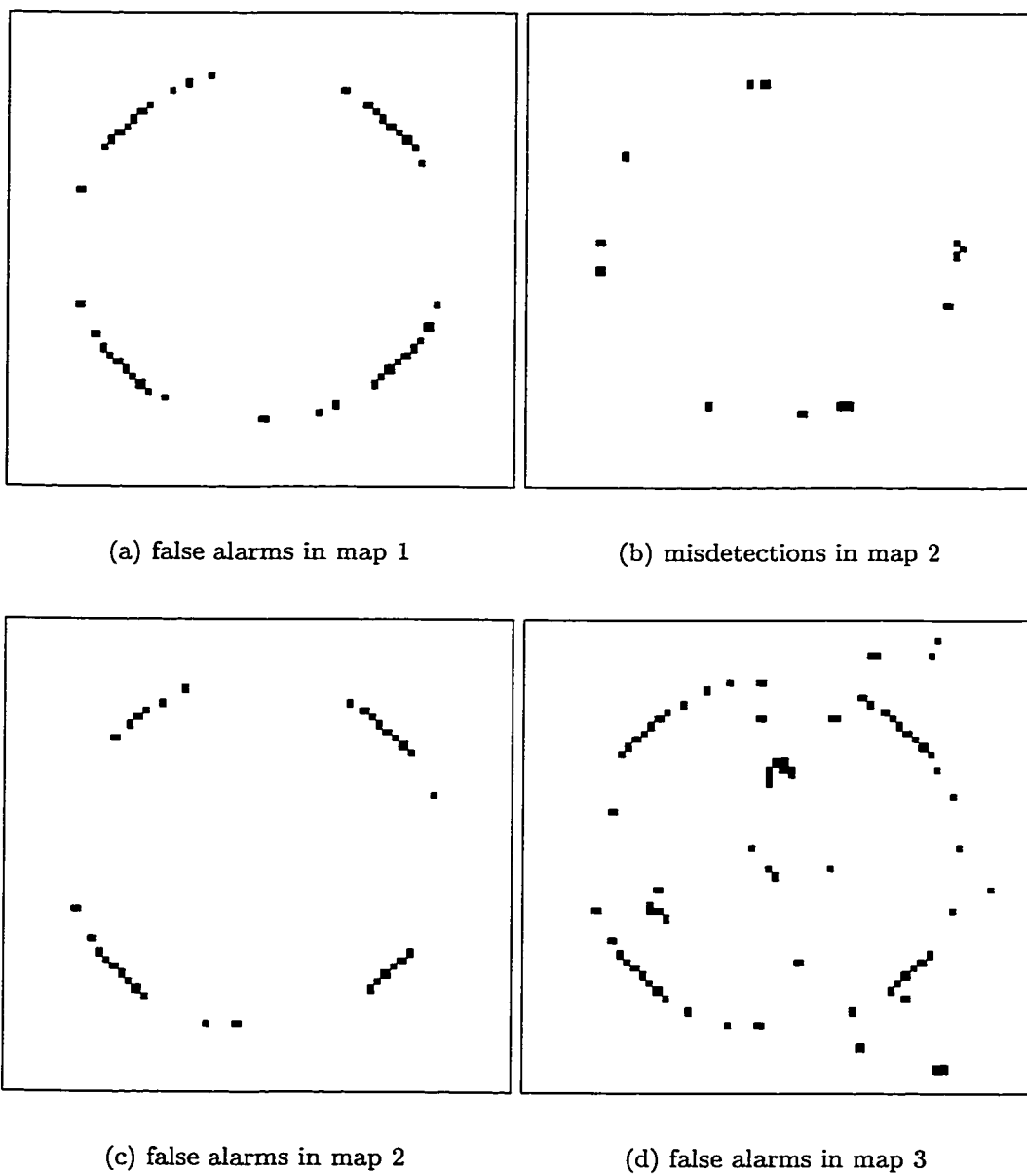


Figure 6.3: Misdetection and false alarm edge pixels in the edge maps.

be specified in the ground-truth. The false alarms declared by the assignment based method falling into that zone will then be tolerated and not counted as false alarms. If that were done, both methods would claim no false alarm.

The difference between the two methods is clearer on their evaluation results on edge maps 2 and 3. In edge map 2, we clearly see gaps in the declared edge map which we certainly want to call misdetections. At the same time, we also see some false alarms on the thicker parts of the declared edges. Both of these are reflected in the evaluation result by the optimal matching based method. The identified misdetection and false alarm pixels are shown in Figure 6.3(b) and 6.3(c), respectively. The distance transform based method is overly lenient again and declares no misdetection nor false alarm.

In edge map 3, the optimal matching based method treats not only the stray edge pixels as false alarms, but also some on the thicker parts of the edges around ground-truth edges. These pixels are shown in Figure 6.3(d). The distance transform based method treats only those stray edge pixels as false alarms, and tolerates all declared edge pixels around ground-truth edge pixels.

In general, we observe the tendency of the distance transform based method to be overly reluctant to declare false alarms around ground-truth edge locations, and overly reluctant to declare misdetections around any declared edge pixels. It gives inappropriately high performance measures. The optimal matching based method is more precise, and gives performance measures conforming to the intuition and the subjective evaluation.

6.8 ATR performance evaluation

The methodology just described is useful in many applications where a maximal one-to-one correspondence is to be established between two sets, where a distance-like dissimilarity/penalty measure can be made between the elements in the two sets,

and the pairing process is constrained by some threshold on the dissimilarity/penalty measure. We choose to base our discussion first on the edge detection application not only due to the importance of edge detection as a major feature extraction module in many computer vision algorithms, but also due to the relative simplicity of its performance evaluation so that the main idea of the proposed methodology can be described more clearly without confusion from other unrelated application-specific issues.

Now that the methodology has been described clearly, we use it to develop a procedure for ATR performance evaluation. In characterizing ATR algorithm performance, not only do we need to determine the detection rate and false alarm rate, we also need to compute the recognition rate for the correctly detected targets. In this situation, a two-stage assignment based procedure needs to be used. This procedure first identifies targets that are both correctly detected and correctly recognized. It then identifies targets that are correctly detected but incorrectly recognized. In the first stage, the distance matrix is constructed so that only targets of the same class ID can potentially form pairs. The targets paired up by this stage are those both correctly detected and correctly recognized. The left-overs go through the second stage. In this stage, the target class ID does not play any role in the construction of the distance matrix. The pairs formed in this stage involve only targets that are correctly detected but incorrectly recognized.

The number of correctly detected targets is the total number of pairs produced by both stages. Among these pairs, those produced by the first stage are the ones for the targets also correctly recognized. The left-overs from the second stage are the misdetected and false alarm targets. All the pairs and left-overs provide the precise information required in constructing the confusion matrix to show the recognition performance of the system. This is the procedure used in the experiments in Chapter 8 for evaluating ATR algorithm performance.

An example is provided here. Figure 6.4(a) shows one of the test images used

in this study. There are ten target instances in this image, with their silhouettes overlaid and an instance number given by which they are referenced. These targets are referred to as t_1, \dots, t_{10} . The class identification of these targets are given in the first “class” column in Table 6.2. The column and row positions of the centroids of these targets are given in the first “centroid” column.

One algorithm output is shown in Figure 6.4(b) with the silhouettes of the thirteen declared targets overlaid on the image. Also shown in the image are the instance numbers for these declared targets. These declared targets are referred to as d_1, \dots, d_{13} . The classes and locations of these declared targets are given in the second “class” and “centroid” columns in Table 6.2.

The squared Euclidean distance is used in constructing the distance matrix. In tolerating the location error, a threshold on the squared distance of $\tau = 625 \text{ pixel}^2$ is used, corresponding to a circular search region of radius 25 pixels.

In the first stage of the procedure, only targets of the same class (same class ID, not instance number) can be paired up. For any pair of ground-truth and declared targets whose class ID numbers are different, a distance value of 626, which is greater than the threshold $\tau = 625$, is assigned, regardless of the actual distance between the centroids of these targets. The actual distance is used only for pairs of targets whose class ID numbers are the same. The resulting distance matrix is shown in Table 6.3. Subjecting this matrix to the procedure in Section 6.6, we find eight pairs — eight out of the ten ground-truth targets are both correctly detected and correctly recognized. These are shown in Table 6.4.

The targets left unpaired are t_7, t_9 on the ground-truth side and $d_5, d_6, d_7, d_{10}, d_{11}$ on the declaration side. Any valid pairings between them which satisfies the distance constraint identify correctly detected but incorrectly recognized targets. The distance matrix for the second stage of the procedure contains only the actual distances, as shown in Table 6.5. Without actually running the Hungarian algorithm, we can see the solution that only (t_7, d_{11}) is a valid pair with their distance smaller than the

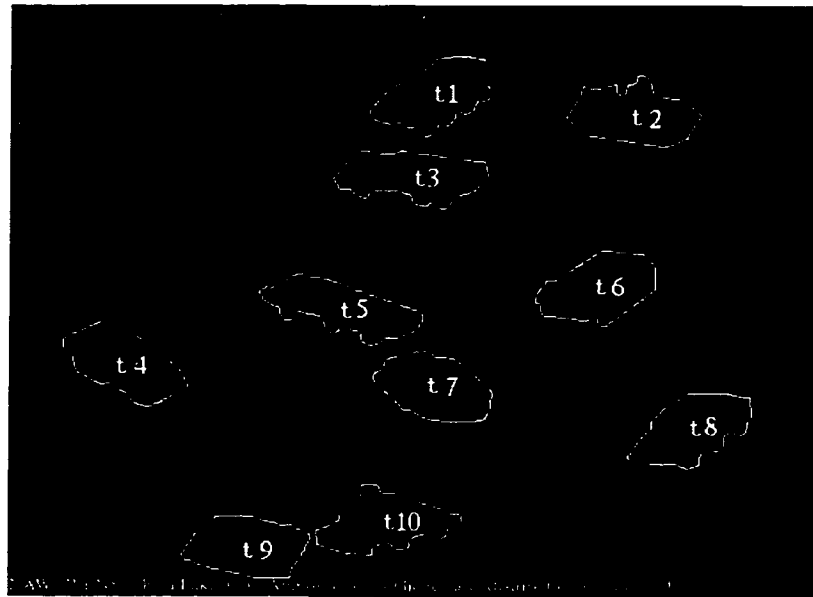
threshold.

Now the full characterization of the algorithm output in Figure 6.4(b) can be summarized as follows.

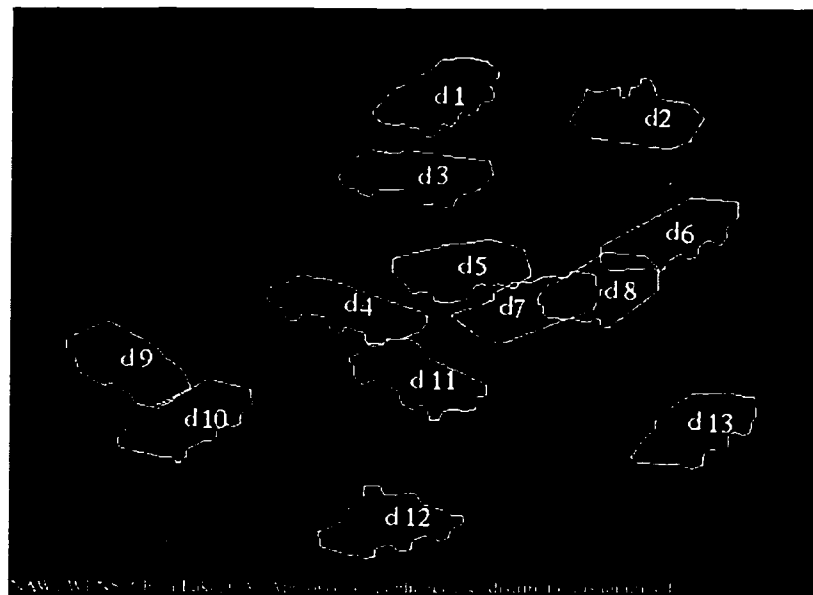
- Of the ten ground-truth target instances
 - nine are correctly detected, eight of which are also correctly recognized;
 - (redundant) one was misdetected; one was correctly detected but incorrectly recognized;
- Of the thirteen declared target instances
 - nine are correct declarations, eight of which are also have the correct class ID;
 - (redundant) four are false alarms which do not correspond to any ground-truth target; one correct declaration has a wrong target class ID declared.

6.9 Summary

We have presented a methodology for most reasonably associating the ground-truth entities with the entities declared by algorithms for a broad class of detection and recognition applications. The central idea is to identify the association problem as an optimal matching problem, which is transformed into an unconstrained optimal assignment problem which has an efficient solution by the Hungarian algorithm. The resulting association establishes a one-to-one correspondence between detected ground-truth entities and the correct declarations in the algorithm output. The determination of misdetected ground-truth pixels and false alarms in the declaration as well as the computation of the localization error for the detected ground-truth entities are made according to the established correspondence. We showed examples where the appropriateness of this technique is verified.



(a) ground-truth targets



(b) declared targets

Figure 6.4: Ground-truth and declared targets subject to the two-stage matching procedure to determine the detected, recognized and false alarm targets.

Table 6.2: Targets in the example on using the proposed method for ATR performance evaluation.

ground-truth			declared		
reference	class	centroid	reference	class	centroid
t_1	21	(268,60)	d_1	21	(269,59)
t_2	1	(391,74)	d_2	1	(392,74)
t_3	18	(253,114)	d_3	18	(255,113)
t_4	26	(72,243)	d_4	24	(211,205)
t_5	24	(209,205)	d_5	19	(284,177)
t_6	31	(370,191)	d_6	25	(414,154)
t_7	29	(268,259)	d_7	15	(327,203)
t_8	23	(429,287)	d_8	31	(371,191)
t_9	27	(149,367)	d_9	26	(73,243)
t_{10}	9	(236,351)	d_{10}	3	(111,281)
			d_{11}	4	(256,251)
			d_{12}	9	(237,351)
			d_{13}	23	(431,286)

Table 6.3: Distance matrix used in the first matching procedure for determining correctly detected and correctly recognized targets. Threshold to be used is 625.

	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8	d_9	d_{10}	d_{11}	d_{12}	d_{13}
t_1	2	626	626	626	626	626	626	626	626	626	626	626	626
t_2	626	1	626	626	626	626	626	626	626	626	626	626	626
t_3	626	626	5	626	626	626	626	626	626	626	626	626	626
t_4	626	626	626	626	626	626	626	626	1	626	626	626	626
t_5	626	626	626	4	626	626	626	626	626	626	626	626	626
t_6	626	626	626	626	626	626	626	1	626	626	626	626	626
t_7	626	626	626	626	626	626	626	626	626	626	626	626	626
t_8	626	626	626	626	626	626	626	626	626	626	626	626	5
t_9	626	626	626	626	626	626	626	626	626	626	626	626	626
t_{10}	626	626	626	626	626	626	626	626	626	626	626	1	626

Table 6.4: Targets that are both correctly detected and correctly recognized.

ground-truth	t_1	t_2	t_3	t_5	t_6	t_4	t_{10}	t_8
declared	d_1	d_2	d_3	d_4	d_8	d_9	d_{12}	d_{13}
ID	21	1	18	24	31	26	9	23

Table 6.5: Distance matrix used in the second matching procedure for determining correctly detected and but incorrectly recognized targets. Threshold to be used is 625.

	d_5	d_6	d_7	d_{10}	d_{11}
t_7	6980	32341	6617	25133	208
t_9	54325	115594	58580	8840	24905

Although the development of the technique is carried out in an edge detector evaluation scenario, this methodology is readily applicable to performance evaluation of other applications where the classification as well as the localization performance are of interest, such as vehicle detection in aerial images, automatic target recognition, and counting certain cells in the medical imaging application. A two-stage association procedure based on the proposed methodology has been outlined for the ATR application. The same procedure can be applied with marginal or no modifications for other detection and recognition applications, such as evaluating the performance of biomedical cell counting algorithms.

Chapter 7

SIMULATED FLIR SCENE GENERATION

7.1 Introduction

Real data in the FLIR ATR application is characterized by great variability in the target appearance due to the thermal activity and meteorological conditions. The natural background also adds greatly to the complexity of the scene. The most difficult challenge presented to ATR algorithms is to handle such great variability and complexity in the uncontrolled outdoor scenes.

To aid the development of algorithms, some effort has been made to provide real FLIR data to the research community, e.g., [10] and several other data sets accessible through the web-page of the Center for Imaging Science (<http://www.cis.jhu.edu>) at the Johns Hopkins University. However, due to the practical difficulties involved in gathering and supplying the data, the target types and conditions, imaging conditions and scene complexity are very limited. Only in rare cases is there precise ground-truth information about the targets in the scene. The situation is made worse by the restriction on the accessibility of the data due to the military classification of the data sets. For example, Li *et al* [84] report empirical performance evaluation results of a number of FLIR ATR algorithms on a single fairly large sized data set with more than 17,000 target image chips. It is highly desirable to run the algorithm developed here on that data set and compare the resulting performance with those reported in [84]. However, the effort made to gain access to that data set for use in this dissertation research was in vain.

Large data sets with ground-truth is essential in assessing the performance of

computer vision algorithms. Due to the difficulty in obtaining such data sets, synthetic or simulated images are often used in studies [162, 31, 117, 111, 27, 159, 82]. We also take this alternative and generate simulated FLIR scenes by planting random target appearances into real FLIR scenes. It has the advantage of having extremely low cost and readily available ground-truth information from the scene generation process. In what follows, we describe the procedure by which simulated FLIR scenes are generated.

7.2 *Scene background*

The FLIR scene generation procedure operates by choosing a background scene from a number of alternative real FLIR scenes and planting into that scene a random number of targets with varied appearances at randomly picked locations. The “land images” in the NAWC Chinalake Presentation data set (referred to as the *Chinalake data set*) contains seventeen real FLIR scenes. These images contains outdoor scenes. The sizes of the images are around 512×400 . The twelve images in this data set that are used in the scene generation are shown in Figure 7.1.

7.3 *Insertion of targets*

From a library of given targets, the procedure randomly picks ten to fifteen targets, simulates their intensity values, and selects locations in the background scene to place them so that their bounding boxes do not overlap.

The best commercial software for infrared target signature (appearance) simulation is the PRISM (Physically Reasonable Infrared Signature Model) running on the SGI workstations. The license for using this software is issued by the ThermoAnalytics, Inc. (<http://www.thermoanalytics.com>) under the authorization of the US Army TACOM (Tank-automotive & Armaments Command). This software, however, is not available for this dissertation research. In our experiments, the target signature

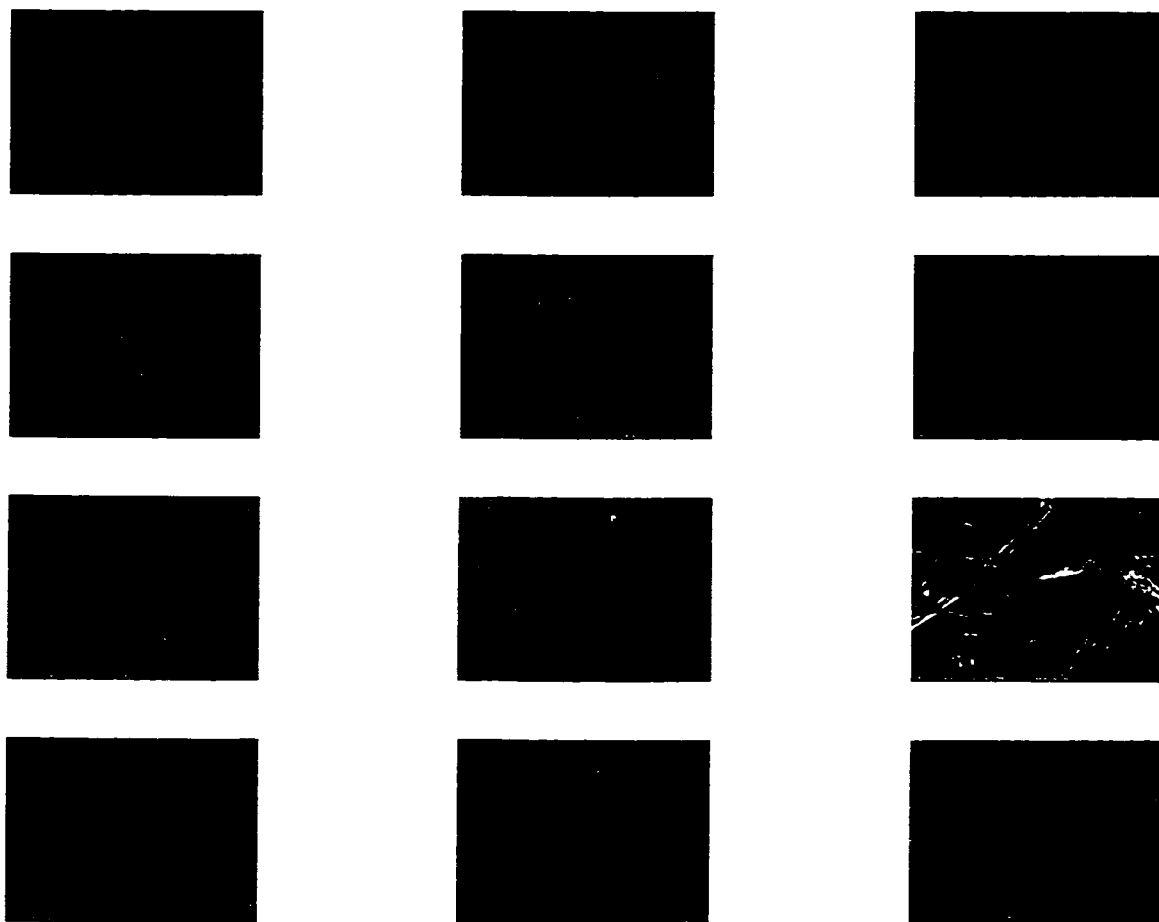


Figure 7.1: Images in the Chinalake data set used as background scenes in the FLIR scene generation.

is simulated through a procedure that uses real target image chips with corruption of some additive texture. The added texture is generated from the statistics of the real target image chip. We do not claim this procedure to be the best resource for target signature simulation. However, when better target signature simulation methods become available, this part can be easily replaced.

