

MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963 A

AD-A189 041

4

DTIC FILE COPY

**A Probabilistic Approach
to Low-Level Vision**

David B. Sher
Department of Computer Science
The University of Rochester
Rochester, NY 14627

TR 232
October 1987

DTIC
ELECTE
JAN 21 1988

DTIC
ELECTE
JAN 21 1988
S D

Rochester

Department of Computer Science
University of Rochester
Rochester, New York 14627

DISTRIBUTION STATEMENT A
Approved for public release
Distribution Unlimited

88 1 14 082

4

A Probabilistic Approach to Low-Level Vision

David B. Sher
Department of Computer Science
The University of Rochester
Rochester, NY 14627

TR 232
October 1987

DTIC
SELECTED
JAN 21 1988
D

The research reported herein was submitted in partial fulfillment of the requirements of the degree Doctor of Philosophy, and was supervised by Christopher M. Brown of the Department of Computer Science.

This work was supported in part by the National Science Foundation Fellowship SPE-8350104, National Science Foundation Grants MCS-8203028 and RCD-8450125, Defense Advanced Research Projects Agency Grants N00014-78-C-0164 and N00014-82-K-0193, the U.S. Army Engineering Topographic Labs under Grant DACA76-85-C-0001, and the Air Force Systems Command, Rome Air Development Center and the Air Force Office of Scientific Research under Contract F30602-85-C-0008, which supports the Northeast Artificial Intelligence Consortium (NAIC). The Xerox Corporation University Grants Program provided the equipment used in preparing the paper.

DTIC
COPY
INSPECTED
6

DISTRIBUTION STATEMENT A.
Approved for public release
Distribution Unlimited

Security Codes	
Dist	Approved for Release
A-1	

A189 041

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER TR 232	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) A Probabilistic Approach to Low-Level Vision	5. TYPE OF REPORT & PERIOD COVERED technical report	
	6. PERFORMING ORG. REPORT NUMBER	
7. AUTHOR(s) David B. Sher	8. CONTRACT OR GRANT NUMBER(s) N00014-78-C-0164 N00014-82-K-0193 DACA76-85-C-0001	
9. PERFORMING ORGANIZATION NAME AND ADDRESS Computer Science Dept. 535 Computer Studies Bldg. Univ. of Rochester, Rochester, NY 14627	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS	
11. CONTROLLING OFFICE NAME AND ADDRESS DARPA 1400 Wilson Blvd. Arlington, VA 22209	12. REPORT DATE October 1987	
	13. NUMBER OF PAGES 205	
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Office of Naval Research Information Systems Arlington, VA 22217	15. SECURITY CLASS. (of this report) unclassified	
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE	
16. DISTRIBUTION STATEMENT (of this Report) Distribution of this document is unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) optimal feature detection probabilistic algorithms edge detection information fusion Bayesian methods		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) A probabilistic approach to low-level vision algorithms results in algorithms that are easy to tune for a particular application and modules that can be used for many applications. Several routines that return likelihoods can be combined into a single more robust routine. Thus it is easy to construct specialized yet robust low-level vision systems out of algorithms that calculate likelihoods. This dissertation studies algorithms that generate and use likelihoods.		

20. ABSTRACT (Continued)

Probabilities derive from likelihoods using Bayes' rule. Thus vision algorithms that return likelihoods also generate probabilities. Likelihoods are used by Markov Random Field algorithms.

This approach yields facet model boundary pixel detectors that return likelihoods. Experiments show that the detectors designed for the step edge model are on par with the best edge detectors reported in the literature. Algorithms are presented here that use the generalized Hough transform to calculate likelihoods for object recognition.

Evidence, represented as likelihoods, from several detectors that view the same data with different models are combined here. The likelihoods that result are used to build robust detectors out of several specialized ones. Results are shown here for combining boundary detectors that assume several levels of noise and combining detectors of several sizes.

The gains in clarity of design, flexibility of use, and the robustness of the resulting algorithms justify a probabilistic approach to low-level vision problems.

Curriculum Vitae

1. Personal

US Citizen Born 1/14/61 No Disabilities

2. Degrees

7/87 Completed Requirements for Ph.D. in Computer Science from University of Rochester. Thesis entitled "A Probabilistic Approach to Computer Vision."

1/84 Masters in Computer Science, University of Rochester

5/82 B.S. in Computer Science and Mathematics, Yale University, magna cum laude

3. Memberships in Academic Societies

Member

 Tau Beta Pi Engineering Society

Student Member

 The Institute of Electrical and Electronics Engineers Membership Number:
 2744290

Student Member

 IEEE Computer Society

4. Fellowship

Sept 1982 - May 1985

NSF Graduate Fellowship

5. Summary of Work

On June 1, 1982 David Sher started as a student employee of the University of Rochester working under the direction of Christopher Brown. Primarily he studied implementations of the Hough transformation with limited memory resources (Brown, Curtis, and Sher 1983b) (Brown and Sher 1982b). This work used a computer generated data source. He also constructed a flexible graphics package for studying rigid body motion that summer.

In September 1982 David Sher matriculated as a full time graduate student in the Computer Science Department of the University of Rochester. He continued studying the Hough transformation under the direction of Christopher Brown (Brown and Sher 1982a) (Brown, Curtis, and Sher 1983a). In this research the Hough transformation with limited memory was applied to aerial imagery.

In January 1983 David Sher accepted a teaching assistantship for CS220 Data Structures, working under Carla Ellis. During this term he took a course on VLSI design, taught by Gershon Kedem. As a final project for this course he worked with Avadis Tevanian on an early version of the vote tallying chip. He spent the summer of 1983 debugging, testing and fabricating the vote tallying chip (Sher and Tevanian 1984b) (Sher and Tevanian 1984a) under the direction of Gershon Kedem.

In his third semester as a graduate student (September-December 1983) David Sher worked as a teaching assistant for VLSI design under the direction of Gershon Kedem. In his fourth semester he was a research assistant under Gershon Kedem

studying custom hardware for artificial intelligence.

The author spent the summer of 1984 at Carnegie-Mellon University working for the Computer Science Department under the joint direction of H. T. Kung, Jon Webb, and Takeo Kanade. He worked on the WARP project coordinating the hardware and software design projects.

He spent his fifth semester at the University of Rochester studying the implementation of vision algorithms on MIMD multiprocessors (Sher 1985b) under the direction of Christopher Brown. In the rest of his time at the University of Rochester he was working as a research assistant for Christopher Brown.

In his sixth semester (January-May 1985) David Sher began studying probabilities as a universal language for vision algorithm output. As an outgrowth of this research David Sher constructed his first primitive likelihood generators and an early version of the evidence theory in Chapter 5 (Sher 1986d) (Sher 1985a). In this semester he proposed a study of probabilistic approaches to low-level vision as his thesis topic. At that time his thesis committee adjourned.

At the end of his seventh semester his thesis proposal was accepted by his committee. In that time he refined and extended his evidence combination rule into its present form. He also began to compile references on the relationship between probability theory and computer vision.

In his eighth semester (January-May 1986) David Sher developed and implemented efficient algorithms to compute likelihoods for boundary detection on one dimensional data. (Sher 1986b) (Sher 1986a). To explain a deficiency in previous probabilistic approaches to vision he also studied the errors resulting from inappropriate use of estimation techniques (Sher 1986c).

In the summer of 1986 David Sher developed a software package to speed implementation of advanced techniques in likelihood generation and programs to test

these techniques (Sher and Inwegen 1987). In the succeeding semester (September-December 1986) David Sher used this package to implement likelihood generators for two dimensional data. He also supervised an undergraduate, Myra Van Inwegen, in the development of a test image generation package using the software written the previous summer. Using this software he tested his theories about boundary detection and evidence combination with two dimensional data (Sher 1987a) (Sher 1987b).

In his last semester as a graduate student at the University of Rochester, David Sher applied his probabilistic approach to the generalized Hough transformation. He derived the results presented in chapter 4 in January and February of that year. He also worked out the rule for combining operators with different sizes described in section 5.7. This work completed the set of theoretical results required for a dissertation. At this point he began construction of the final dissertation document and the final set of experiments presented here.

At the beginning of July 1987 David Sher registered his dissertation with the University of Rochester. At the end of July he successfully defended his thesis.

Acknowledgements

So many people have contributed to my dissertation that it is impossible to thank them all. Thus this acknowledgement section is something of an approximation.

The primary responsibility for shepherding me through a graduate career has fallen on the capable shoulders of Chris Brown. Without his help my whole graduate career would have been very different and much less successful. I can hardly thank him enough for the help he has given me these 5 years.

Jerry Feldman helped me early in my career by giving me a model for a successful research program with his connectionist project. On my committee, he understood what I was doing and helped me explain it. His comments were always insightful.

Henry Kyburg's class interested me in issues of probability theory and evidence combination. His suggestions have helped me make this dissertation more accessible.

Jack Hall's class fine tuned my ability to apply probability. He also has helped me improve my presentation, and he forced me to be precise. I can not imagine a more capable thesis committee than mine.

When choosing my thesis topic discussions with Stuart Friedberg, Mike Swain, and Cesar Quiroz helped me focus and isolate areas of interest. Throughout the development of my thesis they have been supportive and good friends.

Paul Chou has been helpful in a multitude of ways from listening to my practice talks to following up on my work. He is now using the likelihood generators developed for this dissertation to study Markov random fields.

Myra Van Inwegen showed up at an opportune time to generate graphics to test my work. She developed a beautiful graphics package for me and was a model employee. Without her help testing detectors would have been much harder.

Many thanks to Paul Cooper for buiding the software that allowed me to print this dissertation, and for many interesting discussions.

I must thank Robert Haralick for sending me commentary on boundary pixel detection. This commentary helped make my boundary pixel detectors more flexible. Without Vic Nalwa's courtesy and source code it would have been difficult to do the comparison tests in this thesis. Also Glen Shafer of Dempster-Shafer fame inspired my work on maximum entropy with a comment at a conference.

Without financial support I would not have had time to follow this line of research. In the early part of my graduate career I was supported by the national science foundation fellowship SPE-8350104, and NSF grants MCS-8203028, and RCD-8450125 and by DARPA grants N00014-78-C-0164, and N00014-82-K-0193. For the last two years the Air Force Systems Command, Rome Air Development Center, Griffiss Air Force Base, New York 13441-5700, and the Air Force Office of Scientific Research, Bolling AFB, DC 20332 has supported this research under Contract No. F30602-85-C-0008 and the Defence Advanced Research Products Agency U. S. Army Engineering Topographic Labs has supported this research under grant number DACA76-85-C-0001.

I have received moral support from my friend Carlos Calderon whose face is used in this dissertation. I was inspired by the beauty and intellect of Laura Sanchis, Elizabeth Hinkelman, and Lucy Lin, graduate students at the University of Rochester.

Without continual support from my parents and my sister the constant pressure of graduate school would have been difficult to bear. They nurtured my intellect. All my successes are a result of their care (failures are my fault).

Finally without God almighty I would not exist, without God's love I could not succeed.

ABSTRACT

A probabilistic approach to low-level vision algorithms results in algorithms that are easy to tune for a particular application and modules that can be used for many applications. Several routines that return likelihoods can be combined into a single more robust routine. Thus it is easy to construct specialized yet robust low-level vision systems out of algorithms that calculate likelihoods. This dissertation studies algorithms that generate and use likelihoods.

Probabilities derive from likelihoods using Bayes' rule. Thus vision algorithms that return likelihoods also generate probabilities. Likelihoods are used by Markov Random Field algorithms.

This approach yields facet model boundary pixel detectors that return likelihoods. Experiments show that the detectors designed for the step edge model are on par with the best edge detectors reported in the literature. Algorithms are presented here that use the generalized Hough transform to calculate likelihoods for object recognition.

Evidence, represented as likelihoods, from several detectors that view the same data with different models are combined here. The likelihoods that result are used to build robust detectors out of several specialized ones. Results are shown here for combining boundary detectors that assume several levels of noise and combining detectors of several sizes.

The gains in clarity of design, flexibility of use, and the robustness of the resulting algorithms justify a probabilistic approach to low-level vision problems.

Preface

This thesis is designed on the assumption that it be read through in entirety and in order. However several of the sections can be read separately.

Chapter 1 describes the motivation for this work and lays some of the groundwork for it. Chapter 2 describes work on building domain models for low-level vision problems. A domain model is characterized as the interface between the algorithm designer and an algorithm user. This section addresses the problem of deriving priors from an intuitive model.

Chapter 3 describes the construction of limited support boundary pixel detectors that yield likelihoods and probabilities. Efficient algorithms for a class of boundary detectors are derived here and experiments testing such detectors are in this chapter. Chapter 3 often refers back to the work in chapter 2. Some of the work in chapter 3 has been published separately from the work in chapter 2 (Sher 1986b) (Sher 1987a) and most of it stands on its own.

Chapter 4 describes probabilistic feature detection algorithms for the domain of object recognition. Objects are modeled by templates here. A class of algorithms using the Hough transformation are developed that yield probabilities for the presence of objects. Occlusion of multiple objects in a scene is modeled with a Markov random field. The work in this chapter also can stand on its own and will be published separately when experiments testing these algorithms are done. Thus efficient feature detectors that return probabilities exist for boundary pixel detection

and object recognition. This chapter refers to chapter 2 but does not directly depend on chapter 3 and can be read without it.

Chapter 5 describes merging several specialized feature detectors to yield a robust feature detector. The feature detectors from chapters 3 and 4 can be combined using this theory. Experiments have been done using the boundary detectors from chapter 3 to combine detectors that assume various levels of noise and to combine detectors that use large and small support¹. An early version of the evidence combination work appeared in (Sher 1986d) and (Sher 1985a). More advanced work including some experiments has appeared in (Sher 1987b). Chapter 5 depends greatly on chapters 1, and 2 for definitions. The priors derived in sections 2.3 and 3.1 are useful here.

Chapter 6 gives an overview of the previous work that is relevant to this thesis.

Because this paper straddles computer vision and decision theory much terminology must be introduced that is *unfamiliar to at least part of my audience*. If more terminology than is necessary has been introduced, I would appreciate having my kind readers point out any possible improvements. At the end of this paper in appendix A is a glossary. It is provided to help keep track of the terminology. At the beginning of the paper is an index of defined terms that shows where in the paper each term is defined. These tools should help the reader handle the mass of indigestible terms.

¹Boundary detectors that look at large and small windows.

Table of Contents

Chapter 1. : Motivation	1
1. Probabilities	1
2. Decision Theory	4
3. Likelihoods	6
Chapter 2. : Models	10
1. Primitive and Required Statistics	11
2. Simplifications	12
3. Derivation of Priors	15
Chapter 3. : Boundary Detection	20
1. Models for Boundary Detection	21
1.1. Priors for Boundary Pixel Detection	21
1.2. The Facet Model	22
1.3. Linear Boundaries	25
2. Likelihood Generator	26
3. Efficient Computation of Likelihoods	30
4. The Flexibility of the Likelihood Generation Algorithm	32
5. Experiments	34
5.1. Results with Artificial Images	36
5.1.1. Angle and Orientation	41
5.1.2. Sensitivity to Noise	43
5.2. Results with Laboratory Images	58
5.3. Results with Uncontrolled Images	61
5.4. Comparisons with Established Techniques	64
5.4.1. Results For Artificial Images	65
5.4.2. Results For Laboratory Images and Real Images	70
Chapter 4. : Template Matching	75
1. Models for Template Matching	77

1.1. Independence Assumption	80
1.2. Parameters of the Model	81
2. Likelihoods for Objects	82
3. Thresholds	85
3.1. Thresholding the Matcher	85
3.2. Thresholding the Detector	87
4. Multiple Label Feature Detectors	88
5. Feature Detectors that Return Probabilities	90
6. Modeling Multiple Objects	92
7. Proposed Experiment	95
Chapter 5. : Evidence Combination	98
1. Desiderata for Evidence Combination	99
2. Combining Priors	100
3. Combining Likelihoods	101
4. A Simple Example of Evidence Combination	103
5. Maximum Entropy	104
6. Maximum Entropy and Markov Random Fields	106
7. Operators with Several Window Sizes	108
8. Experimental Results	109
8.1. Experiments Combining Noise Levels	109
8.1.1. Combining Noise with Artificial Images	109
8.1.2. Combining Noise with Camera Images	115
8.2. Experiments Combining Sizes	123
8.2.1. Combining Window Sizes with Artificial Images	124
8.2.2. Combining Window Sizes with Camera Images	133
Chapter 6. : Previous Work	138
1. Previous Work on Modeling	138
2. Previous Work on Edge Detection	140
3. Previous Work on Template Matching	144
4. Previous Work on Decision Theory and Artificial Intelligence	145
5. Previous Work on Probabilistic Relaxation	146
Chapter 7. : Final Thoughts	149
1. Summary	149
2. Future Directions	153
Appendix A. : Glossary	171

Curriculum Vitae	ii
1. Personal	ii
2. Degrees	ii
3. Memberships in Academic Societies	ii
4. Fellowship	iii
5. Summary of Work	iii
Acknowledgements	vi
ABSTRACT	ix
Preface	xi
Table of Contents	xiii
List of Figures and Tables	xvi
Index of Defined Terms	xx

List of Figures and Tables

1 : Slice through an image with an ambiguous boundary	2
2 : Slice through an image with an edge strength of 0	3
3 : Slice through an image with a high edge strength	3
4 : Situations using a Boundary Detector	5
5 : Definition of Image and Scene	10
6 : A Designer Confronted with a Complex User Model	13
7 : Example of Simplified Model	16
8 : Small Window on an Image	18
9 : 3 objects (a, b, c) in one window	18
10 : Example of Digitized Area and Perimeter	22
11 : Example of occlusion	23
12 : Relationship of Surface Normals to Boundaries	25
13 : Basis	28
14 : Two Templates for the Same Edge	31
15 : Surface Normals Change Quickly Near Boundaries	34
16 : 4 Orientations for Boundaries	35
17 : Two Circle Image	37
18 : Circles Image	38
19 : Rectangles Image	39
20 : Combination of Circles and Rectangles Image	40
21 : Oriented response for likelihood generator	42
22 : Oriented response for likelihood generator	43
23 : $\sigma=12$ operators applied to images with too little ($\sigma<12$) noise	46
24 : $\sigma=12$ operators applied to image with correct ($\sigma=12$) amount of noise	47
25 : $\sigma=12$ 5x5 operator applied to images with too much ($\sigma>12$) noise	50
26 : False Positive Rate Charts for Operators Applied to Artificial Images	51

27 : False Negative Rate Charts for Operators Applied to Artificial Images	52
28 : Total Error Rate Charts for Operators Applied to Circles Image	54
29 : Total Error Rate Charts for Operators Applied to Rectangles Image	55
30 : Total Error Rate Charts for Operators Applied to Combined Image	56
31 : Total Error Rate Charts for Operators Applied to Artificial Images	57
32 : Application of Likelihood Generators to Play Doh™ Blobs	60
33 : Application of Likelihood Generators to Play Doh™ Gnome	61
34 : Image of Carlos Calderon	62
35 : Application of Likelihood Generators to Carlos' ear	63
36 : Application of Likelihood Generators to the Sewage Treatment Plant	64
37 : Application of Sobel to Artificial Images	66
38 : Application of Sobel to Artificial Images	66
39 : False Positive Rate Comparison Chart	67
40 : False Negative Rate Comparison Chart	68
41 : Total Error Rate Comparison Chart	69
42 : Application of the Sobel to the Play Doh™ Blob Image	70
43 : Application of the Sobel to the Play Doh™ Gnome Image	70
44 : Application of Nalwa's Operator to the Play Doh™ Blob Image	71
45 : Application of Nalwa's Operator to the Play Doh™ Gnome Image	71
46 : Application of the Sobel to Carlos' ear	72
47 : Application of the Sobel to the Sewage Treatment Plant	73
48 : Application of Nalwa's Operator to the Carlos' ear	73
49 : Application of Nalwa's Operator to the Sewage Treatment Plant	74
50 : Square Template	75
51 : Output of Feature Detector	76
52 : Weighted Template	77
53 : Errors in Input to a Template Matcher	79
54 : Two adjacent pixels	80
55 : Computed Template	85
56 : Template with many Labels	89
57 : Output of Multiple Label Feature Detector	89
58 : Correspondence of Labels to Rows	90
59 : Translation of Third Column of figure 57	90
50 : Square Template (Reprised)	91

60 : Output of a Probabilistic Feature Detector	91
61 : Output of a Probabilistic Feature Detector	92
62 : Small Square Template	93
63 : Small Square on the Right Side Template	93
64 : Triangle Template	93
65 : Template for Vertical Line of Length 6	95
66 : Picture of corner in image	96
67 : Coin Flip Observations	103
68 : Table of Probabilities Computed for Coins	104
69 : Pixels in a Markov Random Field	107
70 : Combination of Noise Levels On Artificial Images	112
71 : False Positive Rate for the Combined σ Operator	113
72 : False Negative Rate for the Combined σ Operator	114
73 : Total Error Rate for the Combined σ Operator	115
74 : Evidence Combination of Noise Levels Applied to Vidicon Blobs Image	117
75 : Evidence Combination of Noise Levels Applied to Vidicon Gnome Image	119
76 : Evidence Combination of Noise Levels Applied to Carlos' ear	121
77 : Evidence Combination of Noise Levels Applied to Aerial Image	123
78 : Combination of Noise Levels On Artificial Images	127
79 : The combination of operators with several window sizes on circles image	129
80 : The combination of operators with several window sizes on rectangles image	130
81 : The combination of operators with several window sizes on combined image	131
82 : The combination of operators with several window sizes on all images	132
83 : Combination of Different Window Sizes on Blobs	134
84 : Combination of Different Window Sizes on the Gnome	135
85 : Combination of Different Window Sizes on Carlos' ear	136
86 : Combination of Different Window Sizes on the Sewage Treatment Plant	137
87 : Hueckel's model	141
88 : Hueckel's 2nd model	141

10 : Example of Digitized Area and Perimeter (Reprised)	173
89 : Source of Extraneous Features	175
90 : A Piecewise Polynomial Surface	176
5 : Definition of Image and Scene (Reprised)	182

Index of Defined Terms

posterior distribution	4
marginal probability distribution	4
feature	4
labels	4
false negative	5
false positive).	5
likelihood	7
domain model,	7
prior probability,	7
likelihood generator	7
Bayesian feature detector	7
size	8
power	8
scene.	10
image.	10
boundary pixel,	10
user	11
designer	11
primitive statistics.	12
required statistics	12
isotropy	14
approximation	14
discretization	14
window	17
corner pixel.	20
digitized area	21
digitized perimeter	21

basis	28
separable	31
orthographically	33
false negative	81
occlusion	81
false positive	81
extraneous detection	81
disjoint	101
physical probability distribution.	105
maximum entropy	105
facet models.	139
Edge detection	140
size	145

Chapter 1



Motivation

CHAPTER 1

Motivation

Currently a great variety of tools are available for low-level vision tasks. For tasks such as image reconstruction and edge detection many tools are available. The algorithms in this dissertation should ease certain aspects of low-level vision tool management.

Most of the tools in low level vision are algorithms that attempt intermediate steps towards a goal. As an example, in computer vision systems a boundary detection algorithm is not intended to display outlines pleasing to the human eyes¹. Output from such algorithms is input to higher level routines such as shape recognition programs or surface reconstruction programs.

1. Probabilities

All low-level vision tools make mistakes. Significant error rates must be accepted when finding object boundaries from limited support (often called edge detection). Thus the error characteristics of low-level vision algorithms need to be considered.

Consider some applications for boundary pixel detection. Relaxation and regularization algorithms suffer more from missing a boundary pixel than from extra

¹Compared to data analysis and image presentation systems where the data is directly presented to a human.

boundary pixels. If a boundary pixel is missed, regularization algorithms assume image characteristics change slowly across the boundary with disastrous results. Hough transform techniques often work effectively when the set of boundary pixels detected is sparse. However the cost of using the Hough transform is proportional to the number of boundary pixels detected (Sher 1985b).

If several tools require the same type of information (such as positions of boundaries) it is better to have a single shared routine than have each tool use its own version. But a boundary pixel detector that returns a true/false decision at each pixel can not suit both regularization techniques and Hough transform routines. Take for example the one dimensional slice shown in figure 1.



Figure 1 Slice through an image with an ambiguous boundary

A boundary pixel detector that is used as a first stage before regularization should report a boundary in the center of figure 1. A boundary detector that is used by a Hough transform line detection routine should report that figure 1 has no boundaries in it.

The traditional solution for satisfying differing requirements among intermediate level routines has been low-level detectors that generate numbers such as *edge strengths* rather than true/false decisions (Ballard and Brown 1982c). Strengths describe the confidence of the low-level vision algorithm that an event would occur for example a boundary passing through a pixel. For example, in figure 1 the detector returns a low edge strength. Figure 2 shows a 0 edge strength.

intensity \wedge _____

Figure 2 Slice through an image with an edge strength of 0

Figure 3 shows a high edge strength.

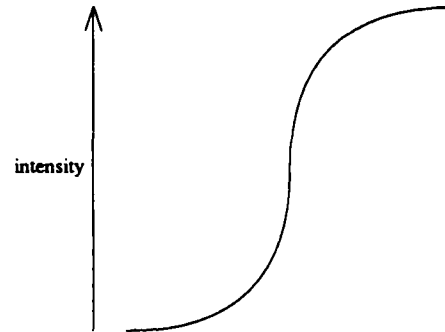


Figure 3 Slice through an image with a high edge strength

Often, the output of an edge detector that returns strengths is thresholded before the intermediate level application sees it. If an edge strength is higher than the threshold, a boundary is reported to the application. For each intermediate level application the threshold is found by experimentation on typical images. If the boundary detector is changed, a new threshold needs to be found for each application.

If all boundary pixel detectors output the probability of a boundary pixel, then the boundary pixel detector could be improved without finding new thresholds. Currently thresholds need to be recomputed whenever the boundary detector is changed because the relationship between the strengths returned by established boundary detectors (such as the Sobel and the Kirsch edge detectors (Ballard and Brown 1982a)) and the probability of an edge is currently unknown. If the error sensitivities of the intermediate level routines are known, thresholds can be determined by an application of decision theory. This dissertation discusses low-level vision algorithms that return probabilities.