7.4 Target signature simulation

The MURI FLIR data set contains thirty-six images from second-generation FLIR sensors. Each of the images contains one instance of a ground vehicle target. The image chips for these targets are hand-segmented from the images, nine of which are shown in Figure 7.2. These nine are used for generating the simulated target signature to be placed into the background scenes. Some measurements on the 2-D geometric shape are computed to give some characterization for these targets. These are the length/height ratio, the ratio of the perimeter squared over the area, the ratio of the standard deviation over the mean of the radii (the distances from the boundary points to the centroid), and the ratio of the target centroid uncertainty over that of the circular disk of the same area and contrast. These values are given in Table 7.1.

Due to the high variability of the thermal history of the targets and the imaging conditions, the target signature can undergo very large changes from scene to scene. However, we still do not expect the signature of the same target in different scenes to be totally unrelated to each other. Therefore, we generate the simulated target signature with certain levels of deviation from the MURI FLIR target chips. In doing this, we introduce additive texture to the real target chips.

Let the real target image chip be denoted by x_0 . Let a texture image chip of the same size and shape be denoted by x_1 . Let λ be a constant between 0 and 1 to control the level of deviation. The simulated target signature is obtained as

$$y = \lambda x_0 + (1 - \lambda)x_1 \quad (7.1)$$

Table 7.1: Geometric shape measurements made for the MURI target chips. The measurements are the target length/height ratio, the ratio of the perimeter squared over the area, the ratio of the standard deviation over the mean of the radii (the distances from the boundary points to the centroid), and the ratio of the target centroid uncertainty over that of the circular disk of the same area and contrast.

target	len/height	perim ² /area	std(R)/mean(R)	cen ratio
1	1.6809	16.0468	0.2508	1.7222
2	1.7500	17.9195	0.2664	1.8184
3	2.6316	23.4842	0.3995	2.4612
4	1.4737	15.6646	0.2979	1.9377
5	1.5818	19.3976	0.2999	2.0731
6	1.5000	14.5795	0.2537	1.7955
7	1.0455	18.4090	0.1903	1.7068
8	1.1579	12.5409	0.1779	1.5071
9	2.0851	22.2816	0.3657	2.3412

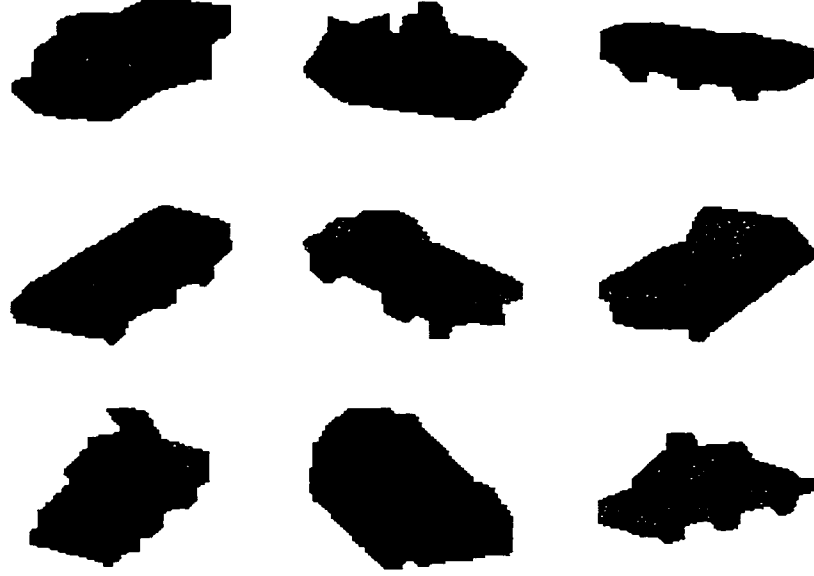


Figure 7.2: Some of the hand-segmented target image chips from the MURI FLIR data set.

The synthesis of x_1 uses the power spectrum method of texture synthesis, so that the spatial correlation of the pixel intensity in x_1 is approximately the same as that in x_0 . This is done as follows. The auto-correlation matrix of the pixel intensity value is estimated from the real target chip x_0 using a square neighborhood whose size is pre-determined. On the mean-subtracted version of x_0 , the correlation in the intensity values at two pixel locations with relative displacement of $(\Delta r, \Delta c)$ is estimated as

$$\sigma^2(\Delta r, \Delta c) = \frac{1}{\#A} \sum_{(r,c,\Delta r,\Delta c) \in A} I(r, c) I(r + \Delta r, c + \Delta c) \quad (7.2)$$

where $I(\cdot, \cdot)$ is the image intensity function for the mean-subtracted x_0 , $\#A$ is the number of elements in the set $A = \{(r, c, \Delta r, \Delta c) | (r, c) \in T, (r + \Delta r, c + \Delta c) \in T\}$, and T specifies the geometric shape of the target.

In practice, due to the asymmetric shape of the targets, the estimated correlation matrix is not symmetric. We arbitrarily make it symmetric by averaging the

corresponding elements on either sides of the center row, center column, and the two diagonals. The discrete Fourier transform (DFT) [112] is applied to the resulting symmetric correlation matrix. This gives us the power spectrum of the target pixel intensity random field. Any instance of negative elements in the power spectrum is reset to zero. Then the inverse DFT is taken on the square root of the power spectrum to give us a digital filter. This filter is used as the target texture generator. When this filter is applied to a white noise image, the auto-correlation of the output image is roughly the same as that of the original real target image chip x_0 . Cutting a piece of the output image gives us x_1 in Equation (7.1).

Figure 7.3 shows some examples of target signatures synthesized using the above procedure with different neighborhood sizes for correlation estimation. Figure 7.4 shows some examples of simulated target signature by combining the real target chip with the texture synthesized using the 11×11 auto-correlation matrix. The left column shows real target chips without any additive texture ($\lambda = 1$); the remaining three columns show the cases with increasing amounts of additive target texture, with $\lambda = 0.7, 0.5, 0.3$.

7.5 Target contrast and dynamic range

We just described a way for simulating target signature which allows some reasonable and controlled deviation from the real target signature. The process of placing the simulated target signature into the background scenes requires some proper scaling and shifting of the simulated target chip intensity values. This is necessary for the resulting simulated scene to be more similar to real scenes.

We first gather some statistics on the contrast and dynamic range of real targets from the MURI FLIR data set. For each target instance and its immediate background area in the images, we first compute the mean and standard deviation of pixel intensity values. These are denoted by μ_t and σ_t for the target, and μ_b and σ_b for the

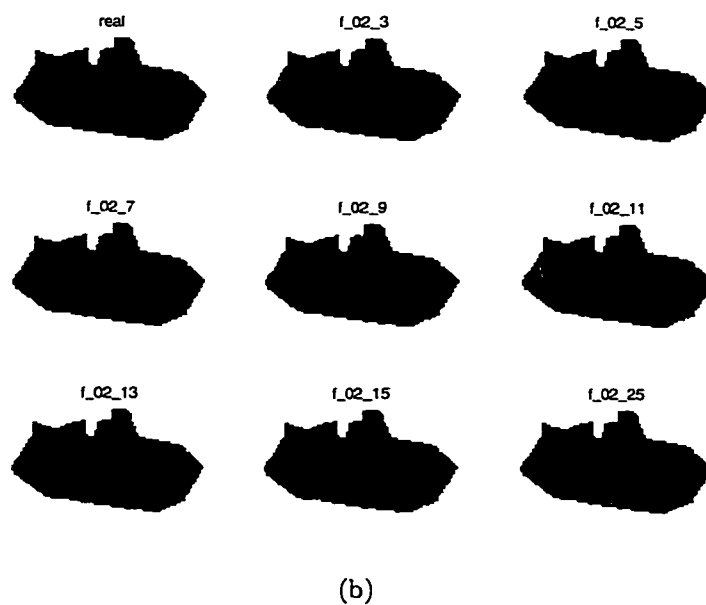
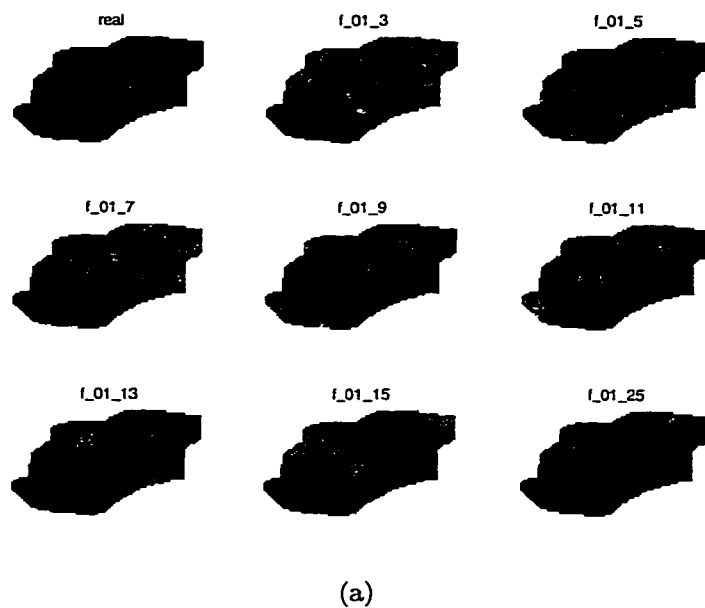


Figure 7.3: Synthesized texture for target signature simulation. Shown here are textures synthesized from two real target image chips. For both (a) and (b), the top-left corner shows the real target chip. The remaining eight images, from top to bottom and left to right, show texture patterns synthesized from correlation windows of sizes 3×3 , 5×5 , 7×7 , 9×9 , 11×11 , 13×13 , 15×15 , and 25×25 .

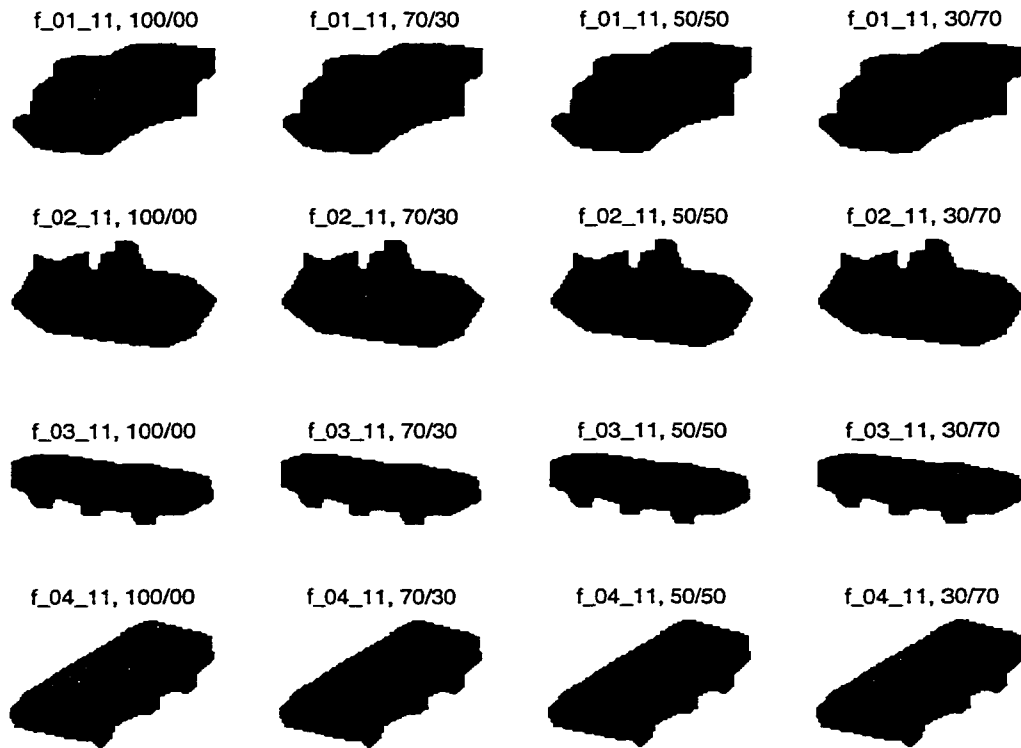


Figure 7.4: Simulated target signature with corruption of additive texture. Left column show original real target chip without additive texture ($\lambda = 1.0$). The remaining three columns, from left to right, show the cases with increasing amount of additive texture, with $1 - \lambda = 0.3, 0.5, 0.7$.

Table 7.2: Relative contrast and dynamic range of the targets in the MURI FLIR data set.

	min	max	median	mean	stddev
c	-0.66	5.62	1.76	1.90	1.44
$ c $	0.02	5.62	1.76	1.95	1.37
d	0.83	4.38	2.02	2.10	0.70

background. The relative contrast of the target is defined as

$$c = \frac{\mu_t - \mu_b}{\sigma_b} \quad (7.3)$$

The relative dynamic range is defined as

$$d = \frac{\sigma_t}{\sigma_b} \quad (7.4)$$

These are taken as constants which characterize the contrast and dynamic range of the target. On the MURI FLIR data set, values for c and d for the targets are summarized in Table 7.2.

When placing a simulated target chip into the background scene, the mean and standard deviation of the immediate local background are computed and denoted as μ'_b and σ'_b . The intensity values of the simulated target chip are then shifted and rescaled so that its mean and standard deviation are

$$\mu'_t = c\sigma'_b + \mu'_b \quad (7.5)$$

and

$$\sigma'_t = d\sigma'_b \quad (7.6)$$

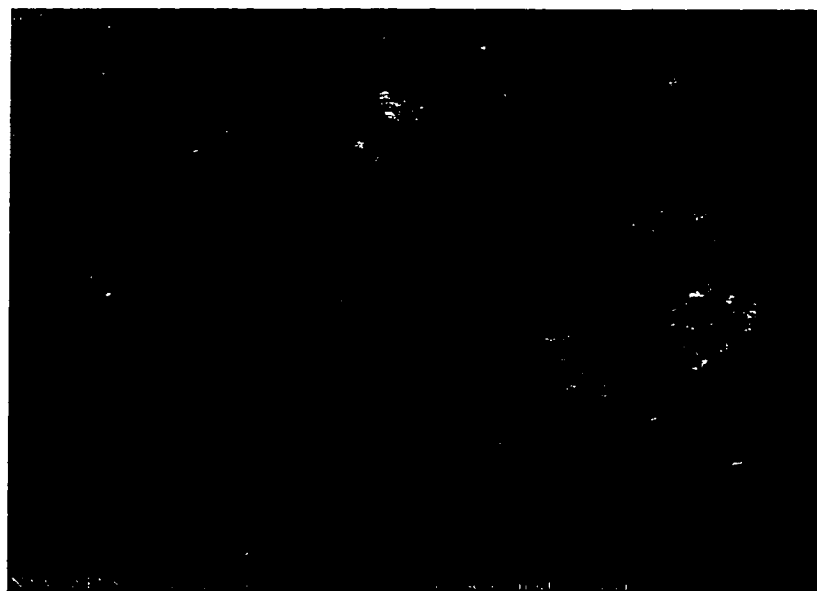
respectively.

After the rescaling and shifting of the intensity values of the simulated target chip are done, the updated target chip is placed into the background scene by replacing

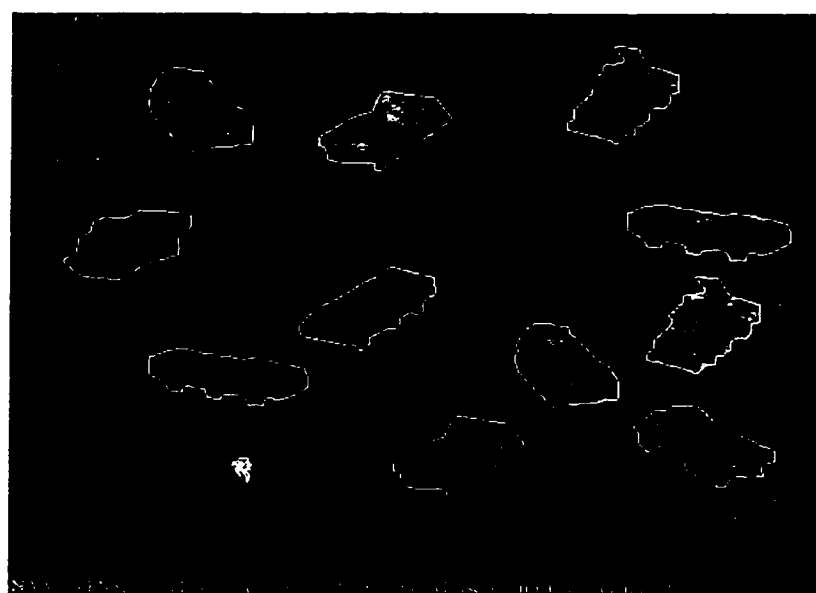
the intensity values at the pixel locations it covers. To smooth the transition of pixel intensity values from the target to the background and vice versa, local smoothing is applied to the intensity values of the target pixels adjacent to the background, and the background pixels adjacent to the target.

In our experiment, an 11×11 neighborhood is used for texture synthesis, λ values of 0.7, 0.5, 0.3 are used in separate experiments to observe algorithm performance on low, medium and high level of target signature deviation. Low, medium and high levels of target relative contrast and dynamic range are also used in the experiment. These levels are set by the values (0.4, 1.3), (1.8, 2.0), (3.2, 2.7) for (c, d) , which are obtained by subtracting the standard deviations from the medians, the medians, and adding the standard deviations to the medians, respectively. As a result, there are nine different combinations target signature deviation and relative contrast and dynamic range. For each combination, a set of 100 test images are generated by the procedure described above. The 100 images in each set contains around 1250 target instances. The performance of all competing algorithms are observed on each test image set.

Figure 7.5 shows an example of the simulated scene from using the medium target signature deviation ($\lambda = 0.5$) and medium relative contrast and dynamic range ($c = 1.8, d = 2.0$). In this scene, 11 targets are planted into the 12th image in the Chinalake data set (the 8th image in Figure 7.1). Because of the low contrast in the scene, some targets are quite hard to see even to the human eyes. To make it easier for the reader to study the image, we also provide an image with the target silhouettes outlined.



(a) Simulated scene.



(b) With silhouettes overlaid.

Figure 7.5: Example of simulated FLIR scene.

Chapter 8

FLIR ATR EXPERIMENTS

This chapter describes the experiment conducted to evaluate the performance of the centroid uncertainty based FLIR ATR algorithm, and to investigate the combination of the centroid uncertainty with other target detection and recognition techniques.

We start by giving some implementation details of the algorithm that are not covered by previous chapters, which mainly deal with the theory behind the centroid uncertainty based target saliency measure. Specifically, in addition to an outline of the algorithm sequence, we briefly discuss the normalization of the centroid uncertainty measure, a generic thresholding procedure and ROC operating point selection.

A few competing algorithms are then briefly described. The performance of these algorithms is compared with the centroid uncertainty based algorithm on the same set of test data. In particular, we describe the use of a linear classifier framework for combining the centroid uncertainty based algorithm with the traditional matched filter algorithm to further improve FLIR ATR performance.

For the test data, it is unfortunate for us not to be able to gain access to some large data set of real images suitable for this study due to the military classification on the data. The performance of the algorithms is evaluated on simulated data sets created from real FLIR scenes. The target image chips inserted into those scenes are obtained from real target image chips and texture patterns generated from statistics gathered from real target image chips. The generation of these simulated FLIR scenes has already been described in Chapter 7. The observed performance of the tested algorithms on data sets generated with various levels of target detection and

recognition difficulty is reported in Section 8.3.

To compensate for the weakness of using simulated data in the experiment, we also apply the centroid uncertainty based algorithm on the real imagery for the aerial image vehicle detection application. Since the algorithm is a general methodology for target detection applications, it also works well in that application domain. The details of the experiment is described in Section 8.5.

8.1 Centroid uncertainty based FLIR ATR algorithm

Remember that our targeted application is where the only reliable target signature in the FLIR scene is the existence of some contrast across the target boundary. The motivation of an algorithm making use of this fact is to compute target significance value by a proper weighted average of the contrast along the target boundary.

The algorithm we are proposing follows this line of reasoning, but takes a detour. Instead of computing a direct weighted average of the contrast estimated for the target boundary locations, we convert the contrast estimates along the boundary locations into the edgel location covariance matrices for these locations. The geometric instability of the target, as measured by the trace of the location covariance matrix of the centroid, is computed via the polygon vertex-to-centroid covariance propagation. Therefore, when the contrast at the boundary locations becomes higher, the traces of covariance matrices of the boundary point locations become smaller and so does the trace of the centroid covariance matrix. Stronger evidence for the existence of the target of interest at the location being studied is indicated by a smaller trace of the centroid covariance matrix.

The significance of this algorithm over *ad hoc* procedures for combining the boundary contrast into a single target presence measure lies in the fact that there are sound mathematical relationships between the contrast estimates made on the boundary locations and the final target presence measure which is the propagated centroid lo-

cation uncertainty. The final target presence measure (or, more precisely, absence measure, since smaller values of this measure show stronger evidence of target presence) can be thought of conceptually as being inversely related to some weighted average of the boundary contrast. The conceptual weights are determined by the entire shape of the target boundary via a rigorous covariance propagation procedure, hence is much preferred over *ad hoc* procedures, such as computing simple arithmetic or geometric means for the boundary point contrast.

8.1.1 Normalized centroid trace as target presence measure

Without repeating the details in contrast estimation (Chapter 4), edgel location covariance estimation (Chapter 5), and polygon vertex-to-centroid covariance propagation (Chapter 3), the proposed FLIR ATR algorithm works as follows. The IDDGO (we used 5×5 in our experiments) is first applied to the input FLIR image. The gradient orientation, the edgel contrast along that orientation, and the variance of the noise variance are computed for all pixels. Then for each target silhouette of interest, we compute its presence measure at each pixel location in the image where the centroid of target silhouette can be placed.

In computing the presence measure for a target at a single location, we first compute the contrast across the target boundary for each point on the target silhouette. This is done by projecting the edgel contrast estimate onto the orientation perpendicular to the local segment of the digitized target silhouette. (In deciding the orientation of the local segment of the target silhouette, we fit a straight line segment to the 5-point target boundary segment centered at the pixel in question.) The contrast across the target boundary is used to index into a pre-computed table (see Section 5.3) of edgel location covariance matrix for the 0° step edge. These covariance matrices are first rotated according to the orientation perpendicular to the local segment of the target silhouette, and then substituted into the polygon vertex-to-centroid covariance propagation formula to find the centroid covariance matrix.

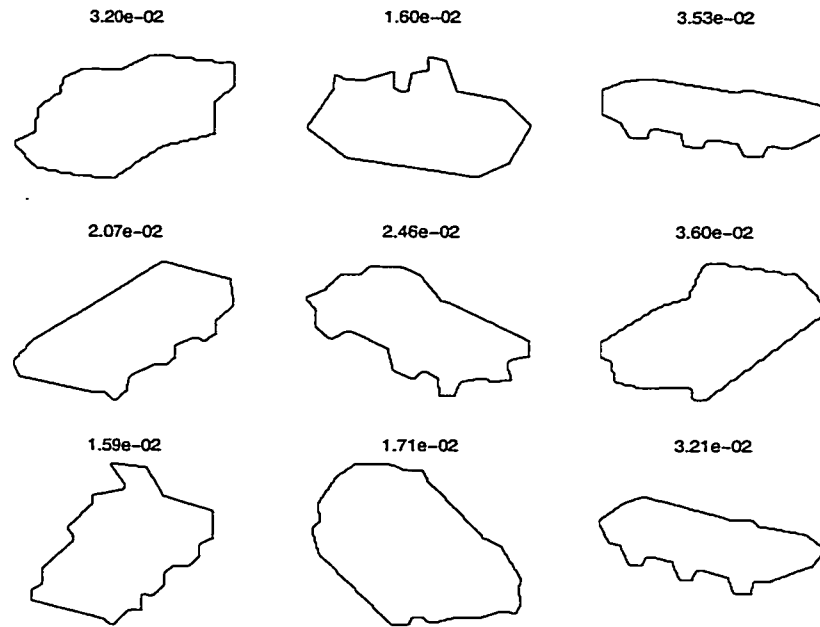


Figure 8.1: The centroid uncertainty needs to be normalized to reduce improperly biased preference toward certain target shapes over others.

The trace, i.e., sum of the two diagonal elements, of this final matrix is a measure of the target location uncertainty. Due to the difference in the shapes, the value for the trace can be very different for different targets, even if the contrast along their boundaries is exactly the same. This is illustrated by the targets and their trace values shown in Figure 8.1. In order to reduce the improperly biased preference toward certain target shapes over others, the trace value is normalized. The normalizing constant for a target is the trace of the centroid covariance matrix computed for the same target shape with zero-contrast across the boundary for all boundary points. It is the centroid uncertainty expected for the target silhouette placed in a pure noise background. The normalization is necessary for bringing the centroid uncertainty for different targets into the same range on the scale. This normalized trace value is the final measure for the target and location in question.

8.1.2 *Thresholding procedure*

A target ID-location pair is used to represent a target candidate if the normalized trace value computed for the said target and location is smaller than the specified threshold. A list of target candidates are obtained. This list is further examined and some of the candidates are dropped from the list when candidates overlap each other too much. In doing this, we first sort the list of candidates in the ascending order of the normalized centroid trace. The first in the sorted list is declared to be the first detected target and taken away from the list. This target occupies the rectangular area in the image that is the target's bounding box. Any target candidate whose centroid falls in this area is dropped from the candidate list. After this is done, the first in the remaining list is declared as the second detected target. Then the procedure repeats itself until the list becomes empty.

The effect of this thresholding procedure is that, for a target with strong appearance, there will be only one target declared instead of multiple declarations of the same target at several locations around the true target location. Also, the correct target declarations often has a quite small centroid uncertainty value. It can often suppress the declaration of some false alarms due to the accidental contribution of parts of the target to the evidence of existence of some non-existent targets.

8.1.3 *Operating point selection*

Virtually all ATR algorithms, including the ones involved in this study, have tuning parameters. Varying these parameters will usually give varied trade-offs between detection and false alarm rates. In characterizing the performance of ATR algorithms, it is customary to use the ROC curve which gives the false alarm versus detection rates over a range of interest. However, in practical applications, an operating point on the curve needs to be chosen which reflects a particular compromise between detection and false alarm rates.

In this experiment, we use an arbitrarily chosen linear cost function for operating point selection.

$$C = N_{MD} + kN_{FA} \quad (8.1)$$

where k is a specified constant to balance the cost of a misdetection and that of a false alarm. N_{MD} and N_{FA} are the numbers of misdetected targets and false alarm target declarations, respectively. It is a common practice in the ATR application to assign the cost of multiple number of false alarms to be the same of the cost of one misdetection. One reason is that a misdetected enemy target can potentially cause very serious damage. Another reason is that ATR systems often consist of cascaded subsystems. Subsequent processing by using other/fused information can possibly reduce the number of false alarms, but misdetected targets in early stages are very difficult to recover. Following this practice, in the experiments conducted here, k is arbitrarily set to be 0.5, which equates the cost of two false alarms to that of one misdetection.

8.2 *Competing algorithms*

This section gives a brief description of some FLIR ATR algorithms whose performance will be observed in the experiments and compared to that of the centroid covariance based algorithm.

Different target presence evidence measures are computed by these algorithms. However, similar post-processing procedures for thresholding the target presence evidence measures that is described in Section 8.1.2 is used to process the list of target candidates to produce the final target declarations. Obviously, the thresholds used for different algorithms are different.

8.2.1 *The Maryland algorithm*

Der and Chellappa [31, 32] of the Center for Automation Research at the University of Maryland proposed a model-based FLIR ATR algorithm for exactly the same application scenarios considered in this dissertation. Targets are modeled by their 2D silhouettes. The image local contrast is measured by “probes” which are the differences in the intensity values of pairs of pixels at certain orientations and variable distances. Sample distributions of the probes for the background hypothesis are estimated locally in the image. The uniform distribution is assumed for the foreground hypothesis. A generalized likelihood ratio test is set up using those distribution models.

To increase the robustness of the algorithm, the probe values need to be thresholded at proper levels and the target evidence is reduced to the number of probes along the boundary that exceed the threshold. Instead of an arbitrarily chosen threshold, an adaptive procedure is used to identify the threshold yielding the smallest possible false alarm rate, which is computed using a binomial model. At each possible target location, this smallest achievable false alarm rate is computed for all targets and the target with the smallest false alarm rate wins the competition. Further, if its rate is below the threshold for the tolerable false alarm rate, it is declared to be a target candidate.

8.2.2 *The likelihood ratio algorithm*

A variant of the Maryland algorithm was proposed [87] where no threshold on the probe value is applied. The probe values are used directly in a generalized likelihood ratio test for target detection and recognition. On the URI data set, this algorithm performed slightly better than the Maryland algorithm [87].

8.2.3 *A variant of the centroid covariance algorithm*

In the early stage of this dissertation research, a development that is based on the covariance propagation in 1-D template matching is used to estimate the location uncertainty of the target boundary points [90, 88, 89]. The cubic spline model is used for estimating the 1-D signal derivative in the covariance propagation computations. The resulting location uncertainty of the boundary points follows a 1-D perturbation model. The centroid covariance matrix is computed using the same formula as appeared in Chapter 3.

8.2.4 *Matched filter for FLIR ATR*

The matched filter and its variations such as the synthetic discriminant function (SDF) filters and the minimum average correlation energy (MACE) filters are popular classical methods for signal detection applications. Here we describe a matched filter based algorithm for detecting targets in the FLIR scenes. The performance of this algorithm is to be regarded as being representative of classical detection algorithms. Also, since the information used by this algorithm complements that used by the centroid uncertainty based algorithm, these two algorithms are combined in a linear classifier framework to further improve FLIR ATR performance.

Target and background instances

The theory behind matched filters is the likelihood ratio test with the assumption that the data vectors for both target and non-target classes have the multivariate normal distribution with the same covariance and different mean vectors. In the simplest form, the covariance matrix is assumed to be the identity matrix and the difference between the mean vectors is used to form the filter.

In the general case, we have N target classes and one general non-target class. A target instance is represented by a vector of the gray-scale values in a rectangular

area with the centroid of the target located at the center of that area. Recall that in this application, we assume that the shape, scale and orientation of the target are known. Let μ_1, \dots, μ_N and μ_0 denote the mean vectors of the N target classes and background class, respectively. Let the common covariance matrix be denoted by Σ . Let d denote the number of pixels in the rectangular area containing the target. The mean vectors are $d \times 1$ vectors. Σ is a $d \times d$ symmetric, positive-definite matrix.

The μ 's and Σ are estimated from training data. Following the same procedure as used in generating the FLIR scenes, we obtain instances of target appearances. The instances for the background class are simply obtained by cropping image chips from the background image at randomly selected locations. The cropped chips are of the same size as the rectangular area for the targets. We obtain M instances for each target class and NM instances for the background class. In other words, we obtain data vectors $x_{1,1}, \dots, x_{1,M}, x_{2,1}, \dots, x_{2,M}, \dots, x_{N,1}, \dots, x_{N,M}, x_{0,1}, \dots, x_{0,NM}$.

Because we are not interested in the constant bias of the gray-scale values, the mean value of each individual vector is subtracted from all its elements. The mean-subtracted vector is subsequently normalized to have unit energy.

The population parameters are estimated as

$$\begin{aligned}\mu_0 &= \frac{1}{NM} \sum_{m=1}^{NM} x_{0,m} \\ \mu_n &= \frac{1}{M} \sum_{m=1}^M x_{n,m} \quad (n = 1, \dots, N) \\ \bar{\mu} &= \frac{1}{2NM} \left(\sum_{n=1}^N \sum_{m=1}^M x_{n,m} + \sum_{m=1}^{NM} x_{0,m} \right) \\ \Sigma &= \frac{1}{2NM - 1} \left(\sum_{n=1}^N \sum_{m=1}^M (x_{n,m} - \bar{\mu})(x_{n,m} - \bar{\mu})' + \sum_{m=1}^{NM} (x_{0,m} - \bar{\mu})(x_{0,m} - \bar{\mu})' \right)\end{aligned}$$

where the prime stands for matrix transposition.

We find a matched filter for each of the target classes which best discriminates the instances of that class from the background class.

$$w_n = \Sigma^{-1}(\mu_n - \mu_0) \quad (8.2)$$

In practice, d is large (around 10000 in our experiments) due to the relative large sizes of the target image chips. Thus, there is a practical difficulty in computing Σ^{-1} . We simplify the issue by ignoring the off-diagonal elements of Σ , essentially setting them to 0.

In making the decision of whether a particular data vector x represents any target class, we compute the discriminant values for all target classes and identify the class giving the largest value.

$$\hat{n} = \arg \max_{n=1}^N |w'_n x| \quad (8.3)$$

This largest discriminant value, $|w'_n x|$, is compared to a chosen threshold value τ to decide whether to claim an instance of the target class \hat{n} or to claim the background class. The threshold value τ is varied in the experiment to give the ROC curve.

Since we follow the classical assumptions of the matched filter theory, i.e., data vectors of all classes follow the multivariate normal distribution with the same covariance matrix and different mean vectors, this test procedure corresponds to a generalized likelihood ratio test.

In our experiments, we have $N = 9$ targets and $M = 50$ instances for each of them. These instances are synthetically generated using the FLIR image generation procedure described in Chapter 7. Figure 8.2 shows the image chips for target #1 and background mean vectors, the overall standard deviation and the resulting matched filter for the first target.

8.2.5 Combining centroid covariance and matched filter for FLIR ATR

The centroid covariance based algorithm picks up only information for target presence along the boundary of the targets. The matched filter uses mainly the information extracted from the area inside and outside of the target boundary. The information being used by these two methods is rather independent of each other. Although the motivating consideration of the application discourages assumptions to be made on

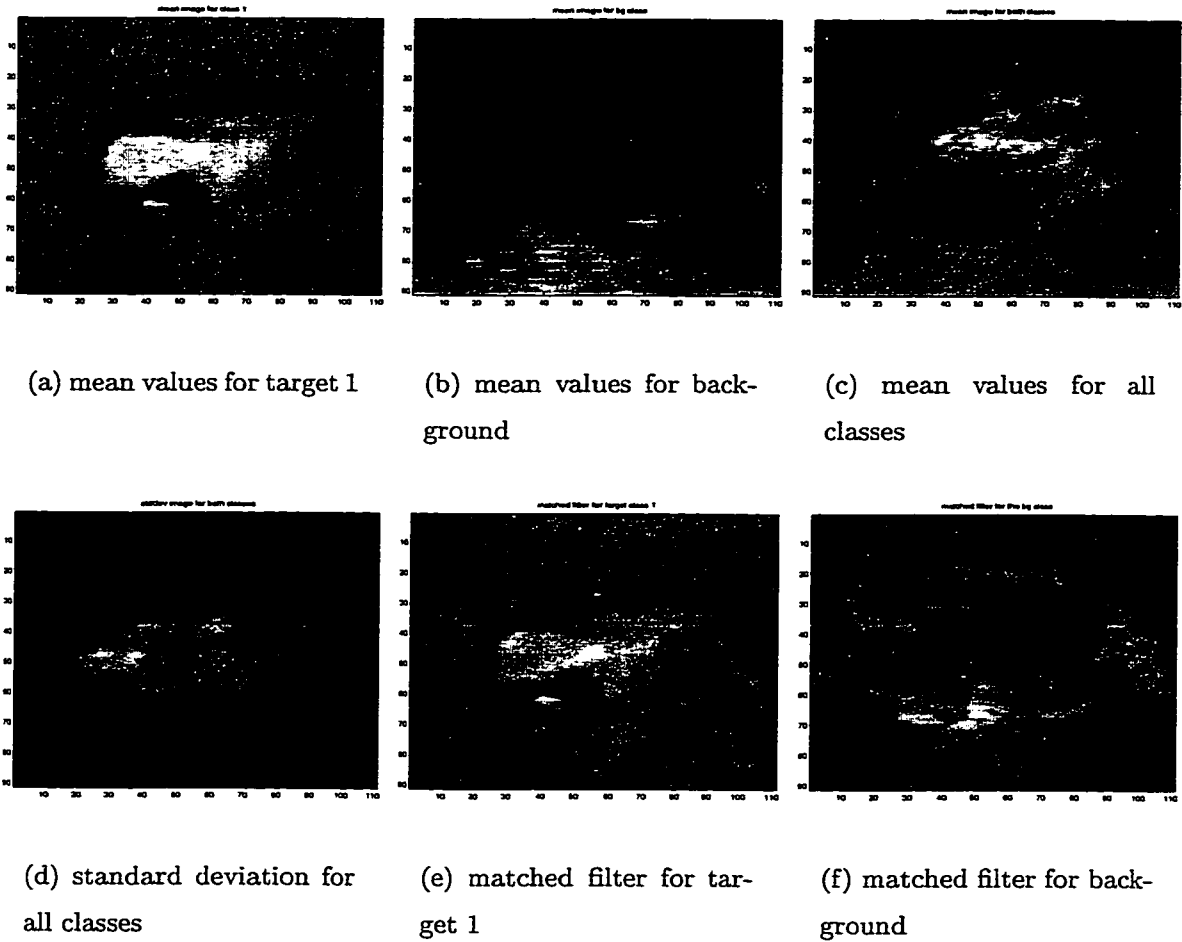


Figure 8.2: Example for the target and background class means, overall mean and standard deviation in obtaining the matched filters for the targets and background.

the inside area of the targets, our experiment results show that matched filters can still pick up some independent information contributing to more reliable detection of targets. Therefore, it is our intention in this section to explore possible ATR performance improvement due to the use of the information from both the centroid covariance based and matched filter based algorithms.

From the characterization of the target saliency measures of the two algorithms, we know that a strong target instance tends to have a small value for the normalized centroid trace and a large value for the matched filter output. These two measurements are combined in a linear classifier framework. Let f_t denote the matched filter output value and f_c denote the normalized centroid trace. These form the two-dimensional input feature vector for a linear classifier, which declares a target if

$$v_c f_c + v_t f_t \geq \tau \quad (8.4)$$

where v_c and v_t are classifier constants with $v_c^2 + v_t^2 = 1$. The value for the threshold τ determines the compromise between the numbers of misdetections (MD) and false alarms (FA) that are associated with the classifier output. The cost associated with these errors is computed as

$$(\#MD) + 0.5(\#FA) \quad (8.5)$$

Refer to Section 8.1.3 for some discussions on this cost function.

Our purpose is to find proper values for the triple (v_c, v_t, τ) which yields the minimum cost. A popular linear classifier is the Fisher linear classifier, where v_c and v_t are determined from the mean vectors and covariance matrix of the training samples. In many applications, this classifier performs very well. We also compute its performance in our experiment.

Following the FLIR scene generation procedure, we obtain 450 target image chips (50 for each of the nine targets in Figure 7.2) and 450 background image chips. The centroid and matched filter features are computed for the center location of the image

Table 8.1: Linear classifier performance in combining centroid uncertainty and matched filter for FLIR ATR.

	v_c	v_t	τ	#MD	#FA	cost
MF-only	0	1	376	19	29	33.5
centroid-only	-1	0	-0.57	13	7	16.5
Fisher	-1	1.62×10^{-4}	-0.53	9	6	12
best linear	-1	3.49×10^{-4}	-0.50	6	10	11

chips. These feature values are shown in the scatter-gram in Figure 8.3. We study the performance of linear classifier in four cases:

1. $v_t = 0$, corresponding to using only the matched filter; and
2. $v_c = 0$, corresponding to using only centroid uncertainty; and
3. v_t and v_c given by the Fisher linear classifier; and
4. $v_t^2 + v_c^2 = 1$, but otherwise unconstrained.

In all four cases, the threshold τ is varied to give the ROC. The best operating point on the ROC is identified which gives the smallest cost by Equation (8.5). The ROC curves are shown in Figure 8.4. The best performing classifier parameters and the associated errors and cost are given in Table 8.1. The parameters of the best linear classifier, except for the threshold τ which is varied to give the ROC curve, are used for combining the centroid and matched filter algorithms in all experiments of this chapter.

From these performance data, we confirm that the centroid uncertainty is a better feature for FLIR ATR than the matched filter. These two features do complement each other and can be combined to improve performance. The Fisher linear classifier

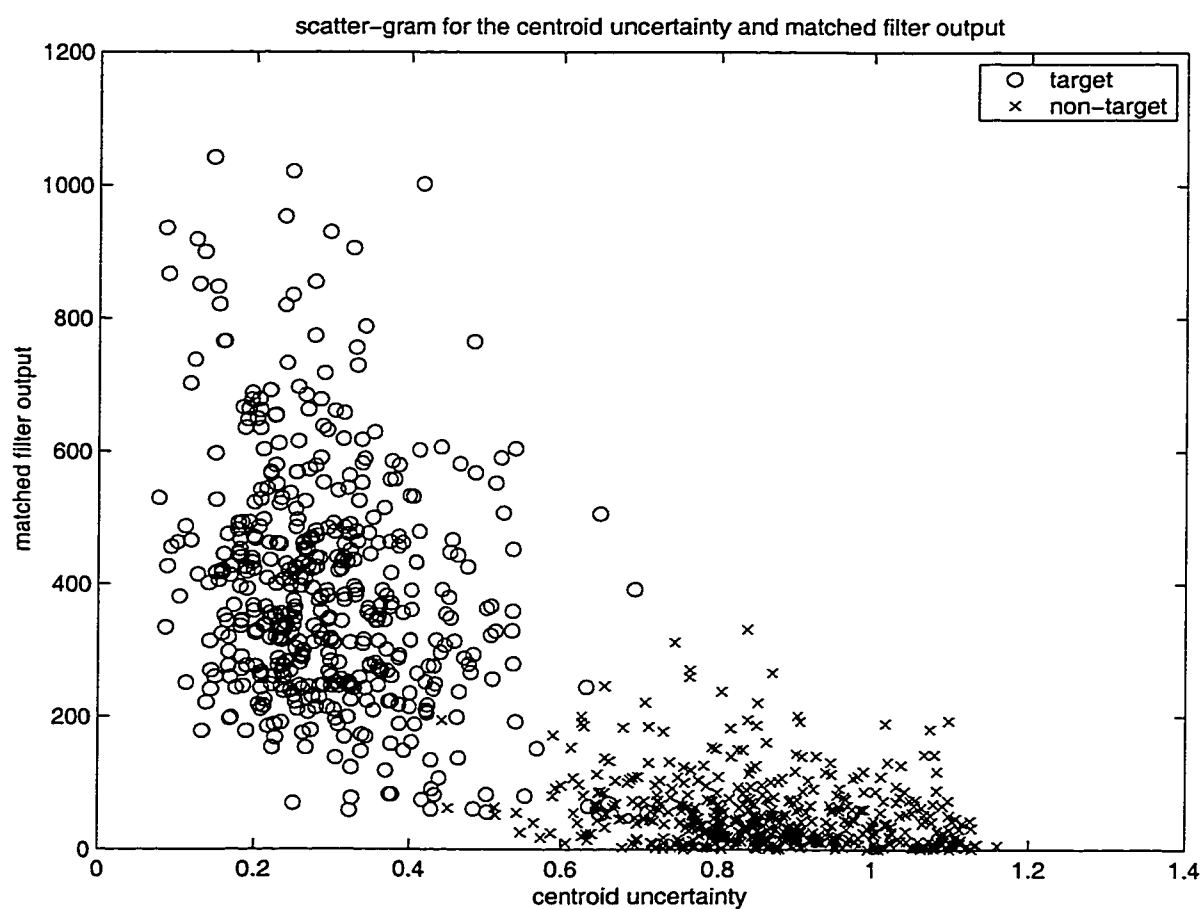


Figure 8.3: Feature vector distribution in combining the centroid uncertainty and matched filter for FLIR ATR. Horizontal axis for normalized centroid uncertainty; vertical axis for matched filter output; circles mark the target-class vectors; crosses mark the background-class vectors.