2. Decision Theory

Decision theory is the statistical theory of decision making under uncertainty. An advantage of algorithms that generate probabilities is that results from decision theory can be applied to such algorithms. Decision theory assumes there is a set of possible states of nature, Θ , and a set of possible actions that can be taken, A . Each action, $a \in A$ when taken under a state of nature $\theta \in \Theta$ has a certain cost for the decision maker. Of course, the decision maker wants to minimize her costs. But she does not know the state of nature, only some observed data, o . o and some knowledge about the structure of the problem yields a probability distribution over Θ . This probability distribution is the *posterior distribution* over Θ . Decision theory assumes the decision maker picks the action that minimizes the expected cost derived from the posterior distribution (Berger 1980c).

In computer vision the observed data is the image or images being processed. For boundary detection the set of possible configurations of boundaries in the image is Θ . The number of boundary configurations in an image is approximately 2 to the power of the number of pixels in the image. It is impossible to calculate or store a probability distribution over such a large set.

Each pixel in the image either has a boundary passing through it or not. For each pixel, p , there is a *marginal probability distribution* over $\Theta_p = \left\{ \text{boundary}, -\text{boundary} \right\}$. In this dissertation Θ_p is called a *feature* and the elements of Θ_p are the *labels* for the feature. The total number of states in such distributions is linear in the number of pixels in the image rather than exponential in it. Thus, instead of one unmanageable decision problem, there are N (number of pixels) interrelated decision problems each with a small set of states of nature.

Hence for each pixel p there is a decision space Θ_p and an action space $A_p \equiv \Theta_p$ ². The decision (action) a_p is to choose an element of Θ_p to report. For example in a boundary detection problem $\Theta_p = \{ \textit{boundary}, \textit{-boundary} \}$ and there are four possible combinations of true states of nature and reported states of nature documented in figure 4.

State of Nature	Reported State	Cost
<i>boundary</i>	<i>boundary</i>	0
<i>boundary</i>	<i>-boundary</i>	c_n
<i>-boundary</i>	<i>boundary</i>	c_p
<i>-boundary</i>	<i>-boundary</i>	0

Figure 4 Situations using a Boundary Detector

Note that there is zero cost associated with reporting the correct state³ but there is a cost, c_n , for reporting *-boundary* when there is a boundary (the cost of a *false negative*) and there is a cost c_p for reporting *boundary* when there is no boundary (the cost of a *false positive*).

Figure 4 describes a decision problem. A strategy (algorithm) for making decisions has probabilities, p_p for the false positive state occurring and p_n for the false negative state occurring. Equation 1 uses these two probabilities to determine the expected cost for a strategy.

$$P(\textit{boundary})p_n c_n + P(\textit{-boundary})p_p c_p \quad (1)$$

Given a model that contains the probabilities and costs in equation 1, our decision maker can choose a strategy with a minimal expected cost.

²The set of actions are isomorphic to the set of states because the purpose of perception is to estimate the state of nature.

³If there is a non-zero cost for reporting the correct state of nature, there is an equivalent decision problem with a zero cost for being correct. For more details see (Berger 1980a).

How much has been lost by splitting the problem into a set of Θ_p (making a decision at each pixel) instead of using the set Θ of boundary configurations (deciding all the boundaries at once)? A theorem of decision theory says if the cost for reporting a wrong configuration is the sum of the costs of the errors made at each pixel then the union of the cost minimizing elements of each Θ_p corresponds to the cost minimizing element of Θ (Berger 1980a).

But, for most intermediate-level vision algorithms, the costs of several mistakes is not the sum of the costs of the individual mistakes. However it is still necessary to break up the space of configurations because of the difficulty of managing probability distributions over sets with cardinality larger than the number of atoms in the universe. It is a research issue to find the most effective way to split up the space of configurations for low-level vision problems (Morgera 1987).

Another way decision problems for low-level vision can be simplified is by not using all the observed data. In particular if a phenomena is restricted to a small part of the image such as a boundary pixel then one might consider the data in the image far from the feature to be irrelevant to the decision. Thus the decision about the presence or absence of a boundary pixel uses only a small region of the image (a window) about the boundary pixel. Section 2.2 discusses this simplification.

3. Likelihoods

Often it is easier to state and solve the image generation problem than the vision problem (which is why computer graphics can generate realistic images that current image understanding systems can't analyze). For example the statement that a signal is corrupted by uncorrelated Gaussian additive noise is a description of a probability distribution over observed signals and thus a description of how observed signals are generated. It is easier in many models to get the probability of the observed data given a labeling for a feature than to derive the probability of a feature labeling

directly from the observed data. The models described in chapter 2 have this property.

The probability that an image o is observed when a feature f has label l is the *likelihood* of l for o . $L_f(o|l)=P(o|label(f)=l)$ is shorthand notation for this likelihood. $prior_f(l)$, the Likelihoods and priors derive from a *domain model*, a set of assumptions about the relationship between the state of nature and the observed data. *prior probability*, is the probability that f has label l , using only the information in the domain model, ignoring the information in the observed image. A *likelihood generator* is an algorithm that uses a domain model M to estimate the likelihood of l for o . Thus $L_f(o|l \& M)$ is notation for the output of a likelihood generator. Given a likelihood generator for M and a prior estimate of the distribution of f 's labels then one can build using Bayes' Rule (equation 2) a feature detector for f that yields $P_f(l|o \& M)$, the probability that f has label l given the observed data and domain model.

$$P_f(l|o \& M) = \frac{L_f(o|l \& M) \text{prior}_f(l)}{\sum_{l' \in V} L_f(o|l' \& M) \text{prior}_f(l')} \quad (2)$$

The feature detector thus derived is called here a *Bayesian feature detector* for model M .

The set of likelihoods for a feature f given an observation o contains more information than (2) uses. The denominator in (2)

$$\sum_{l' \in L} L_f(o|l' \& M) \text{prior}_f(l') \quad (3)$$

is the probability that o would occur given domain model, M . If the probability is too low then the model being used probably is not correct. I use this information combined with *a priori* information about the reliability of the model to derive an evidence theory in chapter 5.

Likelihoods are used by Markov Random field algorithms to encode information about the observed data. Thus a likelihood generator can be used as input for the algorithms described in (Marroquin 1985), (Geman and Geman 1984), (Chellappa 1981a), (Hansen and Elliot 1982), (Cohen and Cooper 1987) and (Chou 1987).

Another use for likelihoods is derived from a classical statistical approach to decision problems. Some work solves estimation problems by reporting the label whose likelihood is maximized (Good 1983c) (Andrews and Hunt 1977c) (Shvaytser and Peleg 1985). This approach avoids considering prior probabilities for the feature labels and the cost of errors in estimation. However if the costs of all errors are not equal, maximum likelihood estimation may lead to costly errors (Sher 1986c).

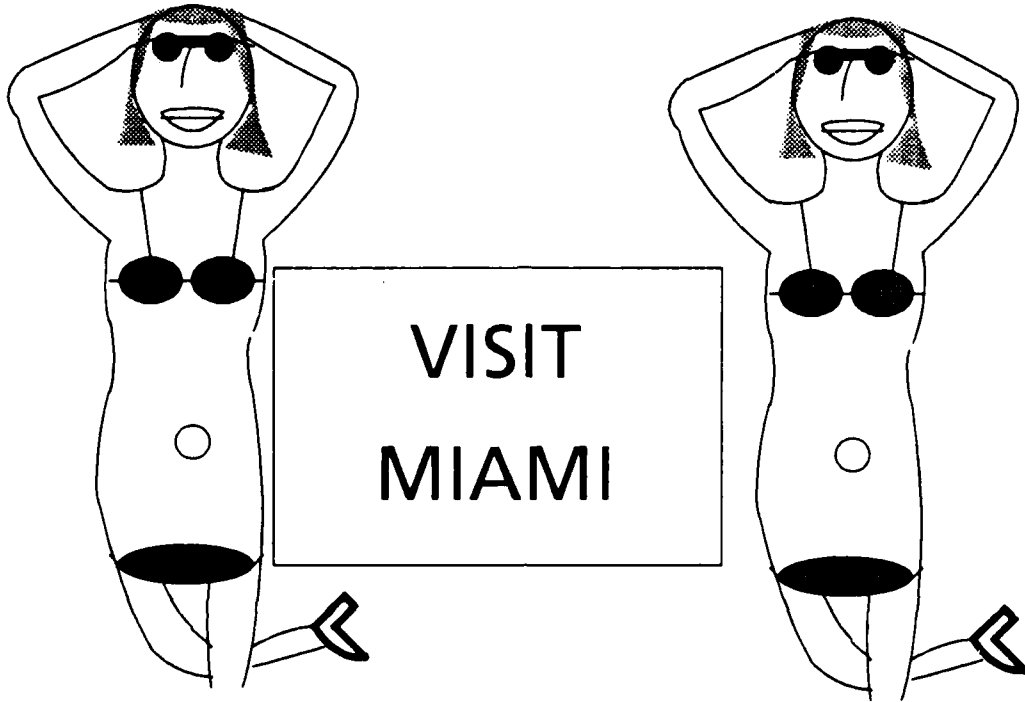
Another classical approach to decision theory is hypothesis testing (Good 1983a). Hypothesis testing is used when there are two hypotheses (labelings) that must be decided between, a null hypothesis and an alternate hypothesis. Consider the null hypothesis a negative result and the alternate hypothesis a positive result. Hypothesis testing involves finding an algorithm whose false positive rate is guaranteed before the data is seen to be less than a specified rate α . α is called the *size* of the test⁴. Given a set of algorithms whose size is α the test that minimizes the false negative rate $1-\beta$ is chosen. β is called the *power* of the test. It is a theorem of statistics that a likelihood ratio test maximizes β for a given α . Thus classical approaches to decision problems also call for calculation of likelihoods.

Hypothesis testing is used in vision algorithms that make binary decision (decisions between two mutually exclusive states of nature). For example in (Li and

⁴Also known as the level of significance for the test.

Dubes 1985) a hypothesis testing approach (using likelihood ratios) was taken to template matching. Here the decision was between the template matching and the match failing. In general, hypothesis testing is not useful for deciding between multiple hypotheses. Some work attempts to extend the hypothesis testing approach to handle multiple hypothesis decision problems (Ferguson 1967) but no such extension has gained universal acceptance. Since perception problems usually involve multiple hypotheses, the hypothesis testing is not a popular approach in vision research.

Chapter 2



Models

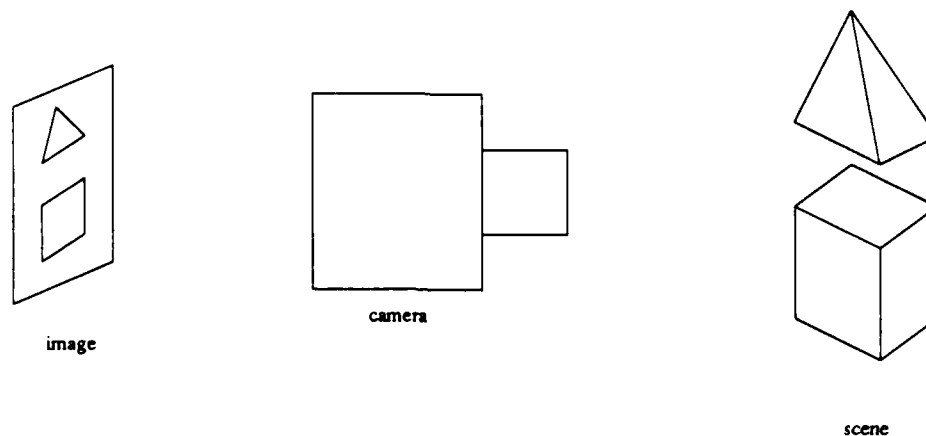
CHAPTER 2

Models

In section 1.3 equation 2 computes posterior probabilities for features from likelihoods and prior probabilities. Where do the prior probabilities come from? How can likelihoods be computed from observed data? The answer to these questions lies in the domain model.

A feature detector tries to label features given an image of a scene. A camera (or other data gathering device) is aimed at a *scene*. The output of a camera is an *image*. Figure 5 is an example of a scene and an image.

Figure 5 Definition of Image and Scene



A feature is a relationship between the scene and image. An example of a feature is a

boundary pixel, which is a pixel that measures light¹ reflected from two objects. A domain model for a feature detector serves several purposes.

- (1) It defines the feature.
- (2) It provides prior probabilities for the feature. Eg. expected frequency of boundary pixels.
- (3) It describes how the scene gets translated into an image and thus guides the construction of a likelihood generator. For example, a model could say that an image is corrupted with Gaussian additive noise of standard deviation σ . Section 3.2 uses such information to construct likelihood generators.

This chapter shows how domain models are constructed and how such models yield prior probabilities. Algorithms that yield likelihoods are derived from domain models in chapters 3 and 4.

1. Primitive and Required Statistics

Domain models bridge the gap between the user of a vision system and the designer. A *user* supplies a model of the world, the imaging device and the features to be extracted. The *designer* writes routines that derive likelihoods or probabilities for feature labels from the model and the image. A user expresses her model in terms of statistics familiar to her. A designer can build a likelihood generator if certain statistics are made available to him. The statistics the designer requires often are not those the user supplies.