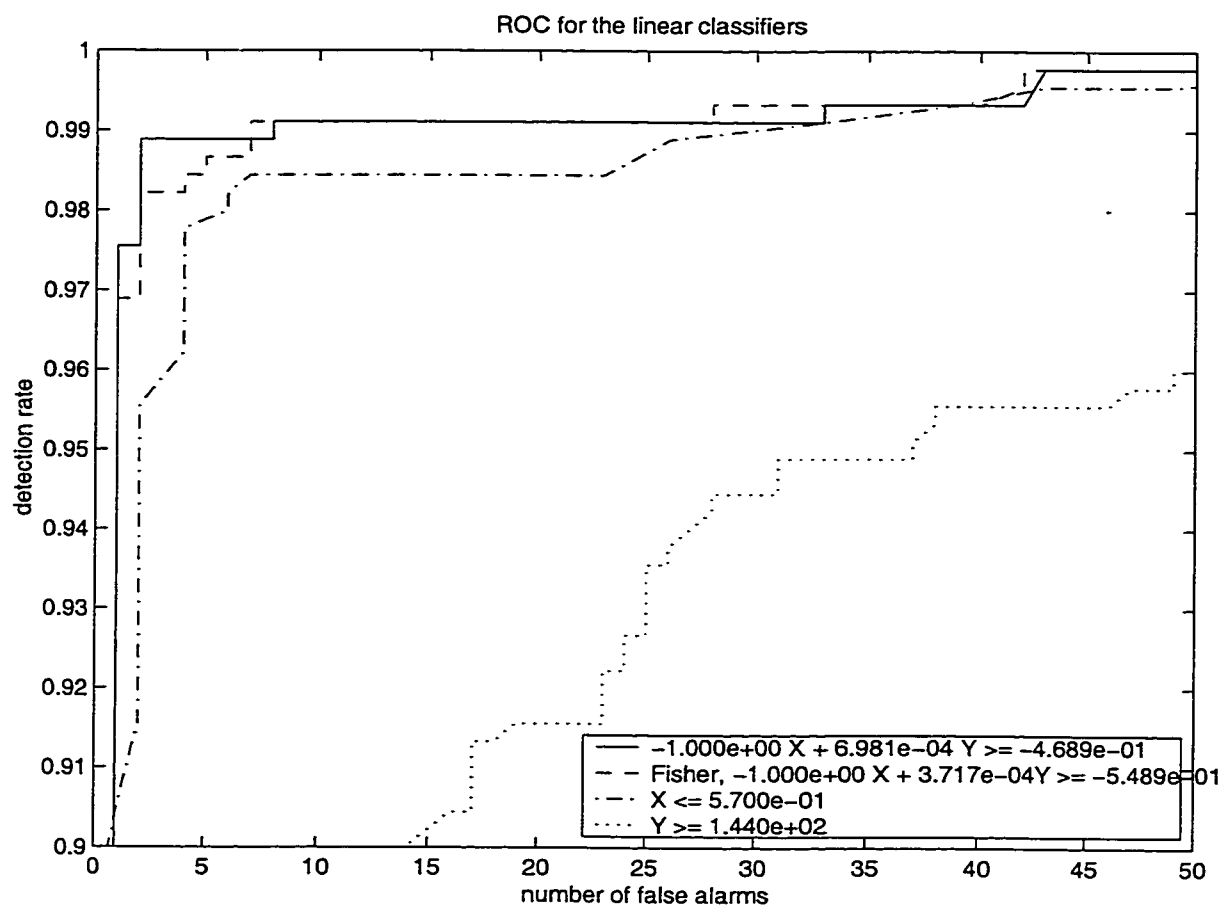


Figure 8.4: ROC curves for classifier performance in combining the centroid uncertainty and matched filter for FLIR ATR. Solid curve for the best linear classifier; dashed curve for the Fisher linear classifier; dash-dotted curve for centroid-only; dotted curve for matched-filter-only.

does a rather good job in combining them. The performance is further improved by the best linear classifier.

These performance data are obtained on the training data set consisting of target and background image chips. The performance of the best linear classifier on the testing data set of full-size FLIR images is reported in later sections.

8.3 Results and discussions

All algorithms tested in this experiment have only one tuning parameter, which is the final threshold for target declaration. Besides this parameter, there is no parameter to tune for any of the algorithms. (The combination coefficients for the combined algorithm, the one combining the centroid uncertainty and matched filter, are given in Table 8.1, and are held constant throughout the experiments.) The final threshold is varied in the experiments to give different levels of detection performance and hence does not need to a predetermined value. Therefore, no training data is needed.

As described earlier, nine sets of simulated FLIR images with 100 images per set are generated. These nine sets of images are characterized by three levels (low, medium and high) of target signature deviation and three levels (low, medium and high) of target relative contrast and dynamic range. (See Chapter 7 for details.) Each set contains a total of around 1,250 target instances. The performance of each algorithm on each test set is observed individually. The performance of different algorithms is compared on the same test sets. The effects of target signature deviation and relative contrast and dynamic range on the performance of individual algorithms are studied by comparing the performance of the same algorithm on the nine test sets.

Each algorithm described in Section 8.2 with the same thresholding procedure described in Section 8.1.2 is applied independently to the test data sets. The performance evaluation procedure described at the end of Chapter 6 is carried out to

Table 8.2: Best performance of the algorithm combining the centroid uncertainty and matched filter (comb-cen-mf) on the simulated FLIR test sets.

	low dev	med dev	high dev
low contrast and range	880	1055.5	1139.5
med contrast and range	427.5	440	507.5
high contrast and range	258	242.5	295

compare the output with the ground-truth and the performance measures are obtained. The thresholds of the algorithms are varied to give the algorithm detection performance in terms of the receiver operating characteristic (ROC) curve. There is one such curve for each algorithm on each test set.

Following the method outlined in Section 8.1.3 for the operating point selection, the optimal (least-cost) operating point is identified. The achieved minimum cost are shown in Tables 8.2 through 8.7.

To show the algorithm detection performance over the entire operating range, we also show the ROC curves for the algorithms on the test set with the medium level target signature deviation and medium level relative contrast and dynamic range. These curves are shown in Figure 8.5. The recognition performance and the localization performance for this test set are shown in Figures 8.6 and 8.7, respectively. In these plots, the centroid uncertainty based algorithm is denoted by “cencovdiratr,” the Maryland algorithm described in Section 8.2.1 is denoted by “matcher,” the matched filter algorithm is denoted by “tmatr,” the combined algorithm, which combines the centroid uncertainty and matched filter, is denoted by “comb-cen-mf,” the algorithms described in Sections 8.2.2 and 8.2.3 are denoted by “mlratr” and “polyspatr,” respectively.

Table 8.3: Best performance of centroid uncertainty based algorithm (cencovdiratr) on the simulated FLIR test sets.

	low dev	med dev	high dev
low contrast and range	978	1111.5	1139.5
med contrast and range	661	410	539
high contrast and range	330.5	354	328

Table 8.4: Best performance of matched filter (tmatr) on the simulated FLIR test sets.

	low dev	med dev	high dev
low contrast and range	1169.5	1260	1254.5
med contrast and range	709	736.5	928
high contrast and range	566.5	594	689.5

Table 8.5: Best performance of the Maryland algorithm (matcher) on the simulated FLIR test sets.

	low dev	med dev	high dev
low contrast and range	1097.5	1177	1225
med contrast and range	859.5	815.5	830
high contrast and range	516	481	444.5

Table 8.6: Best performance of the maximum likelihood ratio algorithm (mlratr) on the simulated FLIR test sets.

	low dev	med dev	high dev
low contrast and range	1114	1161	1186.5
med contrast and range	831	831	890.5
high contrast and range	563	582.5	589

Table 8.7: Best performance of the centroid uncertainty algorithm with 1-D cubic spline model for boundary uncertainty estimation (polyspatr) on the simulated FLIR test sets.

	low dev	med dev	high dev
low contrast and range	1225	1253	1239.5
med contrast and range	1166.5	1109	1145.5
high contrast and range	967	973	903.5

Table 8.8: Sensitivity of algorithm performance to image quality. α and β are the coefficients of the row and column indices in the planar fit to the performance data (minimum cost).

	α	β
combined algorithm	-380	62.8
centroid uncertainty	-369.4	6.17
matched filter	-305.7	71.2
Maryland algorithm	-343.0	4.4
likelihood ratio algorithm	-287.8	26.3
variant of the centroid algorithm	-145.7	-11.7

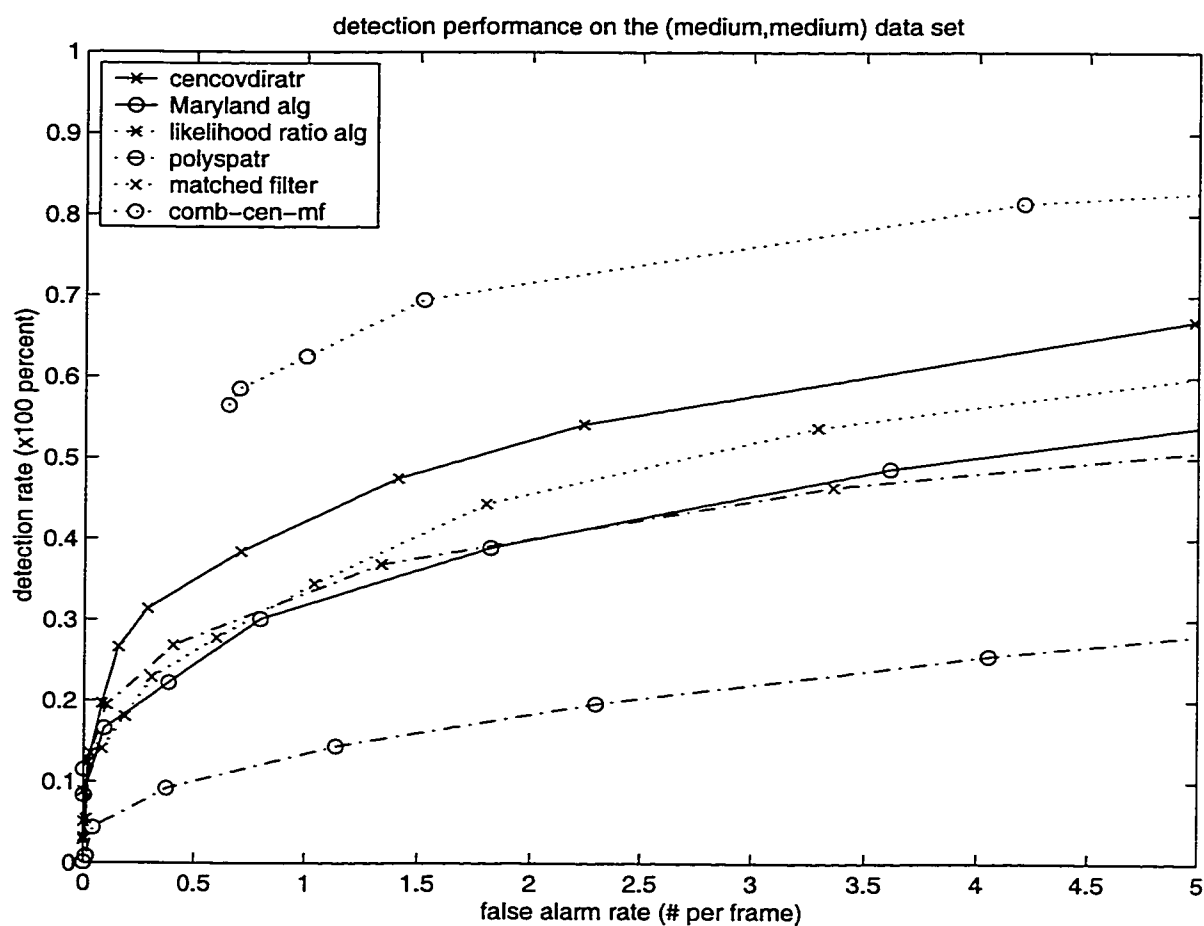


Figure 8.5: Detection performance (ROC curves) of the ATR algorithms.

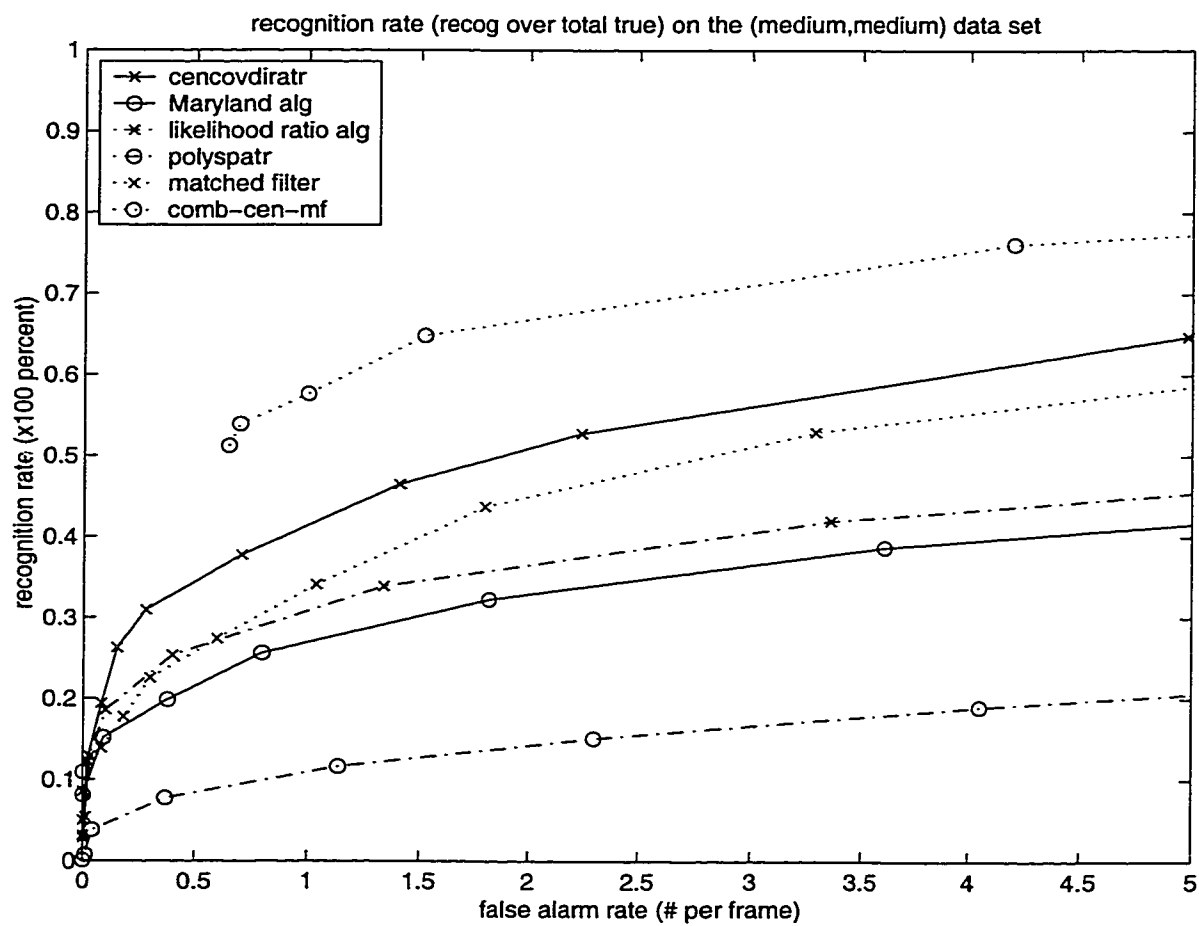


Figure 8.6: Recognition performance of the ATR algorithms.

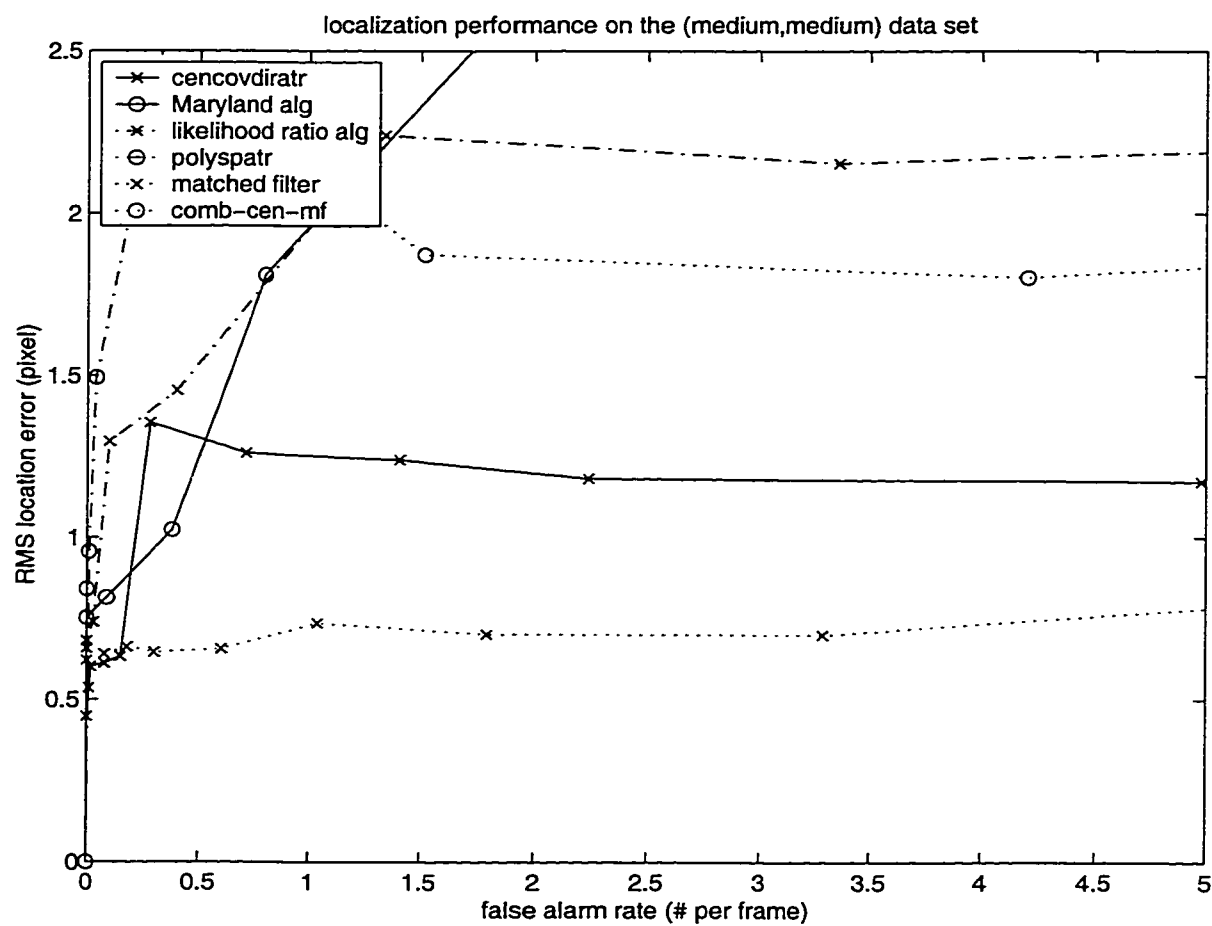


Figure 8.7: Localization performance of the ATR algorithms.

8.3.1 Discussions

- From the performance data in Tables 8.2 through 8.7 and in Figures 8.5 and 8.6, we see that the centroid algorithm uniformly out-performs the competing algorithms in terms of detection and recognition performance over the entire operating range, except the algorithm which combines the centroid uncertainty with the matched filter. The combined algorithm successfully combines the information extracted by the two algorithms and results in a very significantly improved performance in both detection and recognition.

For the test data set for medium contrast and medium level of target appearance deviation, at the optimal operating point, the centroid algorithm has a detection rate 67% with 5 false alarms per frame. The best competing algorithm has detection rate 54% with 3.3 false alarms per frame. The algorithm combining centroid uncertainty and matched filter has detection rate 81% with 4.2 false alarms per frame.

The reason for not giving the detection performance data of these algorithms at the same false alarm level is that we adopt the minimum cost criterion for measuring the algorithm performance, instead of the constant false alarm rate (CFAR) criterion. The performance data reported above are associated with the optimal (least-cost) operating point on the respective ROC curves. In computing the cost of errors, the cost of two false alarms is equal to the cost of one misdetection. (See Section 8.1.3 for details.)

- The best localization performance for the correctly recognized targets is achieved by the matched filter. The next best is the centroid uncertainty algorithm. The localization performance of the combined algorithm is worse than both component algorithms. However, it is still better than other competing algorithms.
- The sensitivity of the tested algorithms with respect to the image quality is

studied by least-squares fitting of planes to the given performance data. Let r and c denote the row and column indices of the table, and let the indices take the value from $\{-1, 0, 1\}$ for the left, center and right column, or the top, center and bottom row, respectively. The slope of the fitted plane along the row and column axes are denoted by α and β , respectively. Their values show the sensitivity of the algorithm performance with respect to the target contrast and dynamic range relative to the background, and to the amount of deviation in the target appearance. These values are listed in Table 8.8. It is interesting to observe that the best performing algorithm, which is the algorithm combining centroid uncertainty and matched filter, is the most sensitive to the target contrast and dynamic range. This is due to the sensitivity of both the centroid uncertainty and the matched filter.

The combined algorithm is also highly sensitive to the deviation to target appearance, only second to the matched filter which is expected to be very sensitive. This shows the influence of the matched filter in the performance of the combined algorithm.

- The confusion matrices for the centroid and the Maryland algorithms are given in Tables 8.9 and 8.10. These matrices reflect the recognition performance of the algorithms on the test data set with medium contrast and medium target signature deviation, and at the algorithms' respective optimal operating points (67% detection with 5 false alarms per frame for the centroid algorithm, and 49% detection with 3.6 false alarms per frame for the Maryland algorithm). From these confusion matrices, we see that the centroid algorithm has much better recognition performance by making a very small number of recognition errors other than misdetection and false alarms. In comparison, the Maryland algorithm makes significantly more recognition errors. The overall correct recognition rates for all true targets are 65% for the centroid algorithm and 39% for

Table 8.9: Recognition performance (confusion matrix) for the centroid uncertainty algorithm at its optimal operating point on the ROC curve. t_1, \dots, t_9 are the true target ID's. d_1, \dots, d_9 are the declared target ID's. The column headed by MD is for the numbers of misdetected target instances. The row headed by FA is for the numbers of false alarm instances.

	MD	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8	d_9
FA		91	24	148	67	51	59	9	27	22
t_1	79	65	1	2	2	0	2	0	2	0
t_2	39	0	100	2	0	0	0	0	0	0
t_3	66	1	0	82	0	0	1	0	0	1
t_4	50	2	1	0	84	0	2	1	0	0
t_5	37	0	0	0	0	88	0	0	0	1
t_6	52	0	0	0	1	0	71	1	0	1
t_7	14	1	0	0	1	0	0	105	0	0
t_8	51	0	0	0	0	0	0	0	98	0
t_9	22	0	0	0	0	0	0	0	0	108

the Maryland algorithm.

8.4 Experiments on uncooled FLIR image sequence data

To have a broader view of the performance of the ATR algorithms, another set of experiments were conducted on some image data from a newly developed FLIR sensor. This new sensor was developed at Texas Instruments. It is an uncooled FLIR device that has very low cost. It is known to produce imagery with much lower quality than most second generation FLIR sensors currently in use.

The video sequence t613loop1 from the Raytheon Company¹ contains 340 frames

¹Courtesy of Dr. Mary Cassabaum and Dr. Harry Schmitt

Table 8.10: Recognition performance (confusion matrix) for the Maryland algorithm at its optimal operating point on the ROC curve. t_1, \dots, t_9 are the true target ID's. d_1, \dots, d_9 are the declared target ID's. The column headed by MD is for the numbers of misdetected target instances. The row headed by FA is for the numbers of false alarm instances.

	MD	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8	d_9
FA		23	45	24	54	46	18	87	15	49
t_1	100	23	1	1	10	2	3	10	0	3
t_2	86	0	37	4	0	4	1	3	2	4
t_3	100	0	9	18	2	5	1	8	3	5
t_4	55	0	0	0	78	0	2	5	0	0
t_5	75	0	2	0	0	39	1	5	4	0
t_6	83	1	1	0	7	0	28	4	1	1
t_7	13	0	0	0	0	0	0	108	0	0
t_8	62	1	0	0	0	1	0	0	85	0
t_9	61	1	1	0	0	0	1	3	0	63

FLIR images. The sequence shows a moving tank target imaged by a moving camera. The original images are of size 256×256 , and the pixel intensity value is quantized into 16 bits. These images are slightly modified by first cropping to 256×236 in size and then using the trimmed linear compression to convert the intensity values to 8 bits. In many of the images, the tank target is present and not significantly occluded. In others, the tank target is either present but with some parts occluded by the terrain, or not present in the image at all. Out of the 340 frames, there are 244 frames where the target is either not occluded, or not significantly occluded.

This video sequence is more appropriate for testing the tracking module, as opposed to the acquisition module considered in this dissertation study, of an ATR system. Since the target appearance does not change very significantly from scene to scene, except for some noticeable change in its pose, the matched filter algorithm is very effective in detecting the target. Other boundary-based ATR algorithms considered in our study for target acquisition are based on the assumption that the target signature is too unreliable to be used. This assumption simply does not hold in this tracking application. The price being paid here is the loss of significant information from the target signature, which results in significantly lower target detection performance as compared to the matched filter. However, even in this case, the algorithm that combines the matched filter with the boundary based centroid uncertainty (to be referred to as the “combined algorithm”) still shows some significant performance improvement over the simple matched filter algorithm.

In the following, we describe some more details of the experiment of applying the ATR algorithms to both the image sequence and the data sets created from this image sequence by the FLIR scene simulation procedure (Chapter 7). In this set of experiments, we are primarily concerned with target detection. Target pose estimation is not of interest.

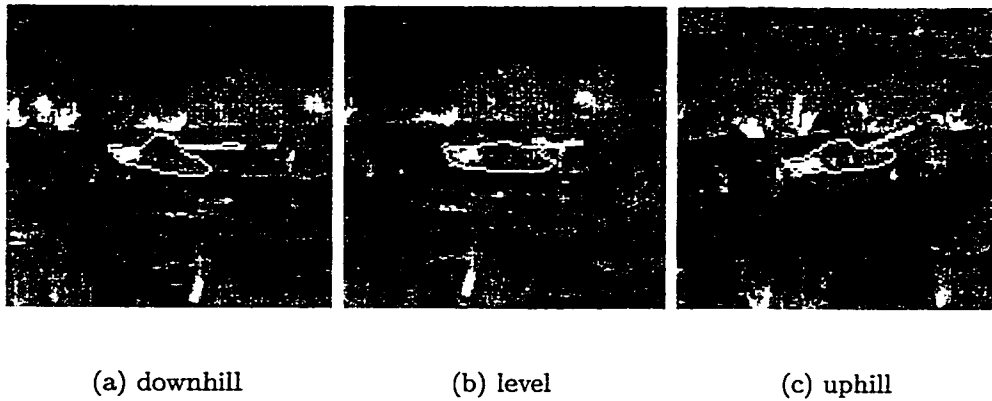


Figure 8.8: Target instances for three different poses used in modeling the target in the Raytheon data experiment.

8.4.1 Target representation

Since the target undergoes significant changes in its pose over the entire sequence, three instances are chosen for representing its three typical poses: downhill, level and uphill. These are chosen from the frames 1, 11 and 135, respectively. The sub-images containing the target in these frames are shown in Figure 8.8, with the boundary silhouette of the target overlaid.

For each of the target poses, the matched filter algorithm uses the center part of the sub-image (without the overlaid boundary silhouette) of size 61 columns by 21 rows for representing the target with that pose. The boundary-based algorithms all use the boundary silhouette, without the part for the protruding gun barrel, for representing the target with that pose. The reason for not including the gun barrel part is two-fold. First, the three selected poses only constitute a very rough approximation for all poses of the target in the entire sequence. The position of the long protruding gun barrel is very sensitive to the actual pose of the target. When the actual target pose is different from all three selected poses, the gun barrel is significantly

away from that with the three poses. This adds significantly to the modeling error by the three poses. On the other hand, the body part of the silhouette is less sensitive to the pose. Secondly, the boundary based algorithms all assume more or less independent measurements along the target silhouette. The gun barrel of the target in this sequence is only one to two pixels wide. Boundary points on the opposite sides of the gun barrel are just right next to each other, or even are the same points. This adversely worsens the violation of the independence assumption of the algorithm, and hence should be avoided. For these reasons, to reduce the modeling error, the gun barrel is not included in the target silhouettes used for modeling the target in the boundary based algorithms.

8.4.2 ATR performance on the image sequence

The ATR algorithms examined in the experiments of last section were applied independently to the 340 frames in the image sequence. Each of the algorithms only has one tuning parameter, which is the threshold on the final target saliency measure. For each threshold value used, the algorithm output is compared by the Hungarian algorithm based performance evaluation procedure (Chapter 6, [91, 92]) against the ground-truth gathered manually about the target in the image. The table of the Euclidean distances squared between the ground-truth and declared targets is used in the procedure. The threshold on the table entries for determining detection is set to 100 pixel squared, corresponding to searching a circular area of radius 10 pixels around the centroid of each ground-truth target for a declared target.

Numbers of misdetection and false alarms are determined for each image independently, and added together for all frames to give the detection performance over the entire sequence. Different values for the threshold is used to give different levels of trade-off between detection and false alarm performance as reflected by the receiver operating characteristics (ROC) curve. The statistical significance of the differences between important ROC curves is assessed by the Z -test for the equality of two pro-

portions [70].

Figure 8.9 shows the image of the first frame of the sequence with the true target boundary overlaid. Also shown is the detection result on this image by the combined algorithm at a threshold level giving the detection rate of 0.62 and the false alarm rate of 0.93 false alarms per frame.

Figure 8.10 shows the ROC curves of the algorithm on the sequence. The low performance of all the algorithms is due to the low contrast of the target in the sequence: the median values of the relative contrast and dynamic range (see Chapter 7 for their definitions) are as low as 0.56 and 1.09, respectively. (As a comparison, the medians for the MURI FLIR data set are 1.76 and 2.02.)

The Z -test is conducted to examine whether difference in the detection rate between the algorithms is statistically significant. This is conducted at the false alarm level of approximately 1.07 false alarms per frame. At this false alarm level, the detection rates are: 0.58 for the combined algorithm; 0.4 for matched filter alone; 0.13 for centroid uncertainty alone; 0.096 for the Maryland algorithm. Table 8.11 shows the results for the tests for some important pairs. From these results, we see that the improvement in detection rate by the combined algorithm over the matched filter algorithm is very significant. The improvement for the matched filter algorithm over the centroid algorithm is also very significant. There is not much significance in the amount of improvement in detection performance by the centroid algorithm over the Maryland algorithm as measured by this experiment.

8.4.3 ATR performance on image data simulated from the sequence

To conduct experiment with the image data appropriate for testing algorithms for the acquisition module, we generated simulated FLIR scenes from the t613loop1 sequence using the procedure described in Chapter 7. Relative contrast and dynamic range of the inserted target instances are set to those estimated from the target image chips in the sequence. The amount of texture added to the target signature



(a) true target



(b) detection result by combined algorithm

Figure 8.9: First frame in the t613loop1 sequence and detection result by the combined algorithm. This result is from a threshold giving a detection rate of 0.62 and false alarm rate of 0.93 false alarms per frame over the entire sequence.

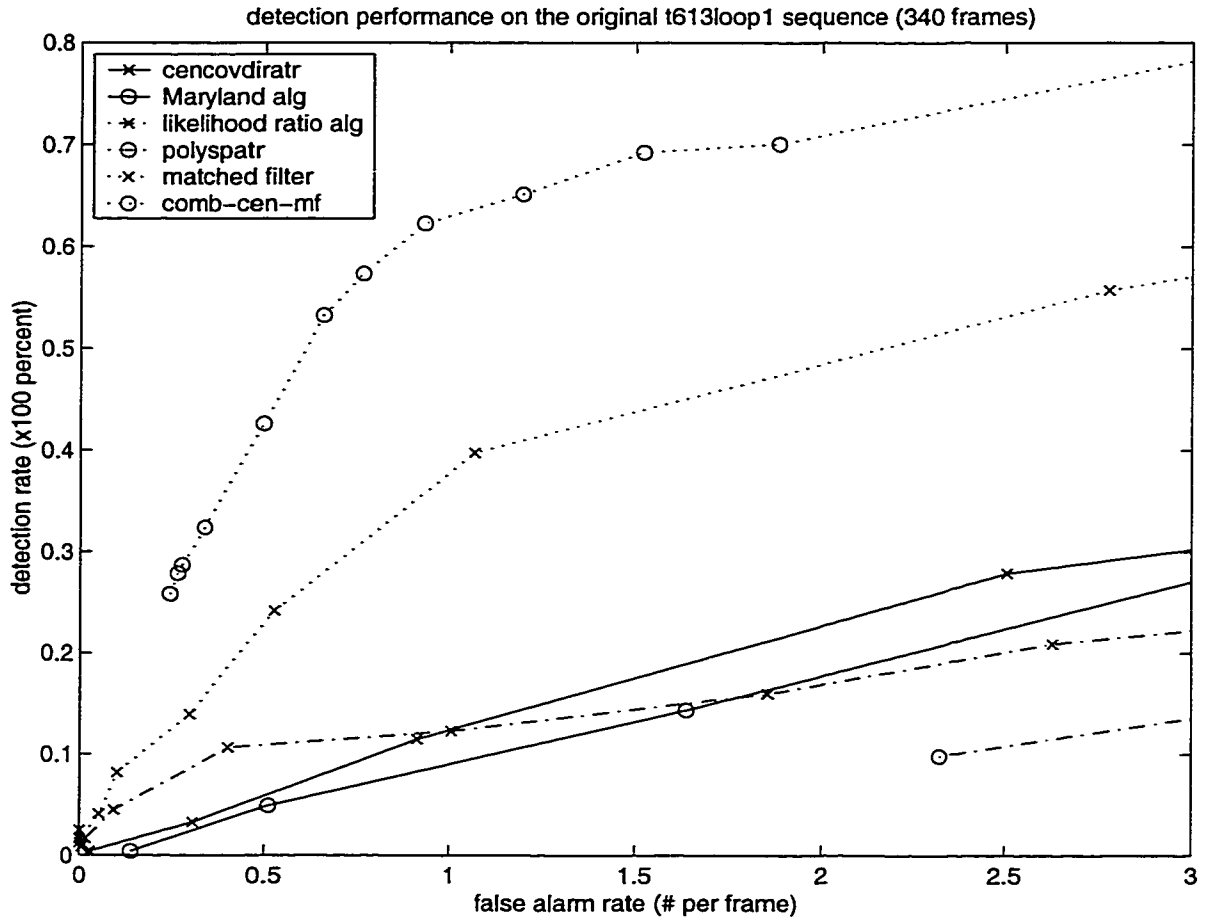


Figure 8.10: Detection performance of the ATR algorithms on the t613loop1 sequence (340 frames and 244 total target instances).

Table 8.11: Results from Z -test for assessing the statistical significance of the difference in the the detection rates.

algorithm pair	Z -value	p -value
comb-cen-mf and mf	3.98	3.49×10^{-5}
mf and centroid alg	6.73	$< 10^{-10}$
centroid and Maryland	1.22	0.11

is varied in different runs. There is no zooming or rotation of the target signature in the experiments described here. In each selected scene, simulated target chips are inserted in the scene so that each scene has 12 to 15 target instances. 100 scenes are generated for each run. There are around 1300 target instances in each set of such 100 scenes.

Figure 8.11 shows one example of the simulated image with the true target boundaries overlaid. This example is from the data set generated with the texture amount of 0.1. Although it might look unrealistic to have so many targets in a single scene, each individual insertion of the simulated target signature seems to fit well in the scene. The reason for having so many target instances in a scene is two-fold. First, in our ATR research, the emphasis is on detecting individual targets. The spatial relationship between target instances, except for overlapping, is not concerned by the algorithms being considered. Therefore, having many target instances in a scene does not bias the relative performance of the algorithms being tested, provided that the same data set is used for all algorithms. Secondly, in order for the detection performance measured on such a data set to carry statistical significance, we need to have some moderately large number of target instances in the data set. Having many target instances in one scene can reduce the number of images required in such a data set. This in turn reduces the time required in conducting the experiment.

Also shown in Figure 8.11 is the detection result by the combined algorithm on this image at the false alarm rate of 0.94 false alarm per frame.

Figure 8.12 shows the ROC curves of the algorithms on the simulated data set from adding 10% of texture to the target signature. Comparing with Figure 8.10, we see that there is a very significant drop in detection performance of the matched filter algorithm. This is due to amount of texture added to the target signature, which causes significant differences in the appearance between the templates used by the matched filter algorithm and the target instances inserted in the images. The Z -test is conducted to assess the significance in the difference in detection performance



(a) true targets



(b) detection result by combined algorithm

Figure 8.11: Example of simulated scene and detection result by the combined algorithm. This example is from the simulated data set with the texture amount of 0.1 added to the target signature. The detection result is from a threshold giving 0.94 false alarm per frame on the data set.

between the centroid algorithm and the Maryland algorithm. At the false alarm rate of approximately 1.1 false alarms per frame, the detection rates are 0.22 and 0.17 for the centroid algorithm and Maryland algorithm, respectively. The total number of target instances is 1332. The Z -test gives $Z = 3.26$ with the p -value of 0.0006. Hence there is strong evidence that the detection performance of the centroid algorithm is significantly better than the Maryland algorithm. From this calculation, we can also easily conclude that the combined algorithm has very significant better performance than all other algorithms.

The ROC curves of the algorithms on some other test data sets generated with more amounts of texture added to the target signature are shown in Figure 8.13 (for texture amount of 30%) and 8.14 (for texture amount of 50%). It can be seen clearly that, with increasing amounts of deviation in the signature of the target, the performance of the matched filter algorithm degrades more and more. On the other hand, the boundary based algorithms which were designed to be robust to this kind of deviation are much less affected by this change. The combined algorithm remains by far the best performing algorithm. The centroid algorithm remains the second best algorithm.

8.5 Experiments on vehicle detection in aerial imagery

To test the generality of the centroid uncertainty based algorithm for automatic target detection and recognition problems, we also conduct experiments to observe its performance in the vehicle detection application [86, 114, 85]. In this application, vehicles of a specified size and orientation are to be detected from vertical view aerial images. In our experiments, vehicles of length 13 ± 5 pixels, width 5 ± 2 pixels, and orientation $\frac{\pi}{2} \pm 0.1$ radians are designated as target vehicles. A perfect vehicle detection algorithm should detect all such vehicles and nothing else from the any given image. Figure 8.15 shows an example image in this application. A proposed system

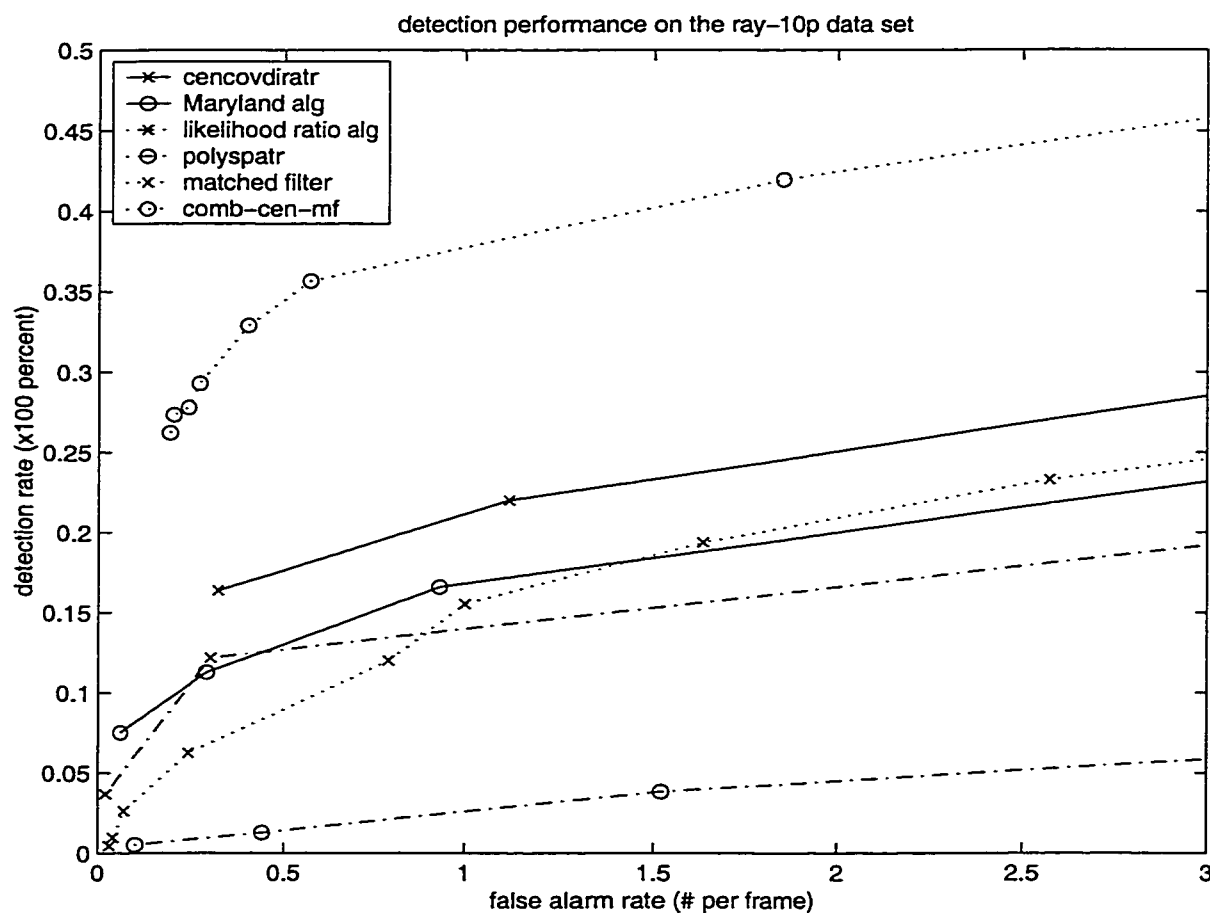


Figure 8.12: Detection performance of the ATR algorithms on data simulated from the scenes in the t613loop1 sequence. 100 scenes simulated with a total of 1332 target instances. Target signature contains 10% of texture. Target relative contrast and dynamic range set to the values estimated from real target image chips from the t613loop1 sequence.