For example a user can generally describe the objects that are likely to appear in images. For an automated factory these objects are components of the manufactured items, the assembly line and the robot arm. For aerial images some objects are

¹A pixel is really a measurement of the light hitting a region of the photosensitive part of a camera.

houses, roads, trees, and fields. Another parameter a user is likely to supply is the optics of the camera. Statistics that the user can supply to the designer are called here *primitive statistics*.

Assume a limited support boundary pixel detector is being designed. If the designer is given:

- (1) The ideal (noiseless) windows that correspond to boundaries and non-boundaries.
- (2) The level and type of noise corrupting the image.
- (3) The prior probabilities of boundaries passing through a pixel.

The designer can use techniques from chapter 3 to build a boundary detector. The statistics he uses are called the *required statistics* because the designer requires them to do his job. A domain model maps primitive statistics into required statistics.

2. Simplifications

A domain model is a mechanism for encoding the user's knowledge of the world in a form that yields required statistics. However the world a user describes is a complex place (see figure 6).

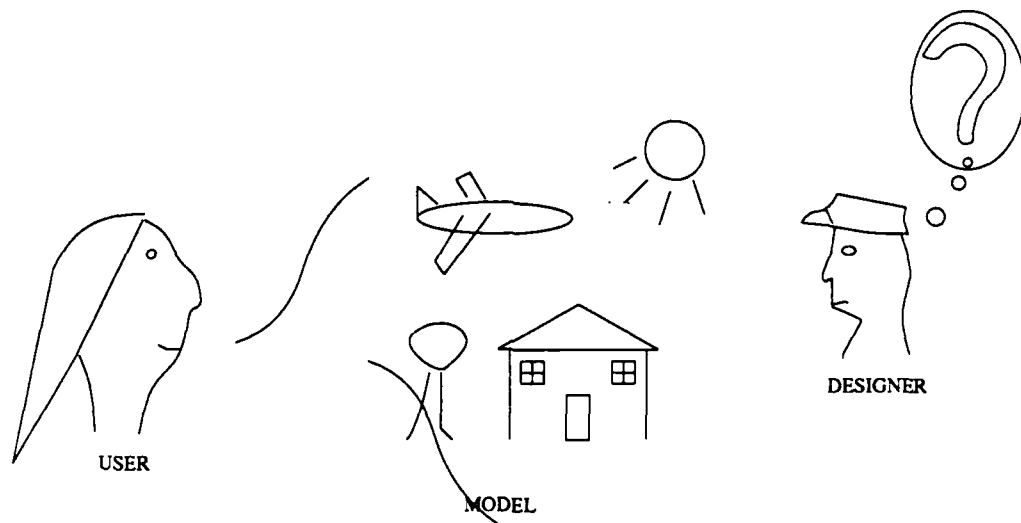


Figure 6 A Designer Confronted with a Complex User Model

Deriving a likelihood generator or priors from a model that is natural from the user's point of view can be difficult or impossible. Often the user's model must be simplified to be useful. Some of the common simplifications are:

Independence

One such simplification is an unwarranted assumption of independence. For example in house scenes garage doors are often highly correlated with kitchen windows. Using this correlation complicates the job of a door and window detector. A much simpler detector results if the positions of doors and windows are independent. So a common simplifying assumption is that the distribution of positions for an object in a scene is independent of the other objects. This simplification degrades the quality of the detector. But without this simplification building a detector might be impossible or require other equally destructive simplifications. A research issue is finding the degree degradation from such simplifications.

Isotropy

Another way to simplify a model is to assume an isotropy. An *isotropy* is a set of states with the same probability. An example is the assumption that all orientations are equally likely for objects (Witkin 1981). Thus at each position the probability of an object over orientations is uniform. This simplification was introduced by Laplace (Berger 1980a). Another common assumption is to assume that objects can occur at any point in the scene with equal probability. This assumption eases the derivation of a likelihood generator since the generator behaves the same at each point of the image. It also eases the user's job of specifying a model since otherwise a probability distribution over positions must be supplied for each object. The more common applications of maximum entropy (Shastri 1985) (Skilling and Gull 1985) are equivalent to certain isotropy assumptions. It should be easier to measure the results of an unwarranted isotropy assumption since the anisotropy is easily measured.

Approximation

An *approximation* assumption is the use of an approximation to a function instead of the function itself. An example of such a simplifying assumption is the *discretization* assumption. This assumption is that events (like objects being present) occur at discrete points in space instead of occurring continuously through space. This assumption eases building likelihood generators by reducing integrals to mere sums. This simplification is well known and its behavior is addressed in the numerical analysis literature about computing integrals and in the vision literature about discretizing gray levels (Andrews and Hunt 1977d) and discretizing accumulator arrays when using the Hough transform (Shapiro and Iannino 1979) (Maitre 1986).

Projection

Projecting a problem into a smaller space and modeling it in that space often is useful. For example in vision it is often simpler to try to model the image as a corruption of an image of 2 dimensional objects instead of modeling the 3 dimensional scene. Such a projection turns scenes like figure 5 into scenes like 7. Projection need not degrade the model or detector. Often a projection paves the way for other simplifications.

Low Probability

It is often convenient for the designer of a likelihood generator to ignore certain low probability events. For example when building a boundary pixel detector it is inconvenient to model at a pixel the low probability event of three or more objects intersecting. A boundary detector designed thus, fails on corners where 3 objects meet. Section 2.3 discusses calculating the probability of such an event. Ignoring low probability events makes a detector behave unpredictably when these events occur. Chapter 5 shows how to merge a detector for low probability events into a detector that ignores these events.

These are some of the common simplifications that are applied to models. Chapters 5 and 4 use all these techniques. Stating the simplifying assumptions when building detectors is useful because those assumptions may be relaxed in further work. For example it is common to start work in boundary detection on the 1 dimensional problem (thus assuming an image is a set of uncorrelated 1 dimensional signals) and then adjust the detector to handle 2 dimensional images. (Canny 1986) (Boie, Cox, and Rehak 1986) (Sher 1986b).

3. Derivation of Priors

This section derives a required statistic, the prior probability of a boundary intersecting a pixel, from a simplified set of primitive statistics.

Here are some simplifying assumptions that ease the derivation. The first is to project the scene into 2 dimensions. Thus the scene is modeled as a 2 dimensional entity. A scene consists of a set of two dimensional objects overlaying a background. Two dimensional objects are laid atop each other as in figure 7.

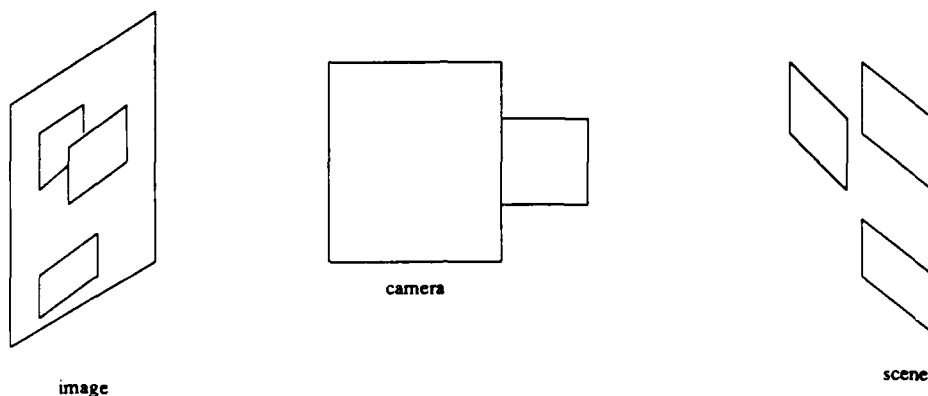


Figure 7 Example of Simplified Model

For many scenes this assumption is safe.

Other simplifying assumptions made here are that all positions are isotropic. Thus objects are equally likely anywhere, and the placements of objects are independent events. Another simplifying assumption, which is relaxed later in this section, is that a sufficiently large number of objects are placed in the scene so that placing a new object does not change the expected number of boundary pixels. Thus the scene is so cluttered that adding a new object covers up as many boundary pixels as it adds. Markov chain theory can be used to prove that this state is approached by a scene with a large number of objects.

Primitive statistics that a user could supply are the average size and perimeter of objects. Thus the average number of pixels an object covers is c and the average number of pixels on the boundary of an object is p . Placing a new object on the scene covers all the boundary pixels below it and adds on average p new boundaries to the

scene. Thus if b is the prior probability that a boundary passes through a pixel then the expected number of boundaries added by a new object is the left side of equation 4.

$$p - cb = 0 \quad (4)$$

cb is the expected number of boundary pixels covered by a newly placed object. We assume, in the scene, the expected change in the number of boundary pixels resulting from placing an object is 0. Thus the prior probability that a pixel is a boundary must be $\frac{p}{c}$. Given our assumptions, this formula is a source of prior probabilities for boundary pixels. For example if the objects in the scene are 10 by 10 pixel squares the probability of a boundary passing through a pixel is 0.36. If the objects are 100 by 100 pixel squares the probability of a boundary is 0.0396.

Another interesting statistic is the number of windows on the image with 3 objects in them. A *window* is a subimage as in figure 8. Windows are usually rectangular.

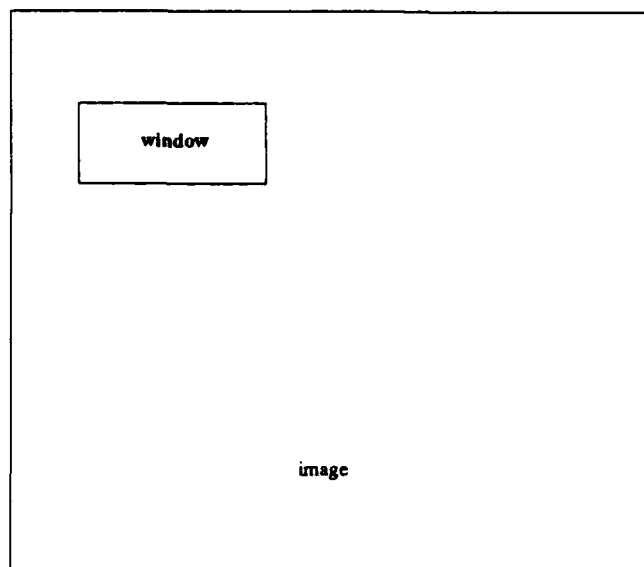


Figure 8 Small Window on an Image

In windows with more than two objects simple edge detectors are likely to fail. Two of the most common ways 3 objects can participate in a window are shown in figure 9.

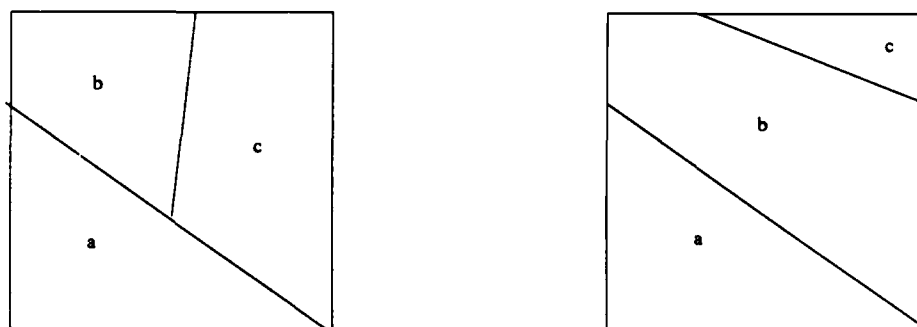


Figure 9 3 objects (a, b, c) in one window

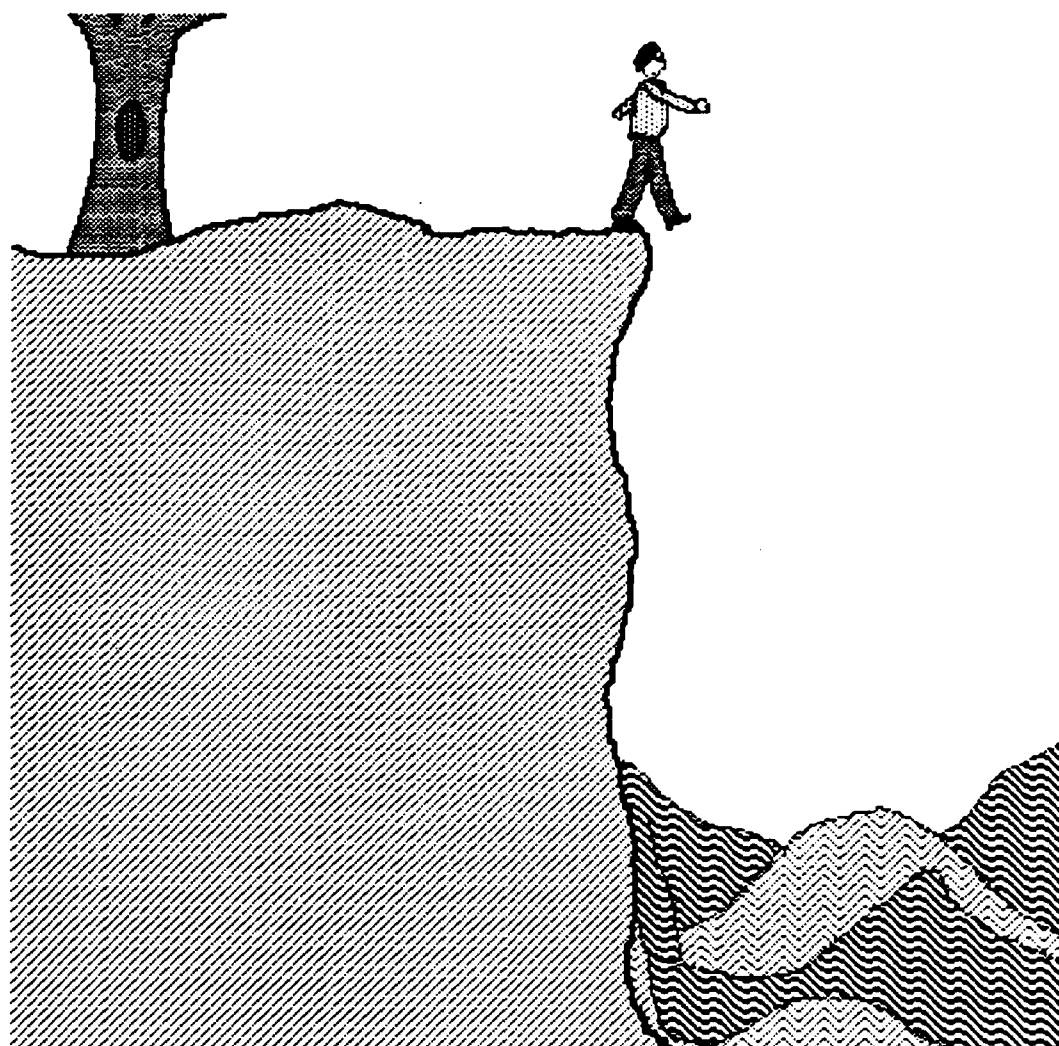
One way 3 objects can participate in a boundary is if two boundaries intersect (the left side of figure 9). The probability of two boundaries intersecting in a pixel is b^2