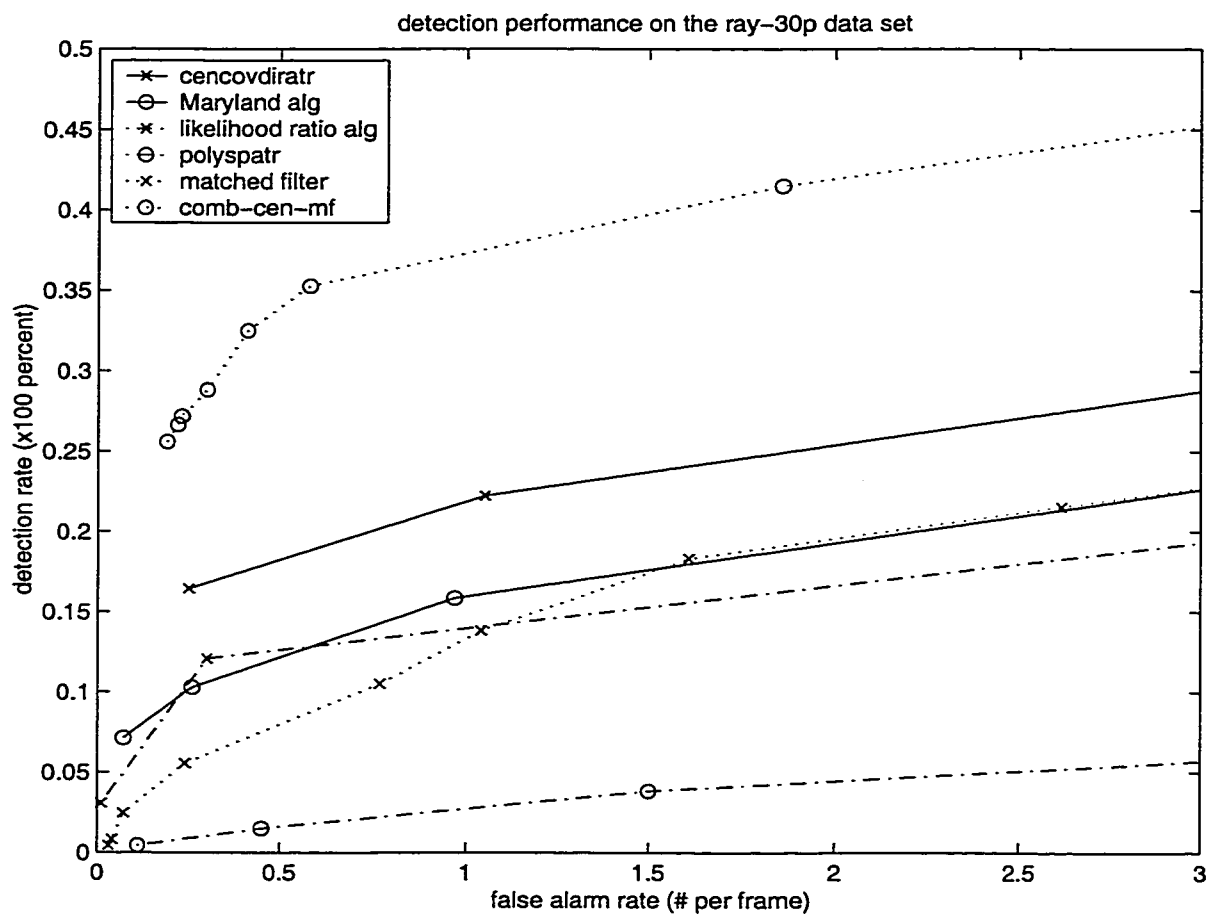


Figure 8.13: Detection performance of the ATR algorithms on data simulated from the scenes in the t613loop1 sequence with 30% texture. Target relative contrast and dynamic range set to the values estimated from real target image chips from the sequence.

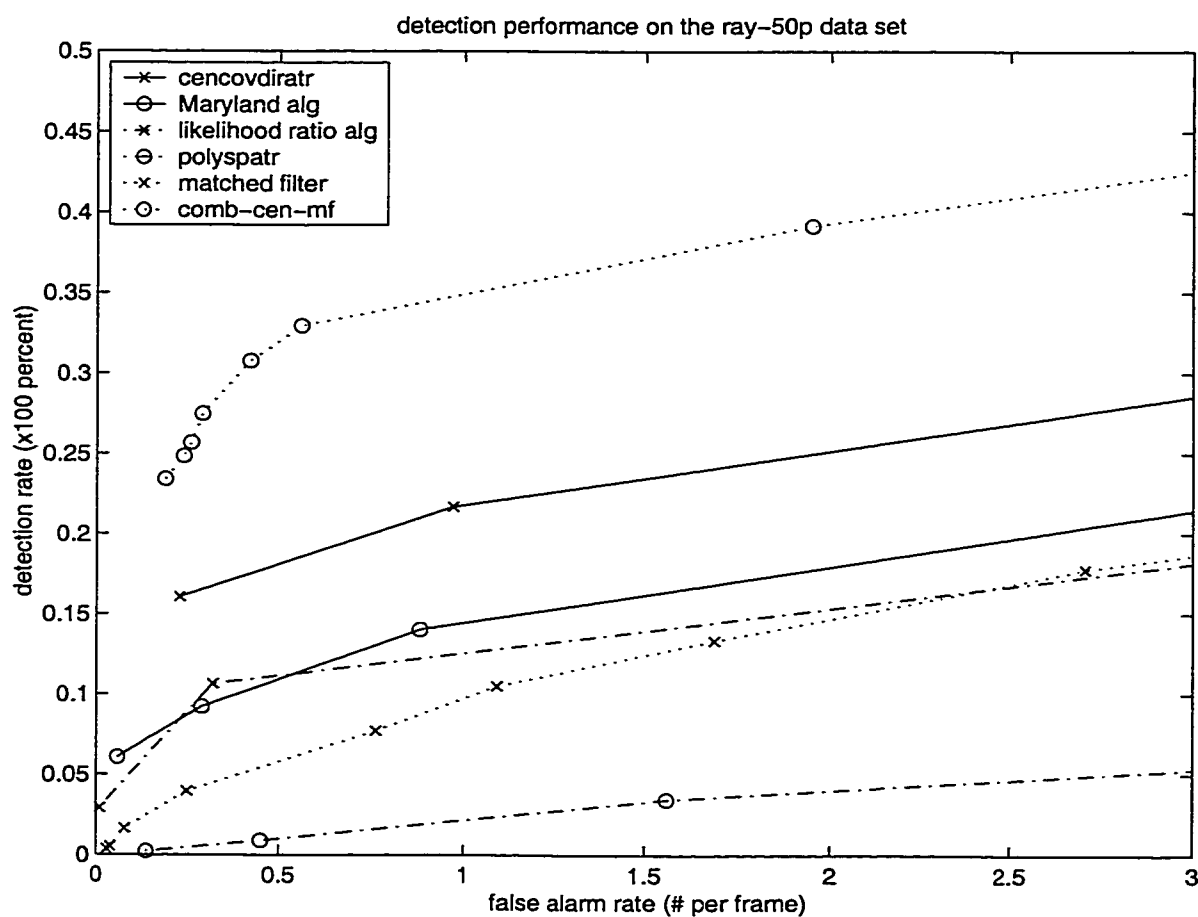


Figure 8.14: Detection performance of the ATR algorithms on data simulated from the scenes in the t613loop1 sequence with 50% texture. Target relative contrast and dynamic range set to the values estimated from real target image chips from the sequence.

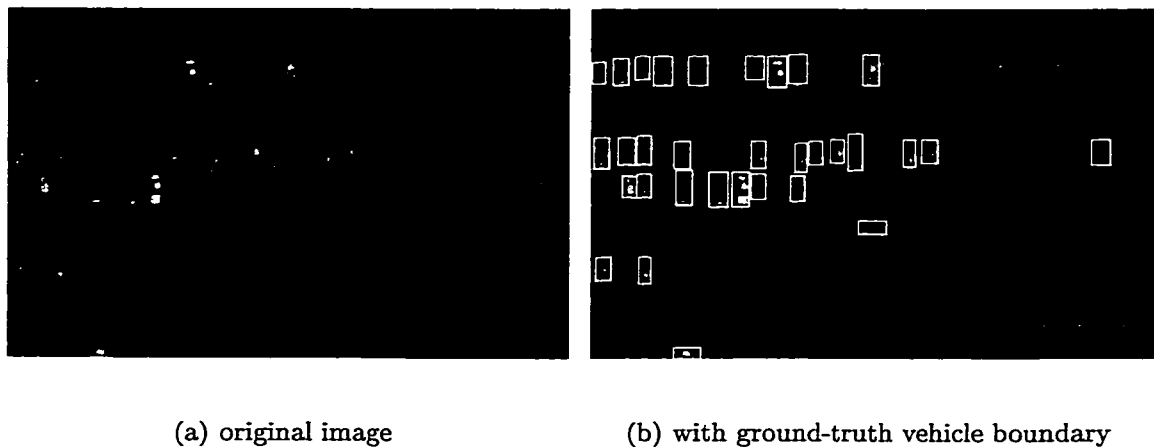


Figure 8.15: Example aerial image for vehicle detection.

[114] for this application suffers greatly from the complex background of natural and man-made objects in the imagery. The relative small size of the vehicles (around 50 to 150 pixels in area) also adds to the difficulty. The result is that the false alarm rate for the system output is unacceptably high. The current research effort is mainly on reducing the number of false alarms.

Parameswaran *et al* [114] propose a system that consists of algorithm modules for edge detection, generalized Hough transform (GHT) [7, 83, 137, 153] and post-processing. The problem with this system lies in its bottom-up nature. The local nature of edge detection makes the resulting edge map very unreliable on the aerial imagery due to the complex scene content. The quality of the detected edge map cannot meet the high requirements by the subsequent GHT operation. As a consequence, the GHT produces large numbers of false alarms [46]. The observed performance of the system on our real test data is not good: the detection rate is less than 20% at the false alarm rate of 50 false alarms per million-pixel image area.

The system performance is significantly improved in [86] where a classical pattern recognition approach is taken. Using the VVFH data set [94, 95, 85], which is a mod-

erately large set of aerial images with manually obtained ground-truth, eight features are extracted for the vehicle detection purpose. Six of the features are the average intensity and gradient magnitude values for the pixels inside, along the boundary of, and in the outside around the vehicles. Two more features are the averages of the projected values of the gradient magnitude and direction unit vector for pixel locations along the vehicle boundary [86]. These eight raw features are used to construct a Bayesian table-look-up classifier to be referred to as the 8-feature LUT classifier. In this classifier, the eight raw are decorrelated and reduced to three by the principle component analysis (PCA) [43] where the first three PCA coefficients are retained as useful features. A three-dimensional look-up table is constructed to be a non-parametric representation of the probability distribution of the resulting feature vectors. (All data points, referred to by "total" in Table 8.12, are used in constructing this 8-feature classifier. For the 9-feature classifier to be described below, only those covered by Part I are used in training.) Specifically, a look-up table of fixed size is constructed where the numbers of quantization levels along the three axes are proportional to the entropy along the axes. The quantization boundaries on the three axes are determined independently by the equal probability quantization (EPQ) [58] on the data projections on the three axes. The vehicle and background class conditional probabilities are estimated by smoothing and normalizing the counts of data samples in the quantization bins. The vehicle class prior probability is specified as 0.01. The economic gain matrix is varied to give vehicle detection performance over a range of false alarm levels. A Bayesian classifier using these data judges whether a test feature vector represents a target vehicle or not.

The performance of two FLIR ATR algorithms, namely the Maryland algorithm and the centroid uncertainty based algorithm, on the VVFH data set are obtained using exactly the same performance evaluation procedure as used in [86]. The performance evaluation procedure is based on the ground-truth information of 25,124 target vehicles in 375 images. These images cover a total of about 200 million pixels.

Table 8.12: Some statistics for the VVFH data set. Size is the total number of pixels in millions; # T.V. stands for the number of target vehicles whose dimension and orientation meet specifications. Part I consists of fhn75, fhn78, fhn711, fhn715. Part II consists of fhn713, fhn717, fhn719.

	size	sub-images	# T.V.
fhn75	23.8	30	1464
fhn78	32.8	84	1745
fhn711	23.0	24	2875
fhn713	28.5	31	3638
fhn715	45.0	60	5641
fhn717	22.2	55	5438
fhn719	24.1	51	4323
total	199.4	335	25124
Part I	124.6	198	11725
Part II	74.8	137	13399

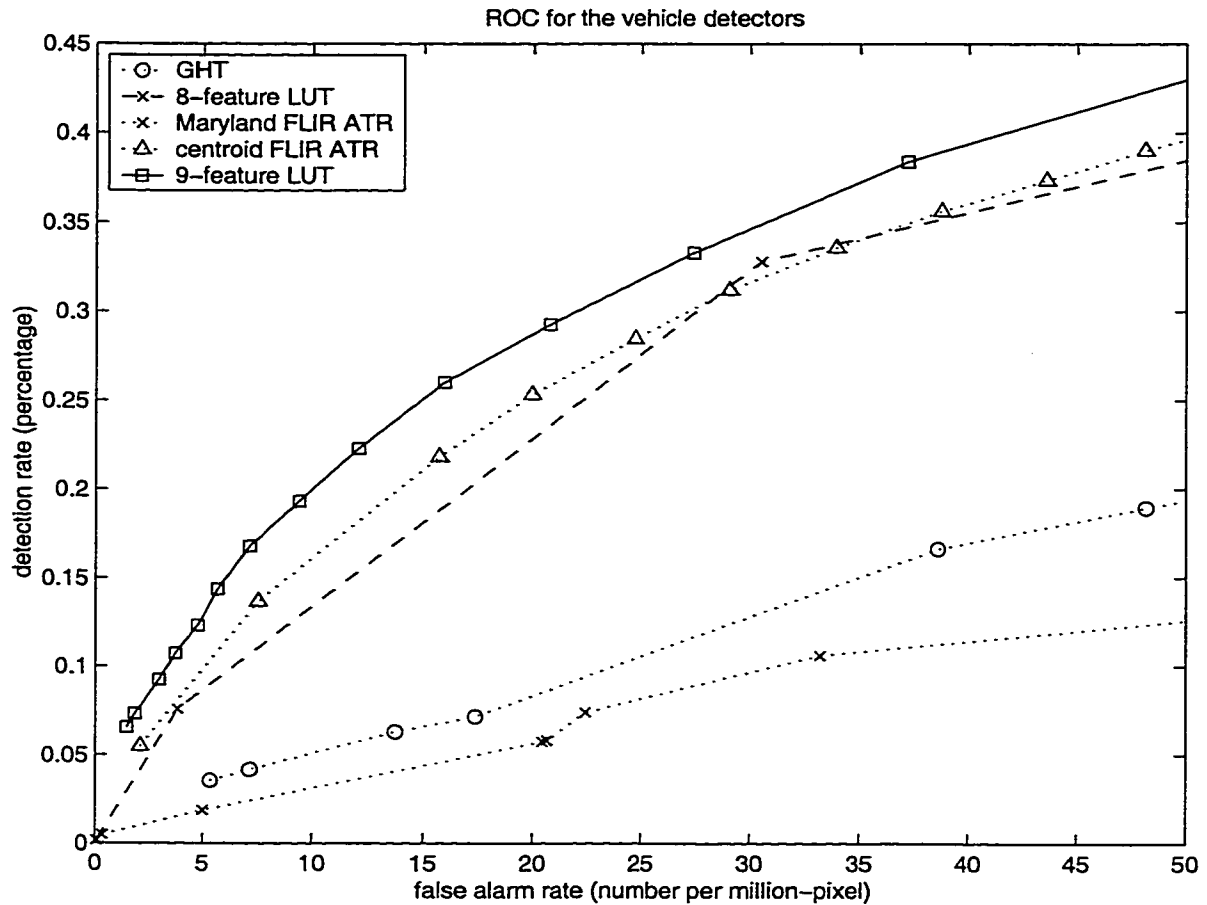


Figure 8.16: ROC curves for the vehicle detection application.

The dimension of a pixel in the image corresponds to the length of 0.31 meters on the ground. The ROC curves of the two algorithms, along with those for the GHT based algorithm and the 8-feature LUT classifier, are shown in Figure 8.16.

From the ROC curves in Figure 8.16, we observe that the centroid uncertainty based algorithm performs relatively well, and the Maryland FLIR ATR algorithm does not fit well into this application. In particular, the performance of the centroid uncertainty based algorithm is about the same as the 8-feature LUT classifier at the first iteration. This is fairly good performance considering the fact that there is no

training of any kind done for this algorithm on the data set, while the 8-feature LUT classifier totally relies on the probability distributions estimated from the data.

Since the Bayesian LUT classifier is a very general framework, the centroid uncertainty can as well be used as an extra feature to be used in the classifier and improve vehicle detection performance. This is done, and the resulting algorithm is referred to as the 9-feature LUT classifier. This classifier is constructed in exactly the same manner as in the 8-feature case. The only difference is in the dimensionality of the raw input data vectors. As can be seen from the ROC curves in Figure 8.16, the performance of the 9-feature LUT classifier is further improved over that of the 8-feature LUT classifier. The significance of the improvement is confirmed by the Z -test results: at the false alarm level of 30.5 per million pixel, the p -value is 3.28×10^{-7} ; at the false alarm level of 2 per million pixel, the p -value is 4.83×10^{-20} .

Chapter 9

CONCLUSION

9.1 *Brief summary*

In this dissertation, we presented a theory on the use of location uncertainty information in detection and recognition problems of computer vision applications. In particular, we applied the theory to the problem of automatic target detection and recognition. To accommodate the great amount of change in the target appearance and scene content, a boundary model is used for the target, and the target centroid location uncertainty measure is computed that is related to a weighted combination of boundary contrast. In computing the centroid uncertainty measure, relative importance of target boundary points are determined geometrically by their contribution to the centroid location uncertainty. Targets with strong visual appearance are characterized by small centroid location uncertainty.

The theory of the location uncertainty information consists of three major pieces. These are the propagation of location uncertainty from boundary points to the centroid, the representation of contrast at boundary points by geometric location uncertainty, and the optimal estimation of image gradient. These new contributions are all based on sound mathematical formulations. In the location uncertainty propagation, a polygon model is used for the target silhouette and the linearization of the perturbation is used for estimating the centroid location covariance matrix. The boundary point location uncertainty estimation is carried out in the edgel detection framework, and the relationship between the signal-to-noise ratio and the location uncertainty is computed and tabulated. A minimax optimization problem is formulated for de-

signing the optimal gradient operator used in signal-to-noise ratio estimation. The criteria function to be minimized is the worst-case mean-squared error in the gradient orientation estimate. Meaningful results from these studies are useful not only in the ATR application, but in other computer vision applications as well. For example, the obtained optimal gradient operator can be used to improve various edge detection or optical flow estimation modules, which often appear as earlier stage modules in computer vision algorithms. The edgel location uncertainty model can be used to improve the precision of Hough transform or be used directly in model-based curve detection. The centroid location uncertainty can be used directly in the biological cell counting module of medical imaging systems.

The normalized centroid uncertainty as a feature for target detection and recognition proved to be very effective in the experiments with synthetic and real imagery. The ATR algorithm using this feature performs significantly better than competing algorithms in all three aspects of the ATR algorithm performance, i.e., detection rate, recognition rate, and localization accuracy. The centroid uncertainty feature is intended to be a powerful addition to existing ATR algorithm modules. The preliminary study of its combination with the traditional matched filter shows significantly further improved target detection and recognition performance.

In characterizing the detection and recognition performance of computer vision algorithms, a new methodology is developed to overcome some problems with existing methods. The center pieces of the new development are the formulation of the situation by an optimal matching problem, and transforming the problem into an unconstrained assignment problem. The Hungarian algorithm is used to solve the resulting problem efficiently. Based on the established one-to-one correspondence between ground-truth and declared entities, the computed performance measures are more precise indications of the algorithm performance. Due to the similar nature of the function performed by detection and recognition algorithms, the general methodology developed here is readily applicable to the evaluation of a wide range of algo-

rithms for detection and recognition applications, such as algorithms for human and face detection, biological cell detection and recognition, and vehicle detection and recognition.

9.2 *Short-comings and future work*

General targets are three-dimensional objects and can have infinite numbers of two-dimensional silhouettes depending on range and pose relative to the imaging sensor. The modeling of targets by the silhouette requires prior knowledge of the target pose. Otherwise, a large number of silhouettes corresponding to all poses and ranges of interest need to be examined. This restricts the use of the algorithm in real applications. The remedy for this problem could possibly be found in combining the work with the model-based active contour formulations such as Friedland and Rosenfeld [42].

The interpretation of the centroid uncertainty feature as a weighted combination of boundary contrast is quite conservative. The use of localization information in general signal detection applications may well deserve its own focus of study. Currently, understanding in this area is still limited. Localization is customarily considered as being secondary to detection, and studied in a framework following the detection stage. We have demonstrated that the localization measurement can be very useful right in the detection stage. Deeper understanding of this topic can possibly lead to brand new interpretations of the centroid feature, taken entirely from the localization point of view.

In our experiments, a general linear classifier has already succeeded in combining boundary based and intensity based features to give significantly improved target detection performance. We believe that further improvement will result from better combination of the features, which requires more detailed specification of targets in terms of both geometric and photometric properties.

The fourth class gradient operator obtained in Chapter 4 by numerically optimizing the worst-case error criterion function over the full parameter space is a very attractive candidate for applications involving gradient or contrast estimation. However, in order for it to be useful in real applications, an effective method needs to be developed to estimate the local image signal-to-noise ratio in a way that is compatible with this gradient operator.

The bi-variate polynomial fitting based image gradient estimation approach is very appealing in that it provides a natural means for estimating the local signal-to-noise ratio in the imagery. It is popular to use a fixed third-order polynomial for the fitting. A more natural consideration is to let the order of the fit adapt to the local image neighborhood. In the appendix, we briefly describe our initial efforts in using the minimum description length principle to adaptively choose the fitting order. Some detailed issues related to the precision of the data need to be carefully studied before the image signal-to-noise ratio can be more accurately and robustly estimated.

Since the centroid uncertainty model ranks the relative importance of boundary points from a global view of the entire boundary, it is a very promising foundation for developing a good polygon approximation algorithm that can overcome problems associated with local operations of many existing algorithms. The minimal description length principle should also be a very useful resource in this development.

The results on the optimal gradient operator design and the relationship between edge signal-to-noise ratio with the location uncertainty can be readily applied to curve detectors such as the Hough transform and the model-based curve segment detectors.

BIBLIOGRAPHY

- [1] Milton Abramowitz and Irene A. Stegun, editors. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*. National Bureau of Standards, 1972.
- [2] J. K. Aggarwal, editor. *Multisensor Fusion for Computer Vision*. Springer-Verlag, 1991.
- [3] Farid Amoozegar. Neural-network-based target tracking state-of-the-art survey. *Optical Engineering*, 37(3):836–46, Mar 1998.
- [4] Theodore Wilbur Anderson. *An introduction to multivariate statistical analysis*. Wiley, 2nd edition, 1984.
- [5] Guy Aviram and Stanley R. Rotman. Evaluation of human detection performance of targets embedded in natural and enhanced infrared images using image metrics. *Optical Engineering*, 39(4):885–96, Apr 2000.
- [6] Simon Baker and Shree K. Nayar. Global measures of coherence for edge detector evaluation. In *CVPR'99 Proceedings*, pages 373–9, June 1999.
- [7] Dana H. Ballard and Christopher M. Brown. *Computer Vision*. Prentice-Hall, 1982.
- [8] Terry Bartlett. Simplified IR signature prediction for model-based ATR. In *Proc. SPIE*, volume 1957, pages 111–21, 1993.

- [9] Jezekiel Ben-Arie and K. Raghunath Rao. A novel approach for template matching by nonorthogonal image expansion. *IEEE Transactions on Circuits and Systems for Video Technology*, 3(1):71–84, Feb 1993.
- [10] J. Ross Beveridge et al. November 1993 Fort Carson RSTA data collection final report. Technical Report CS-94-118, Department of Computer Science, Colorado State University, Fort Collins, CO 80526, August 1994.
- [11] Bir Bhanu. Automatic target recognition: State of the art survey. *IEEE Transactions on Aerospace and Electronic Systems*, AES-22(4):364–79, Jul 1986.
- [12] Bir Bhanu and Richard D. Holben. Model-based segmentation of FLIR images. *IEEE Transactions on Aerospace and Electronic Systems*, AES-26(1):2–11, Jan 1990.
- [13] Bir Bhanu and Terry L. Jones. Image understanding research for automatic target recognition. *IEEE Aerospace and Electronic Systems Magazine*, 8(10):15–22, Oct 1993.
- [14] Joseph Boriotti and Fereydoun Maali. Automatic thresholding for detection of landing aircraft in flir imagery. In *Proc. SPIE*, volume 1957, pages 156–9, 1993.
- [15] Sudhir Borra and Sudeep Sarkar. A framework for performance characterization of intermediate-level grouping modules. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(11):1306–12, Nov 1997.
- [16] Kevin Bowyer, Christine Kranenburg, and Sean Dougherty. Edge detector evaluation using empirical ROC curves. In *CVPR'99 Proceedings*, pages 354–9, Jun 1999.

- [17] Kim L. Boyer and Sudeep Sarker. “on the localization performance measure and optimal edge detection”. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(1):106–8, Jan 1994.
- [18] Barry W. Brown. RANDLIB.C — library of C routines for random number generation. README file, Aug 1997. <ftp://odin.mdacc.tmc.edu/pub/source/randlib.c-1.3.tar.gz>.
- [19] John F. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6):679–98, Nov 1986.
- [20] Joan F. Cartier et al. Target attractiveness model for field-of-view search. *Optical Engineering*, 37(7):1923–36, Jul 1998.
- [21] George Casella and Roger L. Berger. *Statistical Inference*. Duxbury Press, 1990.
- [22] A. Catlin et al. Performance modeling for automatic target recognition systems. In *Algorithms for Synthetic Aperture Radar Imagery IV*, volume 3070, pages 185–93. SPIE, Apr 1997.
- [23] Lipchen Alex Chan and Nasser M. Nasrabadi. Automatic target recognition using vector quantization and neural networks. *Optical Engineering*, 38(12):2147–61, Dec 1999.
- [24] Kyujin Cho, Peter Meer, and Javier Cabrera. Performance assessment through bootstrap. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(11):1185–98, Nov 1997.
- [25] James J. Clark. Authenticating edges produced by zero-crossing algorithms.

- IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(1):43–57, Jan 1989.
- [26] L. G. Clark and J. V. Velten. Image characterization for automatic target recognition algorithm evaluations. *Optical Engineering*, 30(2):147–53, 1991.
- [27] D. Cyganski, B. King, R. F. Vaz, and J. A. Orr. ROC analysis of ATR from SAR images using a model-based recognizer incorporating pose information. In *Algorithms for Synthetic Aperture Radar Imagery II*, volume 2487, pages 190–200. SPIE, Apr 1995.
- [28] E. R. Davies. The effect of noise on edge orientation computations. *Pattern Recognition Letters*, 6:315–22, Dec 1987.
- [29] Dick de Ridder et al. Vehicle recognition in infrared images using shared weight neural networks. *Optical Engineering*, 37(3):847–57, Mar 1998.
- [30] S. Der, V. Mirelli, D. Nguyen, and L. C. Wang. Analysis of probe methodologies for target recognition. In *Signal Processing, Sensor Fusion, and Target Recognition V*, volume 2755, pages 46–57. SPIE, Apr 1996.
- [31] Sandor Zoltan Der. Automatic target recognition using passive infrared and laser radar sensor. Technical Report CAR-TR-762, Computer Vision Laboratory, Center for Automation Research, University of Maryland, College Park, March 1995.
- [32] Sandor Zoltan Der and Rama Chellappa. Probe-based automatic target recognition in infrared imagery. *IEEE Transactions on Image Processing*, 6(1):92–102, Jan 1997.

- [33] Dov Dori and Robert M. Haralick. A pattern recognition approach to the detection of complex edges. *Pattern Recognition Letters*, 16:517–29, May 1995.
- [34] Ronald G. Driggers, Paul Cox, and Michael Kelley. National imagery interpretation rating system and the probabilities of detection, recognition, and identification. *Optical Engineering*, 36(7):1952–9, Jul 1997.
- [35] S. A. Dudani, K. J. Breeding, and R. B. McGhee. Aircraft identification by moment invariants. *IEEE Transactions on Computers*, 26:39–46, 1977.
- [36] Melanie Dutkiewicz et al. Fusion of optical, infrared and radar imagery for coastal surveillance. In *Proc. SPIE*, volume 2744, pages 165–75, 1996.
- [37] J. Egerváry. Matrixok kombinatoricus tulajonsagairol. *Mat. és Fiz. Lapok*, 38:16–28, 1931. Translation by H. W. Kuhn, “On combinatorial properties of matrices,” George Washington University Logistics Papers 11, 1955.
- [38] James H. Elder and Steven W. Zucker. Local scale control for edge detection and blur estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(7):699–716, Jul 1998.
- [39] Brian Ernisse et al. Complete automatic target cuer/recognition system for tactical forward-looking infrared images. *Optical Engineering*, 36(9):2593–2603, Sep 1997.
- [40] Amir H. Fazlollahi and Bahram Javidi. Optimum receivers for pattern recognition problems with nonoverlapping target and background noise. *Optical Engineering*, 36(10):2633–41, Oct 1997.
- [41] Victoria T. Franques and David A. Kerr. Wavelet-based rotationally invariant

- target classification. In *Signal Processing, Sensor Fusion, and Target Recognition VI*, volume 3068, pages 102–12. SPIE, Apr 1997.
- [42] Noah S. Friedland and Azriel Rosenfeld. An integrated approach to 2D object recognition. *Pattern Recognition*, 30(3):525–35, 1997.
 - [43] Keinosuke Fukunaga. *Statistical pattern recognition*. Academic Press, 2nd edition, 1990.
 - [44] Paul D. Gader et al. Segmentation free shared weight networks for automatic vehicle detection. *Neural Networks*, 8(9):1457–73, 1995.
 - [45] P. H. Gregson. Using angular dispersion of gradient direction for detecting edge ribbons. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(7):682–96, Jul 1993.
 - [46] W. E. L. Grimson and D. P. Huttenlocher. On the sensitivity of the Hough transform for object recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12:255–74, 1990.
 - [47] Lalit Gupta et al. Robust automatic target recognition using a localized boundary representation. *Pattern Recognition*, 28(10):1587–98, Oct 1995.
 - [48] John F. Haddon and James F. Boyce. Texture classification of segmented regions of forward looking infrared images. In *Proc. SPIE*, volume 1957, pages 144–55, 1993.
 - [49] M. K. Hamilton and T. A. Kipp. Model-based multi-sensor automatic target identification for FLIR fused with MMW. In *Automatic Object Recognition V*, volume 2485, pages 286–94. SPIE, Apr 1995.

- [50] M. K. Hamilton and T. A. Kipp. ATR architecture for multisensor fusion. In *Signal Processing, Sensor Fusion, and Target Recognition V*, volume 2755, pages 126–33. SPIE, Apr 1996.
- [51] Robert M. Haralick. Digital step edges from zero crossing of second directional derivatives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6(1):58–68, Jan 1984.
- [52] Robert M. Haralick. A Bayesian approach to robust local facet estimation. In C. R. Smith and G. J. Erickson, editors, *Maximum-Entropy and Bayesian Spectral Analysis and Estimation Problems*, pages 85–97. D. Reidel Publishing Compnay, 1987.
- [53] Robert M. Haralick. Performance assessment of near-perfect machines. *Machine Vision and Applications*, 2:1–16, 1989.
- [54] Robert M. Haralick. Dialogue: Performance characterization in computer vision. *CVGIP: Image Understanding*, 60(2):245–9, Sep 1994.
- [55] Robert M. Haralick. Overview: Computer vision performance characterization. In iuw94 [66], pages 663–666.
- [56] Robert M. Haralick. Performance characterization protocol in computer vision. In iuw94 [66], pages 667–673.
- [57] Robert M. Haralick. Propagating covariance in computer vision. *International Journal of Pattern Recognition and Artificial Intelligence*, 10(5):561–572, Aug 1996.
- [58] Robert M. Haralick, K. Shanmugam, and I. Dinstein. Textural features for

- image classification. *IEEE Transactions on systems, Man, nad Cybernetics*, SMC-3(6):610–21, Nov 1973.
- [59] Robert M. Haralick and Linda G. Shapiro. Survey: Image segmentation techniques. *CVGIP*, 29:100–32, 1985.
 - [60] Robert M. Haralick and Linda G. Shapiro. *Computer and Robot Vision*. Addison-Wesley Publishing Company, Inc., 1992.
 - [61] Robert M. Haralick and Layne Watson. A facet model for image data. *Computer Graphics and Image Processing*, 15:113–29, 1981.
 - [62] Michael D. Health, Sudeep Sarkar, Thomas Sanocki, and Kevin W. Bowyer. A robust visual method for assessing the relative performance of edge-detection algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(12):1338–59, Dec 1997.
 - [63] Jerald A. Herstein et al. Model-based system for vehicle detection and identification. In *Proc. SPIE*, volume 1957, pages 122–43, 1993.
 - [64] M.-K. Hu. Visual pattern recognition by moment variants. *IRE Transactions on Information Theory*, 8:179–87, 1962.
 - [65] Anthony Iannino and Stephen D. Shapiro. An iterative generalization of the Sobel edge detection operator. In *PRIP'79*, pages 130–7, 1979.
 - [66] *1994 Image Understanding Workshop Proceedings*, 1994.
 - [67] Ramesh Jain, R. Kasturi, and Brian G. Schunk. *Machine Vision*. McGraw-Hill, 1995.

- [68] Norman L. Johnson and Samuel Kotz. *Distributions in Statistics – Continuous Univariate Distributions*. John Wiley and Sons, 1994.
- [69] R. M. Jones. The probability distribution for the ratio of variables having a normal distribution. Technical Report NOAA Technical Memorandum ERL WPL-84, Wave Propagation Laboratory, NOAA, Boulder, Colorado, Nov 1981.
- [70] Gopal K. Kanji. *100 statistical tests*. Sage publications Ltd, 1993.
- [71] Tapas Kanungo, Byron Dom, Wayne Niblack, and David Steele. A fast algorithm for MDL-based multi-band image segmentation. In *CVPR'94*, pages 609–16, 1994.
- [72] Tapas Kanungo and Robert M. Haralick. Multivariate hypothesis testing for gaussian data: Theory and software. Technical report, Intelligent Systems Laboratory, Department of Electrical Engineering, University of Washington, Seattle, WA 98195, Oct 1995.
- [73] Tapas Kanungo, M. Y. Jaisimha, J. Palmer, and Robert M. Haralick. A methodology for quantitative performance evaluation of detection algorithms. *IEEE Transactions on Image Processing*, 4(12):1667–74, Dec 1995.
- [74] N. Kiryati, Y. Eldar, and A. M. Bruckstein. A probabilistic Hough transform. *Pattern Recognition*, 24:303–16, 1991.
- [75] L. Kitchen and A Rosenfeld. Edge evaluation using local edge coherence. *IEEE Transactions on Systems, Man, and Cybernetics*, 11(9):597–605, Sep 1981.
- [76] Joseph H. Kitrosser. An evaluation of a region-growing algorithm performing detection for automatic target recognition. *Journal of Imaging Science and Technology*, 38(4):311–20, Jul/Aug 1994.

- [77] Donald Ervin Knuth. *The Stanford GraphBase: a platform for combinatorial computing*. Addison-Wesley, 1993.
- [78] D. König. Graphok és matrixok. *Mat. és Fiz. Lapok*, 38:116–119, 1931.
- [79] A. Kramer, D. Perschbacher, R. Johnston, and T. Kipp. Relational template matching algorithm for FLIR automatic target recognition. In *Architecture, Hardware, and Forward-Looking Infrared Issues in Automatic Target Recognition*, volume 1957 of *Proceedings of the SPIE*, pages 29–37, 1993.
- [80] H. W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2:83–97, 1955.
- [81] A. D. Lanterman, M. I. Miller, and D. L. Snyder. Representations of shape for structural inference in infrared scene. In *Automatic Target Recognition VI*, volume 3069, pages 257–68. SPIE, Apr 1997.
- [82] Aaron D. Lanterman, Joseph A. O’Sullivan, and Michael I. Miller. Kullback-leibler distance for quantifying clutter and models. *Optical Engineering*, 38(12):2134–46, Dec 1999.
- [83] V. F. Leavers. Survey: which Hough transform. *CVGIP: Image Understanding*, 58:250–64, 1993.
- [84] Baoxin Li et al. Experimental evaluation of neural, statistical and model-based approaches to FLIR ATR. In *Proceedings of the SPIE*, volume 3371, pages 388–97, 1998.
- [85] Gang Liu et al. Ground-truthing the vehicles in the vertical-view Ft. Hood images. Technical report, Intelligent Systems Laboratory, Department of Electrical Engineering, University of Washington, Seattle, WA 98195, Dec 1997.

- [86] Gang Liu, Lixin Gong, and Robert M. Haralick. Vehicle detection in aerial imagery and performance evaluation. Submitted to the journal *Computer Vision and Image Understanding*, Sep 1999.
- [87] Gang Liu and Robert M. Haralick. FLIR ATR algorithm using maximum likelihood ratio criterion. Technical report, Intelligent Systems Laboratory, Department of Electrical Engineering, University of Washington, Seattle, WA 98195, May 1996.
- [88] Gang Liu and Robert M. Haralick. Automated construction of templates for matching. In *IUW'97 Proceedings*, pages 1247–1254, 1997.
- [89] Gang Liu and Robert M. Haralick. Using polygon centroid covariance in target recognition. Technical report, Intelligent Systems Laboratory, Department of Electrical Engineering, University of Washington, Seattle, WA 98195, Dec 1997.
- [90] Gang Liu and Robert M. Haralick. Using centroid covariance in target recognition. In *ICPR'98 Proceedings*, pages 1343–46, Brisbane, Australia, August 1998.
- [91] Gang Liu and Robert M. Haralick. Assignment problem in edge detection performance evaluation. In *CVPR 2000 Proceedings*, Jun 2000.
- [92] Gang Liu and Robert M. Haralick. FLIR ATR using location uncertainty. *Journal of Electronic Imaging*, 9(2):178–93, Apr 2000.
- [93] Gang Liu and Robert M. Haralick. Two practical issues in canny's edge detector implementation. In *ICPR 2000 Proceedings*, Jun 2000.
- [94] Gang Liu and Robert M. Haralick. Vehicle ground-truth database for the vertical-view Ft. Hood imagery. In *ICPR 2000 Proceedings*, Sep 2000.

- [95] Gang Liu, Feng Zhuge, Lixin Gong, and Robert M. Haralick. Performance evaluation of the CFAR vehicle detection algorithm. In *1998 Image Understanding Workshop Proceedings*, 1998.
- [96] Edward P. Lyvers and O. Robert Mitchell. Precision edge contrast and orientation estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(6):927–37, Nov 1988.
- [97] Kanti V Mardia. *Statistics of directional data*. Academic Press, 1972.
- [98] David H. Marimont and Yossi Rubner. A probabilistic framework for edge detection and scale selection. In *ICCV'98*, pages 207–14, 1998.
- [99] K. C. Markham. Comparison of segmentation processes for object acquisition in infrared images. *IEE Proceedings*, 136, Pt. F(1):13–21, Feb 1989.
- [100] David C. Marr and E. Hildreth. Theory of edge detection. *Proc. R. Soc. Lond.*, B 207:187–217, 1980.
- [101] Roy M. Matchko and Grant R. Gerhart. Parametric analysis of the blackwell-mccready data. *Optical Engineering*, 37(7):1937–44, Jul 1998.
- [102] Roy M. Matchko and Grant R. Gerhart. Target detection for low ambient backgrounds. *Optical Engineering*, 37(7):1945–50, Jul 1998.
- [103] James R. McManamey. Predicting human observer performance with response equivalent delta-t. *Optical Engineering*, 37(7):1960–8, Jul 1998.
- [104] Thomas J. Meitzler et al. Computing the probability of target detection in dynamic visual scenes containing clutter using fuzzy logic approach. *Optical Engineering*, 37(7):1951–9, Jul 1998.

- [105] Maqsood A. Mohd. Performance characterization and sensitivity analysis of ATR algorithms to scene distortions. In *Proc. SPIE*, volume 1957, pages 203–14, 1993.
- [106] Hiroshi Murase and Shree K. Nayar. Visual learning and recognition of 3-d objects from appearance. *International Journal of Computer Vision*, 14(1):5–24, Jan 1995.
- [107] Vishvjit S. Nalwa and Thomas O. Binford. On detecting edges. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6):699–714, Nov 1986.
- [108] N. Nandhakumar and J. K. Aggarwal. Integrated analysis of thermal and visual images for scene interpretation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(4):469–81, Jul 1988.
- [109] N. Nandhakumar and J. K. Aggarwal. Multisensory computer vision. *Advances in Computers*, 34:34, 1992.
- [110] H. Nasr. ATR performance modeling for building multi-scenario adaptive systems. In *Signal and Image Processing Systems Performance Evaluation, Simulation and Modeling*, volume 1483, pages 112–7. SPIE, Apr 1991.
- [111] Adam R. Nolan and William G. Wee. Automatic FLIR target recognition using a hierarchical neural system. In *Proc. SPIE*, volume 1957, pages 50–9, 1993.
- [112] Alan V. Oppenheim. *Discrete-time signal processing*. Prentice Hall, 1989.
- [113] Athanasios Papoulis. *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill, Inc., 3rd edition, 1991.

- [114] V. Parameswaran, Philippe Burlina, and Rama Chellappa. Performance analysis and learning approaches for vehicle detection and counting in aerial images. In *ICASSP'97*, pages 2753–6, 1997.
- [115] Federico Pedersini, Augusto Sarti, and Stefano Tubaro. Estimation and compensation of subpixel edge localization error. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(11):1278–84, Nov 1997.
- [116] T. Peli et al. Morphology-based detection/discrimination of ground vehicles in terrainboard FLIR imagery. In *Proceedings of the SPIE*, volume 2485, pages 2–9, Orlando, FL, Apr 1995.
- [117] Tamar Peli et al. Morphology-based algorithms for target detection/segmentation in flir imagery. In *Proc. SPIE*, volume 1957, pages 85–94, 1993.
- [118] E. Persoon and K.-S. Fu. Shape discrimination using fourier descriptors. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-7:170–9, 1977.
- [119] Quoc Henry Pham, Timothy M. Brosnan, and Mark J. T. Smith. Sequential digital filters for fast detection of targets in FLIR image data. In *Automatic Target Recognition VII*, volume 3069, pages 62–73. SPIE, Apr 1997.
- [120] A. S. Politopoulos. An algorithm for the extraction of target-like objects in cluttered FLIR imagery. *IEEE Aerospace and Electronic Systems Society Newsletter*, pages 23–37, Nov 1980.
- [121] William H. Press et al. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, 2nd edition, 1992.

- [122] Visvanathan Ramesh. *Performance characterization of image understanding algorithms*. PhD thesis, University of Washington, 1995.
- [123] Visvanathan Ramesh and Robert M. Haralick. An integrated gradient edge detector — theory and performance evaluation. In *IUW'94 Proceedings*, pages 689–95, 1994.
- [124] K. Raghunath Rao and Jezekiel Ben-Arie. Optimal edge detection using expansion matching and restoration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-16(12):1169–82, Dec 1994.
- [125] James A. Ratches et al. Aided and automatic target recognition based upon sensory inputs from image forming systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-19(9):1004–19, Sep 1997.
- [126] Jorma Rissanen. Modeling by shortest data description. *Automatica*, 14:465–71, 1978.
- [127] Jorma Rissanen. A universal prior for integers and estimation by minimum description length. *The Annals of Statistics*, 11(2):416–31, 1983.
- [128] Jorma Rissanen. Minimum-description length principle. In *Encyclopedia of Statistical Sciences*, volume 5, pages 523–7, 1989.
- [129] Jorma Rissanen. Fisher information and stochastic complexity. *IEEE Transactions on Information Theory*, 42(1):40–7, Jan 1996.
- [130] R. Tyrrell Rockafellar. *Network flows and monotropic optimization*. Wiley, 1984.
- [131] S. K. Rogers et al. Neural network for automatic target recognition. *Neural Networks*, 8(7-8):1153–84, 1995.