since the two intersecting boundaries are laid down independently. Thus the probability of two boundaries intersecting at least once in a N pixel window (at least once) is $1-(1-b^2)^N$. This probability is a lower bound on the prior probability that 3 objects occur simultaneously in a window. If the objects are 10 by 10 squares and the window is 5 by 5 then the probability of a corner is 0.97; the scene is so cluttered that a 5 by 5 edge detector is not likely to succeed on many windows. If the objects are 100 by 100 squares then the probability of a corner in a 5 by 5 window is 0.038. Thus for any particular window of the image an edge detector is unlikely to fail from a corner.

If scenes are not cluttered, the assumption that placing a new object in the scene does not change the expected number of boundary pixels fails. Aerial images often have few objects placed against a neutral background. Adding a new object adds more boundaries to these scenes. Assume the expected number of objects in an $N \times N$ image is E . Let $f=E/N^2$. Ignoring occlusion the probability of a boundary falling on a pixel is fp . However occlusions delete $fpf(c-p)$ boundaries. Thus the probability that a boundary passes through a specified pixel is $fp(1-f(c-p))$. If objects are 10 by 10 squares and $f=0.001$ (thus a bit less than 0.1 of the image contains objects and the rest background) then the probability of a boundary is 0.033696.

Thus prior probabilities can be derived from the simple models described here. It is a research problem to derive prior probabilities for required statistics from more sophisticated models.

Chapter 3



Boundary Detection

CHAPTER 3

Boundary Detection

This section describes a model for boundary pixel detection, using simplifications *a la* section 2.2. This model yields an efficient algorithm for boundary detection. This algorithm has been implemented and tested. The results of these tests are reported here.

The purpose of a boundary pixel detector is to detect which pixels measure light from exactly 2 objects. If a pixel measures light from 3 or more pixels it is called a *corner pixel*. Because of noise, it is impossible to determine with certainty which pixels are boundary pixels. Thus a boundary pixel detector need determine for each pixel the probability a boundary passes through it. Section 1.2 supports deciding where the boundaries are on a pixel by pixel basis.

Thus a boundary pixel detector tries to determine a probability distribution over a set of labels at each pixel. In the literature several labeling schemes have been used for boundary pixel detection. The simplest is to use the set in equation 5.

$$L = \{ \textit{boundary}, -\textit{boundary} \} \quad (5)$$

Algorithms that assign this labeling are often called undirected edge detectors. Object boundaries often are smooth and thus have well defined tangents and normals. Higher level routines use information about the normals of the boundaries. Thus a labeling scheme that uses a set of angles, A to get equation 6 is often useful.

$$L=A \cup \{\neg boundary\} \quad (6)$$

Boundary pixel detection (often called edge detection) is an important first stage in many low-level vision systems. It can be a first step in segmenting an image by split and merge segmentation (Furst and Caines 1986) (Nazif and Levine 1984). Boundary pixel detection often is used as input to a template matching system for object detection (Ballard 1981) (Turney, Mudge, and Volz 1985) or image line and circle detection (Zucker, Hummel, and Rosenfeld 1975) (Hough 1962).

1. Models for Boundary Detection

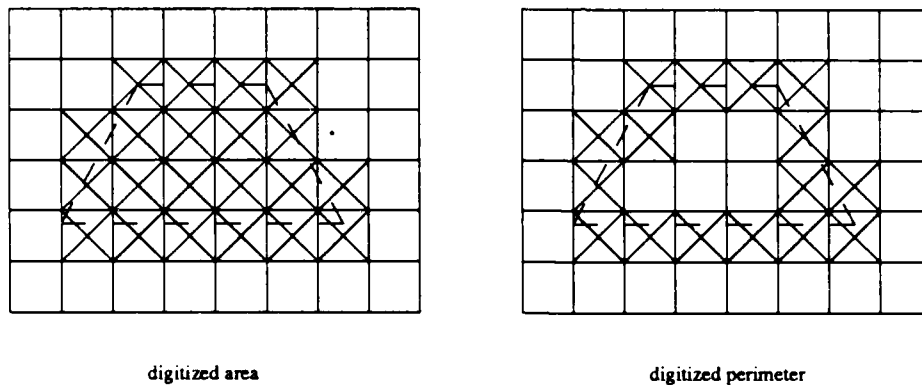
Section 1.3 equation 2 shows that the probability distribution of boundary pixels can be derived from prior probabilities for the labels and likelihoods of labels from the image.

$$P_f(l|o \& D) = \frac{L_f(o|l \& D) \text{prior}_f(l)}{\sum_{l' \in V} L_f(o|l' \& D) \text{prior}_f(l')} \quad (2)$$

These are the required statistics for boundary detection. These probabilities are derived from more primitive statistics with the model refined here.

1.1. Priors for Boundary Pixel Detection

Section 2.3 uses a simplified model to derive prior probabilities for the labels of equation 5 from the expected digitized area and perimeter of silhouettes of objects in the image. The *digitized area* of a silhouette is the number of pixels that measure light from the object. The *digitized perimeter* of an object is the number of pixels that measure light from an object that also measure light from another object or background. Both of these concepts are illustrated in figure 10.



Object outline is shown by dashed lines.

Crossed boxes are the pixels included in the digitized area and perimeter respectively.

Figure 10 Example of Digitized Area and Perimeter

The priors for the labels of equation 6 can be derived from the model of section 2.3 by assuming isotropy of orientation, thus boundary pixels are equally likely at all orientations. On the other hand the user can supply the orientations of the boundaries on the objects expected in the scene and the probabilities that each object appears at each orientation. Those probabilities yield the prior probability for each possible orientation of a boundary pixel.

1.2. The Facet Model

Thus the prior probabilities for each label can be calculated using the reasoning from section 2.3. To apply equation 2 the likelihoods for each label given the data, $L_f(a|I&D)$, need be calculated. This chapter contains algorithms that yield likelihoods given an image and primitive statistics.

Two important models in the edge detection literature are the step edge model and the facet model. The step edge model assumes that an image consists of a set of

piecewise constant regions corrupted by linear blur and Gaussian additive noise. This model lies at the basis of Canny's (Canny 1986) and Nalwa's (Nalwa and Binford 1986) edge detector. The facet model (Haralick 1980) assumes that an image is piecewise polynomial with small gradient corrupted by linear blur and Gaussian additive noise and digitization. The degree of the polynomials is specified in the domain model. A facet model with 0 degree polynomials is close to the step edge model. Haralick's work often uses a cubic polynomial approximation (Haralick 1986a). Haralick has also developed detectors using the step edge model (Haralick 1984). Here a windowed facet model is derived using simplifying assumptions about objects' shapes and reflectance maps. The degree of the model depends on the size of the window and the degree of simplification introduced into the model.

The first assumption is that objects are opaque and large compared to the wavelength of light. Thus if one object completely occludes another no light from the other object gets to the image as in figure 11.

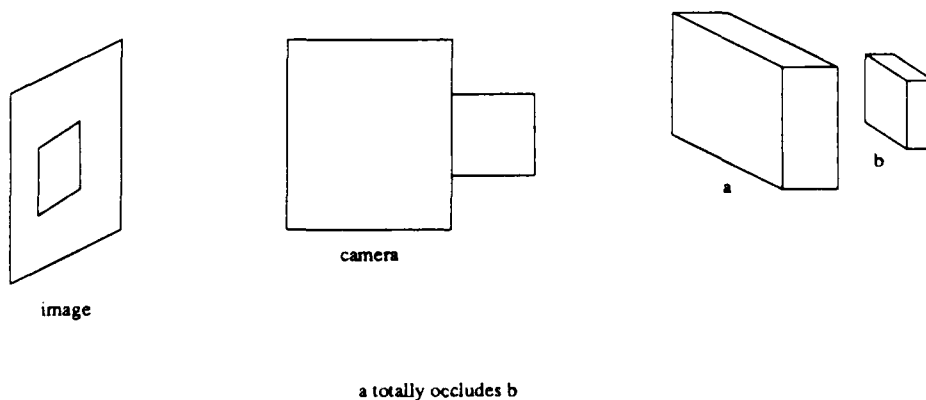


Figure 11 Example of occlusion

Without these assumptions boundaries are not well defined.

This section accepts the simplifying assumptions from section 2.3 for cluttered scenes. All the light in the scene is assumed to be reflections from objects (if necessary postulate a large object in the background). Internal reflectances (light reflected from one object onto another and then to the camera or light reflected from an object onto itself and then to the camera) and shadows are considered low probability events and thus ignored in our model. When these assumptions are violated a detector derived from this domain model behaves unpredictably. Thus if shadows are actually not low probability events then the detector can make frequent mistakes. Using the evidence combination theory in section 5 to combine such a detector with a detector that knows about shadows mitigates this problem.

(Horn 1986) discusses in detail how light is reflected from an object and how light from objects is formed into an image by a camera. Suffice to say that if

- (1) objects have Lambertian reflectance
- (2) their surface normals are slowly varying
- (3) the lighting changes slowly over the scene

then the light entering the camera is a piece-wise slowly varying function with each *piece* corresponding to the light reflected from an object. The camera introduces blur to the image and Gaussian additive noise. Shadows, sharp corners on objects and non-Lambertian reflectances cause this domain model to fail.

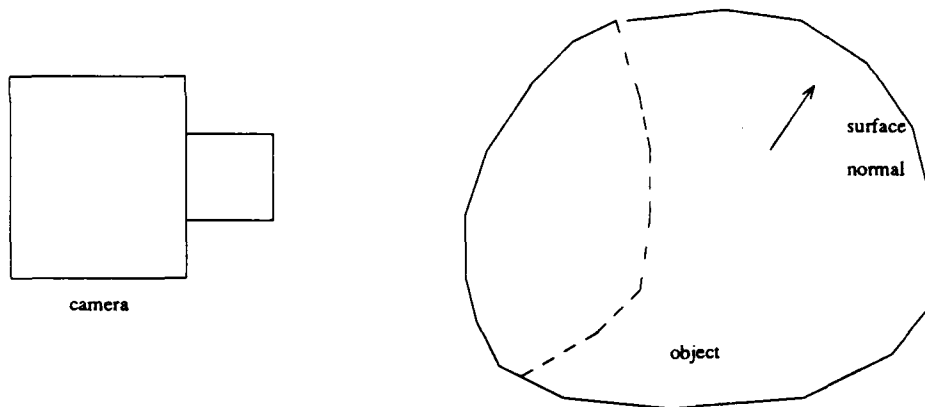
Small windows on an image often behave more regularly and are easier to model than an entire image. In a small window a slowly varying region can be approximated closely by a polynomial of small degree. Thus a window whose pixels measure light from a single object is modeled by a polynomial of small degree. If a window measures light from two objects it is modeled by two small degree polynomials with pixels on the boundary interpolating the two polynomials. Thus given this set of simplifying assumptions the facet model of Haralick is true for small

windows. Section 3.2 derives a likelihood generator for boundary pixels from this model.

1.3. Linear Boundaries

To derive a likelihood generator for a facet model for small windows several more statistics need be derived from the model.

In the model in section 3.1.2 objects have slowly varying surface normals. The projections of objects into silhouettes have boundaries with little curvature since the curvature of the boundary is a smooth function of the surface normals of a path on the surface of the object as illustrated by figure 12. If the object is locally smooth then the path (dashed curve in figure 12) that is the edge of the object that the camera sees is projected into a smooth curve on the image.



Edge of what the camera sees is dashed.

Figure 12 Relationship of Surface Normals to Boundaries

Thus boundaries between two objects are locally smooth. One can assume boundaries between two objects are linear in small windows. This assumption is an approximation assumption.

In section 2.3 the probability that more than two objects intersect in a window is calculated for a model. For small windows this probability can be small. Thus for small windows one can use the low probability simplification to only consider the cases where one or two objects' images fill a window.

Thus a small window is assumed to be entirely filled with a polynomial digitized and corrupted by blur and noise or we assume there are two polynomials with a linear boundary between them digitized and corrupted by blur and noise. To calculate the likelihood one more parameter need be supplied. This parameter is the prior probability distribution for the polynomials that are yielded by object images. These parameters can be derived from probability distributions over object reflectances, lighting conditions, surface orientations, and surface curvatures. Thus if the user supplies these distributions the mathematics in (Horn 1986) yields the distribution of polynomials. Now our model supplies enough required statistics to build a likelihood generator for boundary pixel detection.

2. Likelihood Generator

This section derives an algorithm for generating likelihoods for boundaries from the facet models described in section 3.1. We assume that the feature label l represents a probability distribution over image windows before noise. For example if l is that a window contains no boundaries in the step edge model¹ then l represents a distribution over constant functions on that window. If l is a vertical boundary then it represents the sum of a constant function on the left side of the window with a constant function on the right side of the image whose intensities are independently selected from a known distribution. Thus to each label l corresponds to a set of functions $\{f_l\}$ and a probability distribution over it. Section 3.1 describes a model

that yields such a mapping.

Assuming a blur function B and a correlation matrix for the Gaussian additive noise C the likelihood of a label l given a window on the observed image W is approximated by equation 7. This equation is an application of probability theory and the formula for Gaussian additive noise.

$$L(W|l) = \int_i \kappa e^{-\frac{1}{2}(W-B(f_i^l))C(W-B(f_i^l))^T} dP(f_i^l) \quad (7)$$

Equation 7 assumes that W and $B(f_i^l)$ are vectorized. Equation 7 is an approximation since the effect of discretization on W is ignored. Equation 7 can be simplified into a more usable form in equation 8.

$$L(W|l) = \kappa e^{WCW^T} \int_i e^{WCB(f_i^l)^T - \frac{1}{2}B(f_i^l)CB(f_i^l)^T} dP(f_i^l) \quad (8)$$

For facet models each member of $\left\{ f_i^l \right\}$ can be characterized as a weighted sum of a small set of functions. For example when l is a window with no boundaries in the step edge model then all f_i^l are of the form $\alpha \mathbf{1}$ where $\mathbf{1}$ is the unit constant function. Similarly when l is the window with no boundaries in the facet model of degree 1 then all the f_i^l are of the form $\alpha \mathbf{1} + \beta \mathbf{x} + \gamma \mathbf{y}$. If l is a vertical boundary in the center of the window in the step edge model then f_i^l is of the form $\alpha \mathbf{l} + \beta \mathbf{r}$ where \mathbf{l} is the unit constant function on the left side of the window and 0 on the right, and \mathbf{r} is the unit constant function on the right side of the window and 0 on the left (see figure 13).

¹Remember that the step edge model is a 0th order facet model.

1	1	0	0
1	1	0	0
1	1	0	0
1	1	0	0

l

0	0	1	1
0	0	1	1
0	0	1	1
0	0	1	1

r

Figure 13 Basis

Call the functions whose weighted sums span $\{f_i^l\}$ the *basis* of l . Thus in the facet model a label, l is a probability distribution over sets of scalars $\{\alpha_i^l\}$. $\{\alpha_i^l\}$ is translated by the basis of l , $\{b_j^l\}$, into f_i^l .

Equation 8 is modified to take the basis of l into account in equation 9.

$$L(W|l) = \kappa e^{-1/2 WCW^T} \int_i e^{WCB(\sum_j \alpha_j^l b_j^l)^T - 1/2 B(\sum_j \alpha_j^l b_j^l)CB(\sum_j \alpha_j^l b_j^l)^T} dP(\{\alpha_i^l\}) \quad (9)$$

If B is a linear function the sum inside B can be distributed in equation 9 yielding equation 10.

$$L(W|l) = \kappa e^{-1/2 WCW^T} \int_i e^{\sum_j \alpha_j^l WCB(b_j^l)^T - 1/2 \sum_{j,k} \alpha_j^l \alpha_k^l B(b_j^l)CB(b_k^l)^T} dP(\{\alpha_i^l\}) \quad (10)$$

Equation 10 shows that $L(W|l)$ can be computed from WCW^T and $\{WCB(b_j^l)\}$ and prior knowledge about l . Thus the likelihood of l is a function of the autocorrelation

of the window and the correlation² of the window with the blurred basis of l .

When $P(l|W)$ is calculated from $L(W|l)$ and $L(W|\neg l)$ using Bayes' law the term dependent on WCW^T drops out of the formula. Thus the probability distribution over the labels depends only on the correlations not the autocorrelation WCW^T . However the evidence combination theory of chapter 5 requires that $L(W|l)$ be calculated and uses the autocorrelation of W .

In the step edge model if l is that there is no edge in the window then its basis is the unit constant function. Assuming no blur the likelihood of an edge is a function of W 's autocorrelation and the correlation W with the unit constant function. For a vertical central boundary in the step edge model the basis is $\left\{ \mathbf{1}, \mathbf{r} \right\}$.

$\log L_f(W|l)$ is the sum of a constant, the autocorrelation of W and a function F of the correlations of W with the elements of the blurred basis of l . F has been found experimentally to be smooth and locally near quadratic for the step edge model with uncorrelated noise and no blur. Thus F can be calculated accurately by table lookup and interpolation. The functional form of F indicates that it should be smooth in more complex models that include correlated noise, linear blur and higher order polynomials. However the larger the basis of l the more convolutions are required to compute a likelihood and the more difficult the interpolation (because interpolation in high dimensional spaces is difficult).

For the experiments in section 3.5, F was computed by table lookup.

²The unnormalized correlation mediated by the covariance matrix C . Similarly, in this dissertation, WCW^T is referred to as the autocorrelation of W .

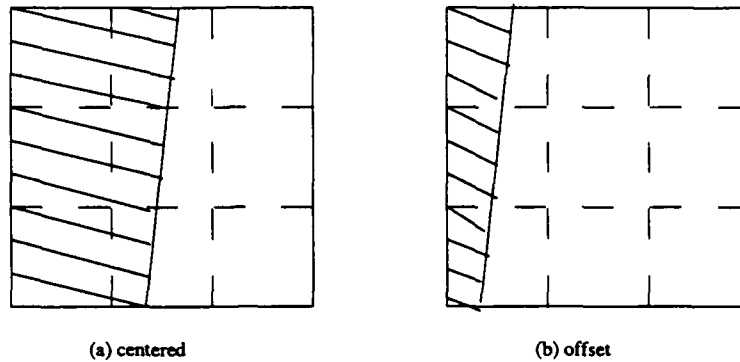
3. Efficient Computation of Likelihoods

Section 3.2 describes the functional form of the likelihood of a label l and an algorithm that calculates likelihoods from a probability distribution on the scalars of l . This section provides methods for increasing the efficiency of this algorithm.

The algorithm of section 3.2 requires correlating the image with a windowed, discretized and blurred basis for l . Thus any technique that speeds correlations speeds this algorithm. Correlations can be implemented by convolutions. For large window sizes convolutions are speeded by the Fourier transform (Ballard and Brown 1982b).

In the likelihood generation algorithm the image is correlated by many functions and the work of transforming the image is amortized over all the correlations. Thus the cost of using the Fourier transform for correlation for the likelihood generator is nearly half the time required for correlating a single image since only the multiplication by the transformed function and the inverse transformation is required for the correlation with each function. However for small windows it is faster to compute the correlation for each window directly. (Sher 1985b) discusses the choice of algorithm for correlation on multiprocessors.

For boundary detection the basis of one label often is an offset version of part of the basis for another label. For example in figure 14 the template on the right, (b), is an offset version of the template on the left (a). 14a is part of the basis of a centered edge. 14b represents a part of the basis for having no edge in the center of the window (since the boundary does not pass through the center pixel).



pixel boundaries shown by dashed boxes

Figure 14 Two Templates for the Same Edge

The likelihood computed from 14a for the window one pixel to the left approximates the likelihood computed for 14b. Thus only the likelihoods for boundaries through the center pixel need to be calculated. Likelihoods for boundaries that pass through another pixel in the window are calculated as for a boundary passing through the center of an offset window. This technique speeds up likelihood generation substantially.

Another efficiency gain can be found if equation 10 is separable for the correlations. $L(W|I)$ is the product of a function of the autocorrelation of W and a function F of the correlations of W with the basis. F is *separable* if F is the product of functions of each correlation as in equation 11.

$$F\left\{WCB(b_j^l)\right\} = \prod_j f_j(WCB(b_j^l)) \quad (11)$$

F can be separable if the probabilities of the scalars (that the bases are multiplied by) are independent so that equation 12 is true and the b_j^l are orthogonal in the sense of equation 13.

$$p\left\{\alpha_i^l\right\}=\prod_j p\left(\alpha_i^l\right) \quad (12)$$

$$\forall_{i \neq j} B\left(b_i^l\right) C B\left(b_j^l\right)=0 \quad (13)$$

As an example when l and r are used as a basis, the scalar corresponding to l is proportional to the reflectance of the object on the left side and the scalar corresponding to r is proportional to the reflectance of the object on the right side of the window. These reflectances are independent under the assumptions made here. Hence the probability distributions for these two scalars are independent.

F is separable if the probabilities of the scalars are independent and the basis is orthogonal. A basis is orthogonal if any two elements have a correlation of 0. Under no blur and with independent noise and no discretization l and r have a correlation of 0. Thus they are an orthogonal basis. However if after discretization the boundary passes through a pixel (rather than between two pixels as in figure 13) then the discretized l and r both contribute light to the boundary pixels. Thus the discretized l and r have a positive correlation. If the boundary occurs between pixels then even discretized l and r have a correlation of 0.

When the probability distribution is independent and the basis is orthogonal F is separable and can be computed as the product of a set of functions of single correlations. Separability allows efficient and accurate computation of the likelihood of features.

4. The Flexibility of the Likelihood Generation Algorithm

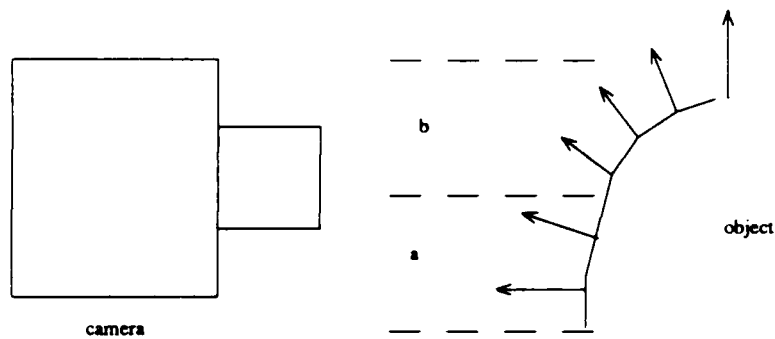
The algorithm of 3.2 was designed with a facet model in mind. However this algorithm can be used to generate likelihoods for any feature described by a probability distribution over the scalars for a basis. Many simplifying

approximations were required to derive a facet model, from an image model of the type documented in (Horn 1986). One could directly derive a basis for boundary detection from the model in that book.

First a simplifying assumption of local planarity is required for the interior of objects. Such an assumption is common in shape from texture work (Ikeuchi 1980) (Swain 1985). Also calculations are simplified if the scene is observed orthographically. A scene is viewed *orthographically* if the camera has infinite focal length. Such imaging is approximated by using a long range lens on the camera.

A window that does not fall over a boundary under these assumptions observes a region of an object of uniform reflectance. Thus it should have uniform graylevel. The basis for this type of window is **1**, and the probability distribution is calculated from the probability distribution over lightings and reflectances for objects and the of surface normals at pixels in the image. The probability distribution of lightings and reflectances of objects need to be supplied by the user. Given an assumption of isotropy of surface normals for objects (Witkin 1981) calculates a probability distribution for 3d orientations in the image under orthography.

Near boundaries the assumption of local planarity fails. In a small region of the image the normals of the part of the object projected into the region changes rapidly. Figure 15 shows why local planarity works except near boundaries. 15a does not contain a boundary and the surface orientation changes slowly. 15b does contain a boundary and the surface orientation changes quickly there.



dashed lines demark regions a and b
 arrows show surface normals

Figure 15 Surface Normals Change Quickly Near Boundaries

A basis for a central vertical occlusion boundary has the occluded object modeled by a uniform graytone with a probability distribution for its intensity. The occluding side is modeled by the side of a cylinder under unit reflectance, and lighting. The radius of the cylinder is determined by the average radius of curvature of the objects in the image. If there is a large variance in radii of curvature several cylinders may be used as a basis. The probability distribution over the scalars is determined by the distribution of lighting reflectance products. The probabilities of scalars for each side are independent since the reflectances of the two objects are independent.

Using the basis described here an occlusion boundary detector that is superior to a facet model detector can be built, if the user supplies a sufficiently accurate model.