- [132] Songnian Rong and Bir Bhanu. Modeling clutter and context for target detection in infrared images. In *Proceedings CVPR'96*, pages 106–113, Jun 1996.
- [133] Paul L. Rosin. Edges: saliency measures and automatic thresholding. *Machine Vision and Applications*, 9:139–59, 1997.
- [134] M. W. Roth. Survey of neural network technology for automatic target recognition. *IEEE Transactions on Neural Networks*, 1(1):28–43, Mar 1990.
- [135] S. R. Rotman, G. Tidhar, and M. L. Kowalczyk. Clutter metrics for target detection systems. *IEEE Transactions on Aerospace and Electronic Systems*, 30(1):81–91, Jan 1994.
- [136] F. Sadjadi and M. Bazakos. A perspective on ATR evaluation technology (automatic target recognition). In *Signal and Image Processing Systems Performance Evaluation*, volume 1310, pages 2–15. SPIE, Apr 1990.
- [137] Ashok Samal and Jodi Edwards. Generalized Hough transform for natural shapes. *Pattern Recognition Letters*, 18:473–80, 1997.
- [138] B. J. Schachter. A survey and evaluation of FLIR target detection/segmentation algorithms. In *Proceedings of DARPA Image Understanding Workshop*, pages 40–57, Sep 1982.
- [139] H. C. Schau. Shape recognition with scale and rotation invariance. *Optical Engineering*, 31(2):268–74, Feb 1992.
- [140] David E. Schmieder and Marshall R. Weathersby. Detection performance in clutter with variable resolution. *IEEE Transactions on Aerospace and Electronic Systems*, AES-19(4):622–30, Jul 1983.

- [141] Klamer Schutte. Improving object recognition using range information. In *Proc. SPIE*, volume 3374, pages 317–28, 1998.
- [142] Min C. Shin, Dmitry Goldgof, and Kevin Bowyer. Comparison of edge detectors using an object recognition task. In *CVPR'99 Proceedings*, pages 360–5, Jun 1999.
- [143] M. V. Shirvailkan and M. M. Trivedi. Developing texture-based image clutter measures for object detection. *Optical Engineering*, 31(2):2628–39, Dec 1992.
- [144] Jefferey A. Shufelt. Performance evaluation and analysis of monocular building extraction from aerial imagery. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(4):311–26, Apr 1999.
- [145] Jerzy Siuzdak. A single filter for edge detection. *Pattern Recognition*, 31(11):1681–86, Nov 1998.
- [146] Stephen M. Smith and J. Michael Brady. SUSAN — a new approach to low level image processing. *International Journal on Computer Vision*, 23(1):45–78, 1997.
- [147] Mark Richard Stevens and J. Ross Beveridge. Precise matching of 3-d target models to multisensor data. *IEEE Transactions on Image Processing*, 6(1):126–142, Jan 1997.
- [148] Roni Succary et al. Relative effects of distortion and noise on target acquisition: the advisability of image restoration. *Optical Engineering*, 37(7):1914–22, Jul 1998.
- [149] Hemant D. Tagare and Rui J. P. deFigueiredo. On the localization performance

- measure and optimal edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(12):1186–90, Dec 1990.
- [150] Hemant D. Tagare and Rui J. P. deFigueiredo. Reply to “on the localization performance measure and optimal edge detection”. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(1):108–10, Jan 1994.
- [151] Barnabás Takács and Lev Sadovnik. Three-dimensional target recognition and tracking using neural networks trained on optimal views. *Optical Engineering*, 37(3):819–28, Mar 1998.
- [152] Quang Minh Tieng and W. W. Boles. Recognition of 2d object contours using the wavelet transform zero-crossing representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(8):910–6, Aug 1997.
- [153] Du-Ming Tsai. An improved generalized Hough transform for the recognition of overlapping objects. *Image and Vision Computing*, 15:877–88, 1997.
- [154] Michihiro Uenohara and Takeo Kanade. Use of Fourier and Karhunen-Loeve decomposition for fast pattern matching with a large set of templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-19(8):891–8, Aug 1997.
- [155] Jacques G. Verly, Richard L. Delanoy, and Dan E. Dudgeon. Model-based system for automatic target recognition from forward-looking laser-radar imagery. *Optical Engineering*, 31(12):2540–52, Dec 1992.
- [156] Lin-Cheng Wang, Sandor Z. Der, and Nasser M. Nasrabadi. Composite classifiers for automatic target recognition. *Optical Engineering*, 37(3):858–68, Mar 1998.

- [157] J. S. Weszka and A. Rosenfeld. Threshold evaluation techniques. *IEEE Transactions on systems, Man, nad Cybernetics*, SMC-8:622–9, 1978.
- [158] Xing Wu and Bir Bhanu. Gabor wavelet representation for 3-d object recognition. *IEEE Transactions on Image Processing*, 6(1):47–64, Jan 1997.
- [159] Jianping Yao and Guy Lebreton. Invariant-pattern recognition: smart algorithms for segmented two-dimensional patterns with an axis of symmetry. *Optical Engineering*, 38(12):2022–8, Dec 1999.
- [160] Seungku Yi, Robert M. Haralick, and Linda G. Shapiro. Error propagation in machine vision. *Machine Vision and Applications*, 7:93–114, 1994.
- [161] Oh Young and M. Cook. Analytical modeling for automatic target recognition performance evaluation. In *Signal and Image Processing Systems Performance Evaluation*, volume 1310, pages 162–73. SPIE, Apr 1990.
- [162] Y.T. Zhou and T. Gutschow. A highly accurate IR automatic object recognition system. In *Proc. SPIE*, volume 1957, pages 60–7, 1993.
- [163] Oscar A. Zuniga and Robert M. Haralick. Integrated directional derivative gradient operator. *IEEE Transactions on Systems, Man and Cybernetics*, SMC-17(3):508–17, May-Jun 1987.

Appendix A

FACET MODEL ORDER DETERMINATION BY MDL PRINCIPLE

The facet model [61, 52, 51, 60] uses bivariate polynomials to model the image intensity surface. It is a very important tool in low-level vision. It has been customary to use the cubic facet model, i.e., bivariate polynomials of up to the third order, in small neighborhoods such as 5×5 and 7×7 . The particular choice of the highest order of 3 is roughly a rule-of-thumb.

The minimum description length (MDL) principle [126, 127, 128, 129] is a well established principle for data modeling. Here we summarize some issues in applying this principle in facet parameter estimation to adaptively determine the optimal order of polynomial fit for the local neighborhood.

A.1 *The MDL principle*

According to the MDL principle, the description of a particular observed data vector x using a particular model with the parameter vector θ is broken into two parts: the model part and the deviation part. The total coding length is expressed as

$$L(x, \theta) = L(\theta) + L(x|\theta) \quad (\text{A.1})$$

where $L(\theta)$ is the number of bits required to describe the model and $L(x|\theta)$ is that for describing the deviation of the data vector from the described model. There is an intrinsic trade-off between the model complexity and the complexity of the data given the model. The objective is to reach a compromise with the smallest sum.

A.1.1 Encoding the data using a model

For computing $L(x|\theta)$, it is very popular to use Shannon's theory which states that the expected shortest coding length is

$$L(x|\theta) = -\log_2 p(y),$$

where $y = x - \hat{x}(\hat{\theta})$ is the difference vector between x and the estimated values \hat{x} given by the chosen model $\hat{\theta}$. $p(\cdot)$ is the probability of y . Notice that this statement assumes that y has a discrete distribution. In many applications, continuous models are involved and have to be quantized in order to use this result.

It has been pointed out that Shannon's formula is asymptotic for large samples. An alternative formula is given by Rissanen [129] for small sample sizes

$$L(x|\theta) = -\log p(x|\hat{\theta}(x)) + \frac{k}{2} \log \frac{n}{2\pi} + \log \int_{\Gamma} \sqrt{|I(\theta)|} d\theta + o(1), \quad (\text{A.2})$$

where k is the number of real-valued parameters in $\hat{\theta}(x)$, n is the number of elements in the sample x , Γ is the region of allowable θ values, and $I(\theta)$ is the *Fisher information matrix*,

$$I(\theta) = -\mathbf{E}_{\theta} \left\{ \frac{\partial \log p(x|\theta)}{\partial \theta_i \partial \theta_j} \right\}.$$

We have not yet closely studied the use of this formula. In the rest of this document, in discussing $L(x|\theta)$, we restrict our attention to Shannon's formula.

Elements in y are commonly assumed to have the iid normal distribution. In quantizing it into a discrete distribution, we use a quantization interval δ which is a very important parameter in applying the MDL principle. It significantly affects the relative portion of $L(\theta)$ and $L(x|\theta)$ in the final minimized $L(x, \theta)$. δ is usually specified by considerations for specific applications. We then have

$$p(y) = \prod_{i=1}^N \left(\Phi \left(\frac{y + \frac{\delta}{2}}{\sigma} \right) - \Phi \left(\frac{y - \frac{\delta}{2}}{\sigma} \right) \right), \quad (\text{A.3})$$

where N is the dimension of the data vector, σ is the standard deviation of the distribution for y , $\Phi(\cdot)$ is the cumulative distribution function (cdf) of the standard

normal distribution. σ needs to be estimated from the data, usually by using the residual fitting error.

Let K denote the number of facet parameters in θ , and let ξ denote the total residual fitting error (computed as the sum of the squared difference between x and the fitted values \hat{x} specified by the K facet parameters.) The noise standard deviation σ is estimated as

$$\hat{\sigma} = \sqrt{\frac{\xi}{N-K}}.$$

Substituting into $p(y)$, we have

$$L(x|\theta) = -\log_2 p(y) = -\sum_{i=1}^N \log_2 \left(\Phi \left(\frac{y + \frac{\delta}{2}}{\sqrt{\frac{\xi}{N-K}}} \right) - \Phi \left(\frac{y - \frac{\delta}{2}}{\sqrt{\frac{\xi}{N-K}}} \right) \right). \quad (\text{A.4})$$

A.1.2 Encoding the model parameters

To transmit the model parameters θ with finite coding length, θ has to have finite precision. In most applications, many elements in θ are real-valued, hence need to be quantized, too. Let K denote the number of elements in θ , and ϵ_k denote the quantization interval for the k -th element. Rissanen [126, 127] derived how ϵ_k should be chosen to give the smallest total coding length, which results in the so-called universal prior for the integers. For large sample size N , real-valued parameters are quantized to \sqrt{N} quantization levels. Here, N is not very large for small image neighborhoods, e.g., $N = 25$ in the 5×5 neighborhood. Therefore, we take an alternative approach and use a particular coding scheme for the facet parameters.

When the facet parameters θ are quantized, the corresponding facet polynomial will have error due to this quantization. We choose the value for ϵ_k to be such that the maximum error due to the quantization in the value of the facet polynomial does not exceed δ , the quantization interval for the observation.

$$\epsilon_k = \frac{\delta}{\max_{i,j} |b_{ij}^{(k)}|} \quad (\text{A.5})$$

where $b_{ij}^{(k)}$ is the value of the (i, j) element of the k -th 2-D DOP facet basis [60].

Let the 2-D DOP facet parameters be denoted by the vector α . If we base our description of the data on the first K facet parameters, the code length for the parameter part is computed as

$$L(\theta) = 2K + \sum_{k=1}^K \log_2^0 \frac{|\alpha_k|}{\epsilon_k} \quad (\text{A.6})$$

where $\log_2^0(x)$ is defined for $x \geq 1$ to be $\log_2(2.865064) + \log_2(x) + \log_2(\log_2(x)) + \dots$ up to all positive terms [127, 71]. The term $2K$ appears as a result of using one bit for the sign and one bit for the zero/non-zero flag for each α_k .

The encoding of K itself makes use of the particular nature of the facet model. For the facet model orders of 0 through 4, the values for K are 1, 3, 6, 10, 15, respectively. We use a constant length of 3 bits to encode the facet model order, which is the same for all possible orders. Hence this term is dropped from the code length expression.

A.2 Facet model order determination

In this discussion, we restrict our interest to facet model orders 0 through 4. The method developed, however, is applicable to any order up to the highest possible.

For a given data vector, we first apply the DOP analysis and obtain the DOP coefficients α . For a particular model order $d \in \{0, 1, 2, 3, 4\}$, the first K ($= 1, 3, 6, 10, 15$) parameters are encoded with a code length specified by Equation A.6. The residual fitting error ξ from using this model order is computed and plugged in Equation A.4. The total code length is the sum of the two parts. This total code length is computed for every d value of interest. The d value yielding the smallest code length is the optimal model order as specified the MDL principle. The signal component of the observation is specified by the first K (corresponding to this chosen d) DOP coefficients. The remaining DOP coefficients correspond to the noise component.

In the procedure just described, the relationship between d and K is based on the assumption that all facet parameters up to the specified order are of interest. However, this assumption might not be necessary for special applications where only a subset

of the facet parameters up to the specified order are of interest. In that situation, a different relationship between d and K needs to be specified. Corresponding values for K will be smaller than the ones given here. This is one possible direction for improving the current procedure. In our experiments, we observe the reluctance of the procedure to choose high model orders, possibly due to the increasingly larger increment in the number of model parameters when the model order is increased.

A.3 Experiment

In our FLIR ATR application, the SNR at each pixel location needs to be computed. Classic cubic facet analysis tends to let too much noise energy leak into the signal subspace and also over estimate noise variance at the places of high contrast. The result is that the SNR at high contrast places are often erroneously estimated to be very small, and that at noisy background locations are often erroneously estimated to be quite large. By using the MDL principle to adaptively choose the order of facet fit, we should be able to obtain more accurate SNR estimate.

Figure A.1(a) shows one simulated FLIR scene used in our FLIR ATR algorithm development. The gradient magnitude, noise standard deviation and the SNR estimates based on the cubic facet model are shown in Figures A.1(b), A.1(c) and A.1(d), respectively. The SNR estimate is proportional to the ratio of the gradient magnitude and noise standard deviation estimates. In these plots, larger values are shown in darker tones.

From these images we observe that, although the gradient magnitude estimate is quite high along the target boundaries, the SNR estimate is low due to the over estimate of noise at those places. On the other hand, in the background regions, the cubic model accounted for quite significant portions of the noise and clutter, resulting in under estimate of the noise and over estimate of the signal. This causes many high SNR estimates in the background regions. This example shows the inadequacy of

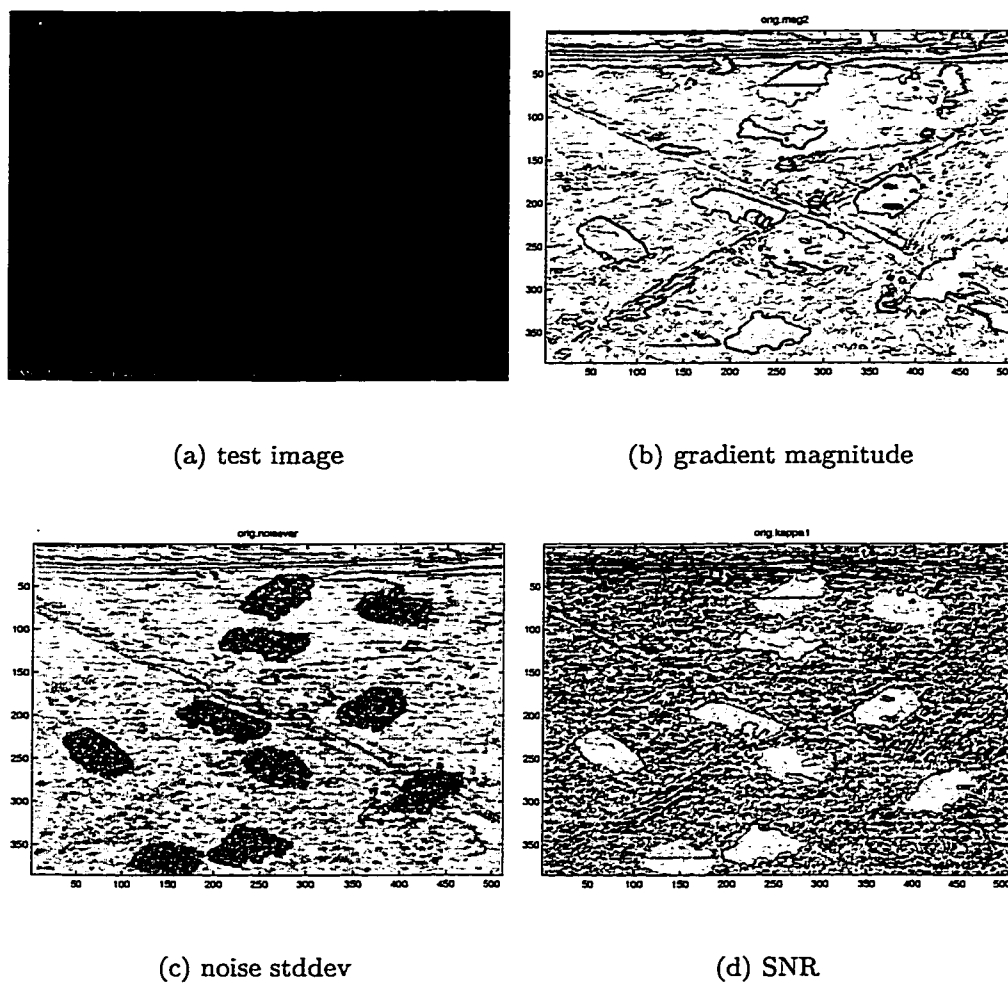


Figure A.1: Test image and estimates made by the cubic facet model.

using a fixed facet model order for the entire image.

The situation can be improved significantly by applying the MDL-based adaptive model order determination idea described earlier. Figure A.2 shows the gradient magnitude, noise standard deviation and SNR estimates from using that idea with the observation quantization interval of $\delta = 30$. The zero-th order has been adaptively chosen for much of the background regions, fixing the previous problem of over estimating the signal and under estimating the noise in those regions. For most of the target boundaries, the first and second orders are chosen. This still has the problem of under estimating the signal and over estimating the noise, the latter being evident in the image of the noise standard deviation estimates. However, the noise estimate is more homogeneous over the entire image, which is intuitively correct. The combined effect of the adaptive order determination scheme in the target boundary and background regions gives a very significantly improved SNR estimate.

We note that the model order determined by the MDL-based procedure relies heavily on the value used for the quantization interval δ , which is the only tuning parameter in the procedure. This parameter allows the procedure to be tuned for different application scenarios according to the particular signal-to-noise ratio of the input. There has not been much in-depth study of the automatic selection of this parameter in the MDL setting. It is usually determined on training data by trial-and-error, which is how we chose the value 30 in our experiment. Further investigation is warranted to develop a systematic method for determining δ .

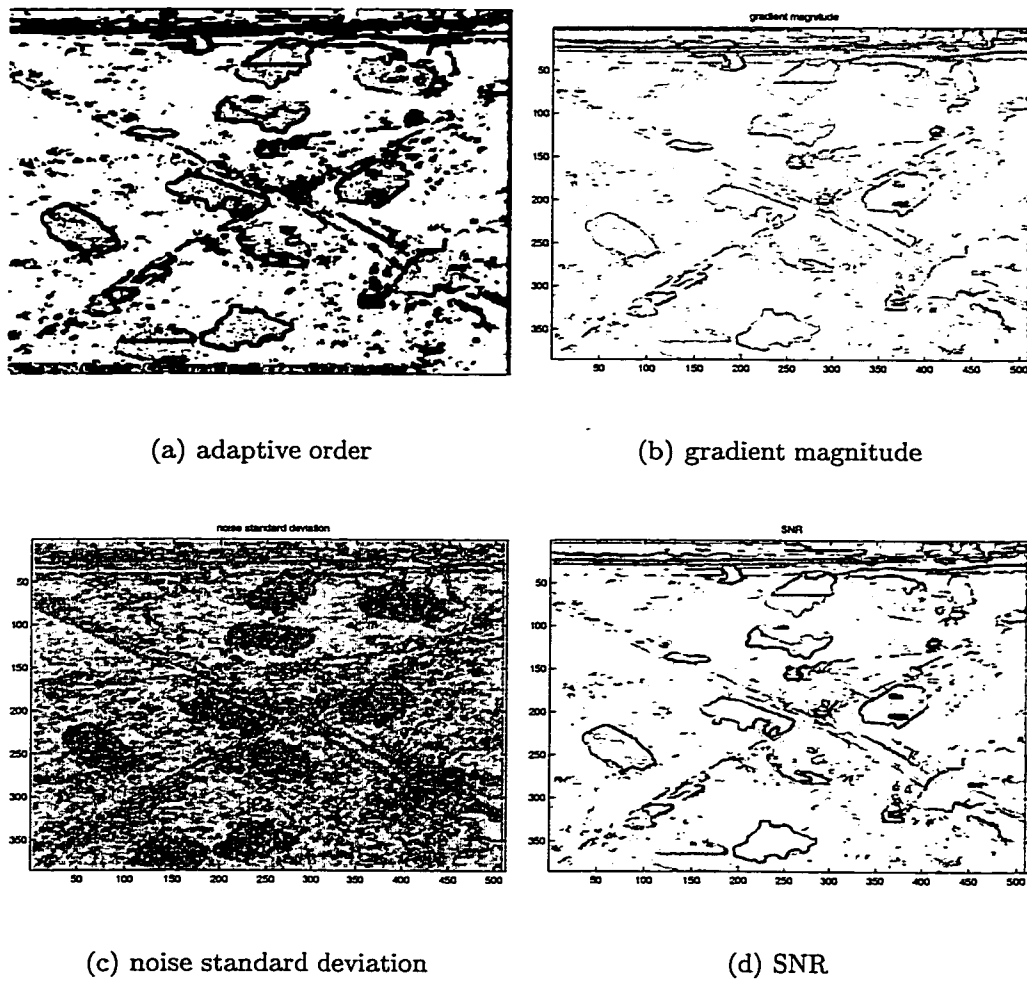


Figure A.2: The model order adaptively determined by the MDL-based procedure and the estimates made by the facet model with that order. For the model order, zero-th order is shown in white and non-zero orders are shown in black.

VITA

Gang Liu was born in Xi'an, China. He received his bachelor's and master's degrees from the Information Engineering Department at Xidian University in Xi'an, China and the Electronics Engineering Department at Tsinghua University in Beijing, China, in 1992 and 1995, respectively. His research interests include computer vision, pattern recognition, image processing, information retrieval and telecommunications. Within computer vision, he is particularly interested in early vision tasks such as gradient estimation, edge/corner/curve detection, segmentation, and performance evaluation.