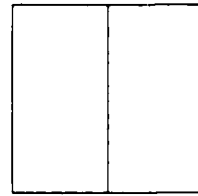
5. Experiments

Experiments have been run to test the likelihood generators described in section 3.2, by determining error rates and comparing them to established algorithms. The algorithm tested here is the algorithm derived in 3.2 for the step edge model. We find

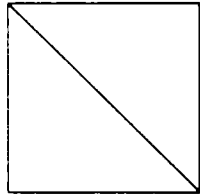
likelihoods for boundaries of four orientations shown in figure 16.



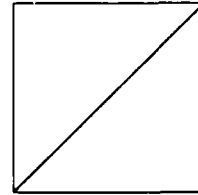
I. horizontal



II. vertical



III. diagonal



IV. slash

Figure 16 4 Orientations for Boundaries

For small windows, finding likelihoods for only these 4 orientations is sufficient to build an oriented boundary detector. The technique described in section 3.3 for taking windows with non-central edges into account was used for these results.

The prior probability of a boundary pixel in these experiments was set (somewhat arbitrarily) to 0.1. Boundaries in all 4 directions were considered equally probable. This prior was used both for applying Bayes' law to the likelihoods of the boundaries and when taking into account the likelihood of boundaries that pass through the window but not the central pixel.

3 kinds of aliasing were considered in the detector,

- (1) each object contributes 50% of the light for each boundary pixel

- (2) the left object contributes 100% of the light for each boundary pixel
- (3) the right object contributes 100% of the light for each boundary pixel

Evidence combination techniques from section 5 are used to combine the results from these 3 kinds of aliasings.

Likelihood generators have been built for window sizes of 5x5, 7x7 and 9x9.

5.1. Results with Artificial Images

To test these likelihood generators, they are applied to test images generated by a package of graphics routines. This package was written by Myra Van Inwegen and is described in an upcoming technical report (Sher and Inwegen 1987). This graphics package is used for testing because it generates images with known noise levels and known boundary positions. Other parameters that are specified are the distribution of object graylevels and positions.

The likelihood generators have been applied to four test images generated by this package. One is an image of two circles shown in figure 17. It tests the response of a boundary pixel detector to curvature, angle, noise, and contrast.

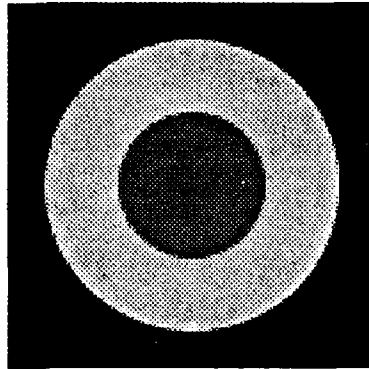


Figure 17 Two Circle Image

Three more challenging and complex images have been tested. They are as follows:

- (1) Circles with gray-level uniformly distributed between 0 and 254, with uniformly distributed positions and normally distributed radii, placed on a black background: figure 18.

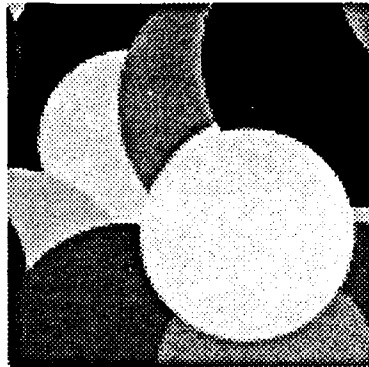


Figure 18 Circles Image

- (2) Rectangles with gray-level uniformly distributed between 0 and 254, with uniformly distributed positions and normally distributed lengths and widths and uniformly distributed orientations, placed on a gray background: figure 19.

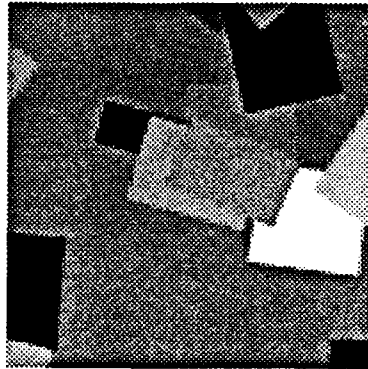


Figure 19 Rectangles Image

- (3) Rectangles and circles with gray-level uniformly distributed between 0 and 254, with uniformly distributed positions and normally distributed shapes, placed on a gray background: figure 20.

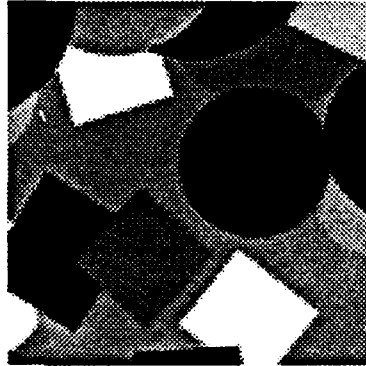


Figure 20 Combination of Circles and Rectangles Image

The software that generates these images also generates the positions of the boundaries. There is software that counts how many mistakes (false positives and negatives) are made in boundary determination. *False negatives* are when a boundary that is in the image is missed. *False positives* are boundaries that are reported where there is no boundary there.

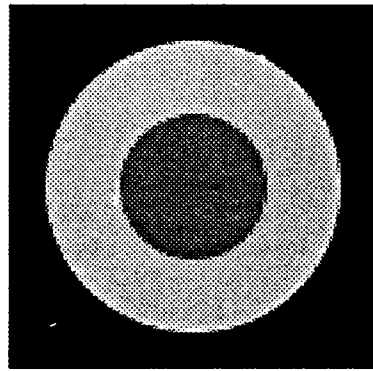
One tricky point is that multiple reports of boundaries are usually considered a bad result (Canny 1983). However systems that report a boundary only once sometimes have a high false negative rate because they report an edge one pixel off from where it really is. This error has low enough cost to be ignored. So the false negatives that are next to positives (true or false positives) and are parallel with them. Thus a false negative for a vertical boundary is not counted if there is a reported boundary to the left or right of it. Similarly a false negative for a horizontal boundary is not counted if there is a positive above or below it.

Artificial images are useful because the ground truth is known and error rates can be computed. It would take sophisticated equipment to make laboratory images with this property. Another convenient property of artificial images is that one parameter of the image such as the noise level or the distribution of objects' gray-levels can be varied without varying any other parameter.

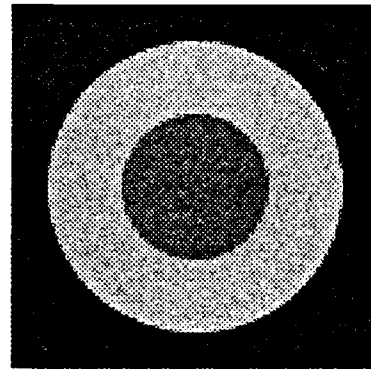
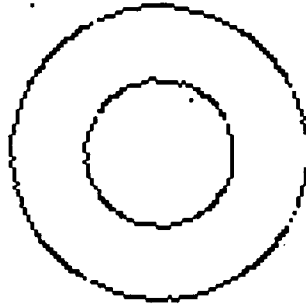
5.1.1. Angle and Orientation

The two circle image (figure 17) is a particularly good image to test the effect of boundary orientation, curvature and contrast on boundary detection. Figure 21c shows the result of using a 5x5 operator tuned to standard deviation 12 additive noise³ on image 17 with standard deviation 12 noise added to it. Figure 21d shows the result of using a 7x7 operator tuned to standard deviation 12 noise on image 17 with standard deviation 12 noise added to it. Figure 21e shows the result of using a 9x9 operator tuned to standard deviation 12 noise on image 17 with standard deviation 12 noise added to it. The images are black at pixels with greater than 50% probability of having a boundary and white at points less than 50% probability. Note that the larger operators are more sensitive to orientation and less sensitive to noise.

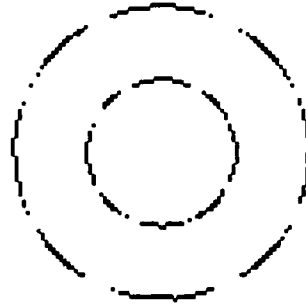
³The images in this paper always range from 0 to 254 in gray level. The standard deviation of the noise is also reported in graylevels.



a: Image

b: Image with $\sigma=12$ noise

c: 5x5 Operator



d: 7x7 Operator



e: 9x9 Operator

Figure 21 Oriented response for likelihood generator

Figure 22 shows the response of the 5x5 $\sigma=12$ operator for the 4 directions described in figure 16 with $\sigma=12$ uncorrelated Gaussian additive noise.

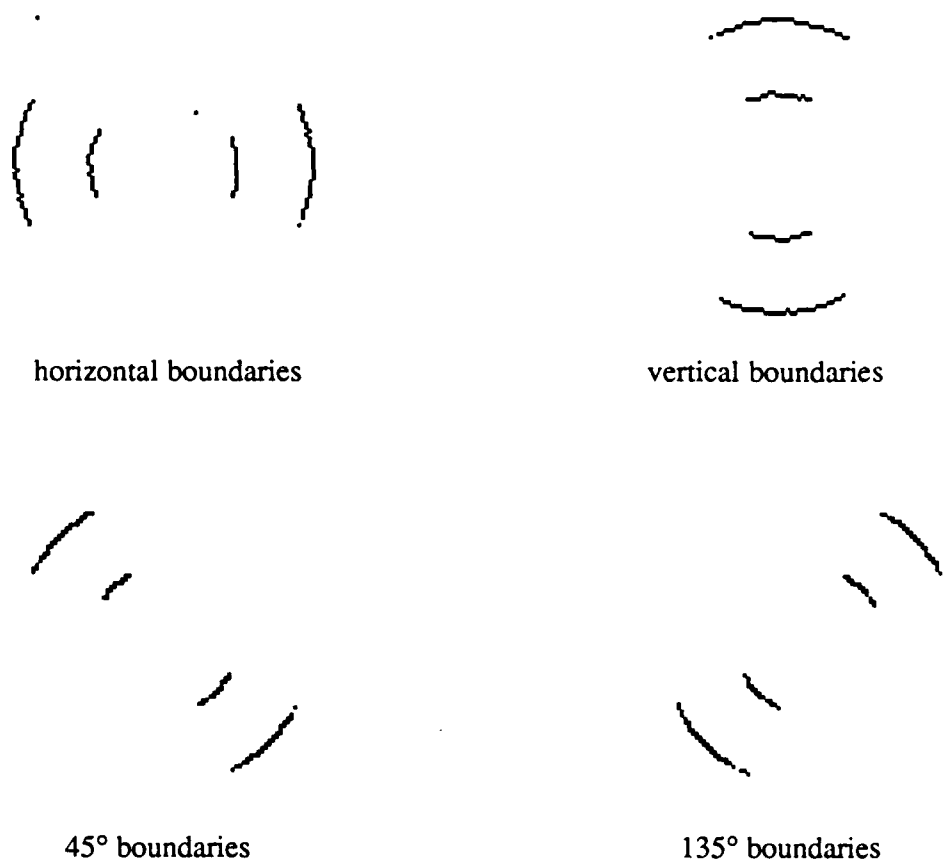


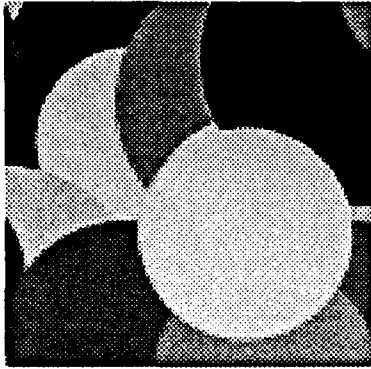
Figure 22 Oriented response for likelihood generator

5.1.2. Sensitivity to Noise

This section measures the response of the likelihood generator to expected and unexpected levels of noise.

Figures 23, 24, and 25 applies the 5x5, 7x7 and 9x9 operators tuned to standard deviation 12 noise to image 18 with too little, just right and too much noise respectively. The 5x5 operator output is black when there is greater than 50%

probability of a boundary. Note that with too little noise the detector misses boundaries that are there. With too much noise the detector detects boundaries that are not there.



a: image with $\sigma=0$ noise



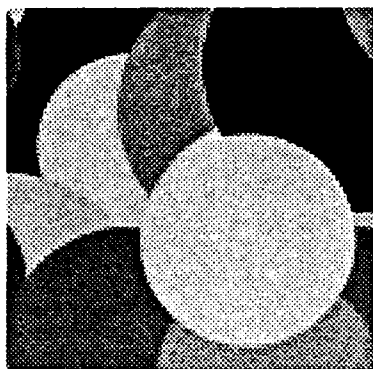
b: 5x5 operator



c: 7x7 operator



d: 9x9 operator



e: image with $\sigma=4$ noise



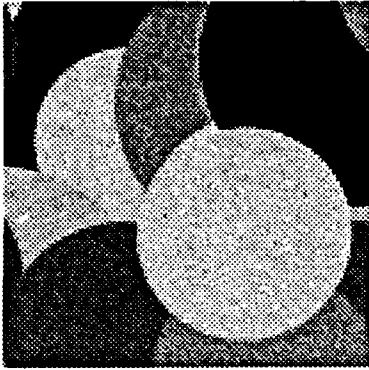
f: 5x5 operator



g: 7x7 operator



h: 9x9 operator



i: image with $\sigma=8$ noise



j: 5x5 operator

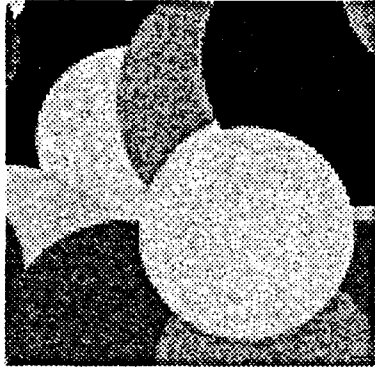


k: 7x7 operator



l: 9x9 operator

Figure 23 $\sigma=12$ operators applied to images with too little ($\sigma<12$) noise



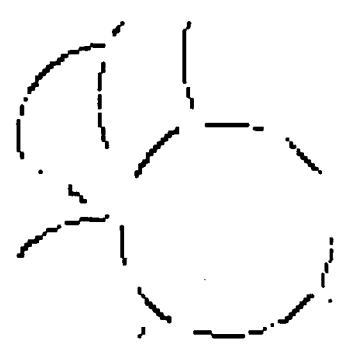
a: image with $\sigma=12$ noise



b: 5x5 operator

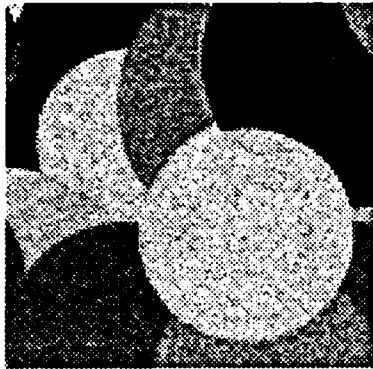


c: 7x7 operator



d: 9x9 operator

Figure 24 $\sigma=12$ operators applied to image with correct ($\sigma=12$) amount of noise



a: image with $\sigma=16$ noise



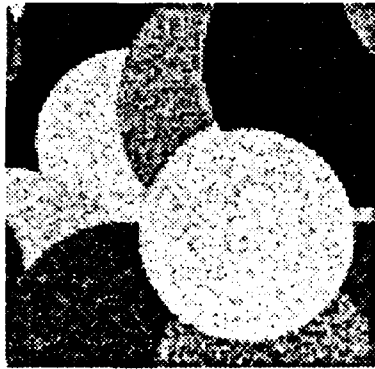
b: 5x5 operator



c: 7x7 operator



d: 9x9 operator



e: image with $\sigma=20$ noise



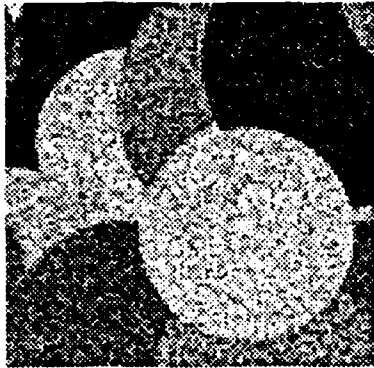
f: 5x5 operator



g: 7x7 operator



h: 9x9 operator



i: image with $\sigma=32$ noise



j: 5x5 operator



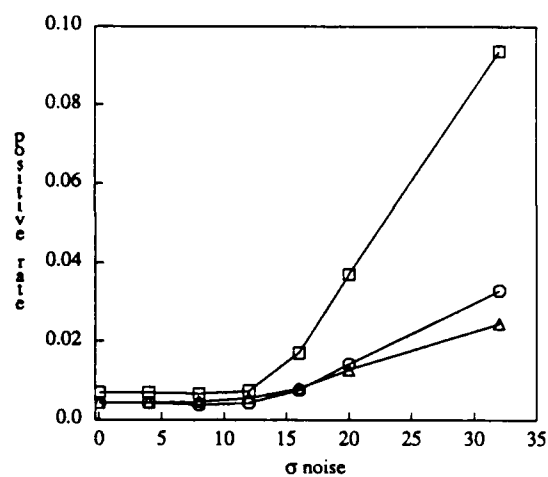
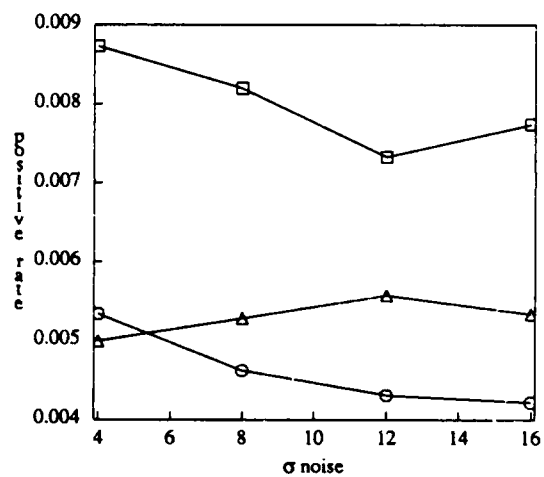
k: 7x7 operator



l: 9x9 operator

Figure 25 $\sigma=12$ 5x5 operator applied to images with too much ($\sigma>12$) noise

Error rates are available measuring the error rates of the operators when applied to the 3 images in figures 18, 19, and 20. Figure 26 show the false positive rates for the 5x5, 7x7 and 9x9 operators tuned to standard deviation 12 noise applied to all 3 artificial images. Here, error rates are charted as functions of increasing noise. Figure 27 charts the false negative rates for the operators applied to the images.

(a) $\sigma=12$ tuned operators

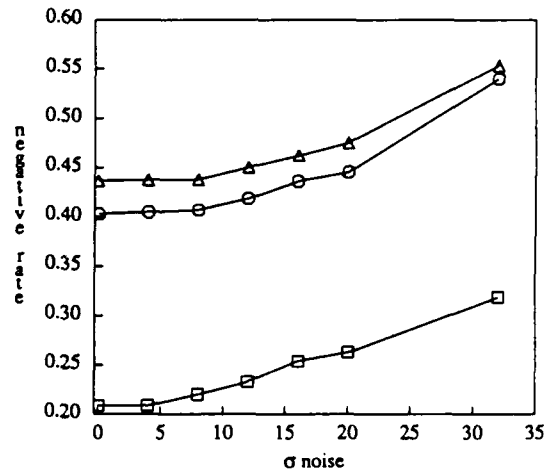
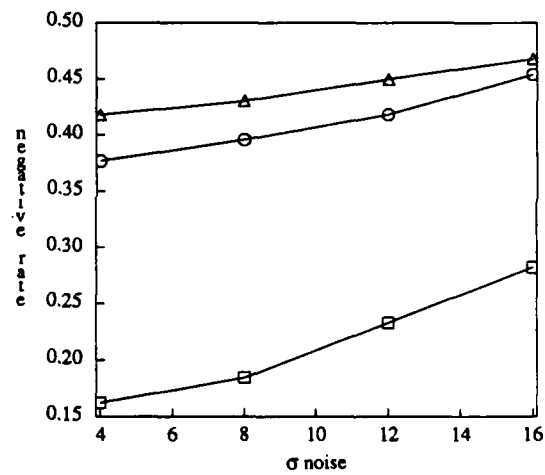
(b) operators tuned to the noise level in the image

squares 5x5 Operator

circles 7x7 Operator

triangles 9x9 Operator

Figure 26 False Positive Rate Charts for Operators Applied to Artificial Images

(a) $\sigma=12$ tuned operators

(b) operators tuned to the noise level in the image

squares 5x5 Operator

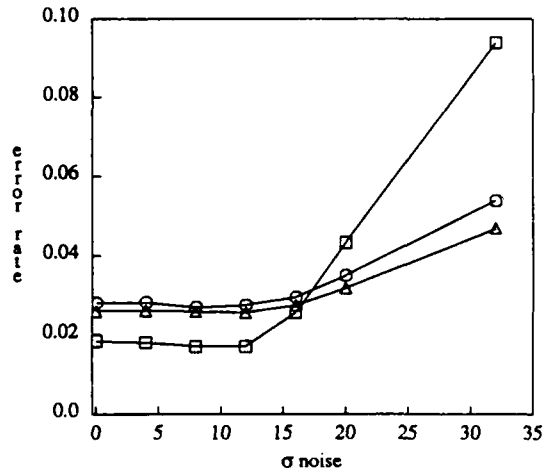
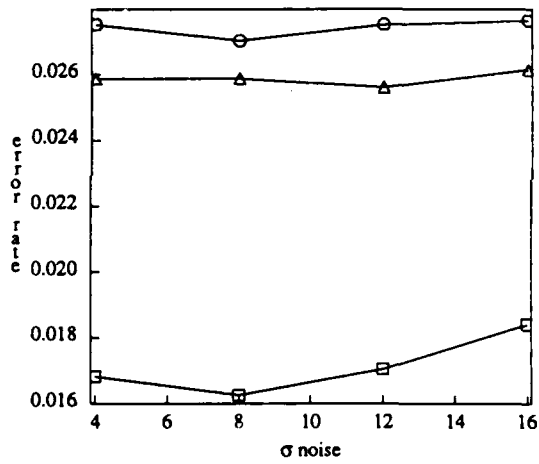
circles 7x7 Operator

triangles 9x9 Operator

Figure 27 False Negative Rate Charts for Operators Applied to Artificial Images

Figures 28, 29, and 30 charts the total errors for operators applied to the images from

figures 18, 19, and 20 respectively. Figure 31 charts the total errors from all the images.

(a) $\sigma=12$ tuned operators

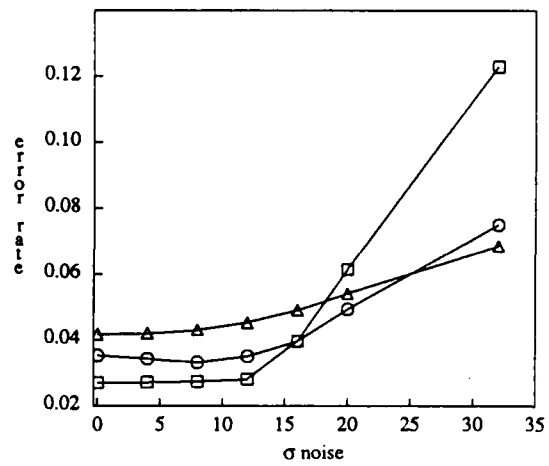
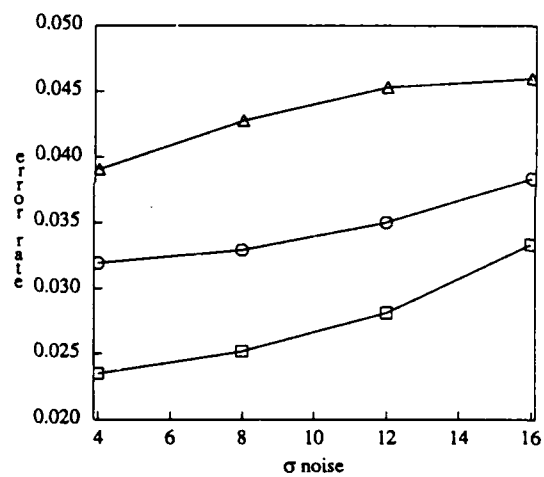
(b) operators tuned to the noise level in the image

squares 5x5 Operator

circles 7x7 Operator

triangles 9x9 Operator

Figure 28 Total Error Rate Charts for Operators Applied to Circles Image

(a) $\sigma=12$ tuned operators

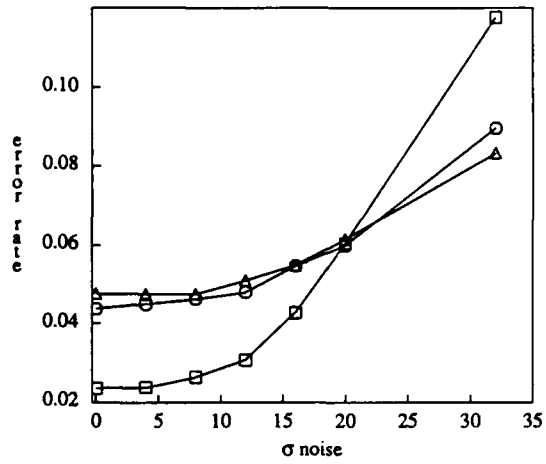
(b) operators tuned to the noise level in the image

squares 5x5 Operator

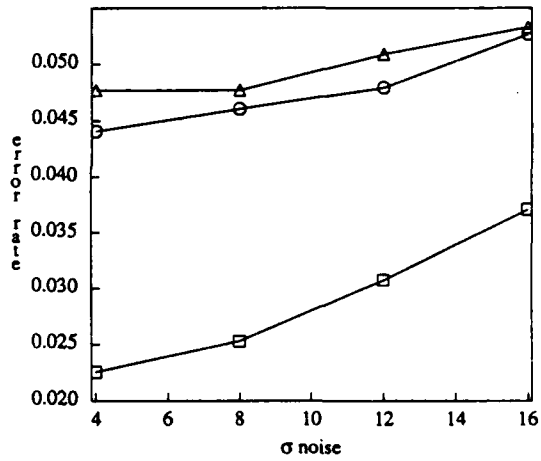
circles 7x7 Operator

triangles 9x9 Operator

Figure 29 Total Error Rate Charts for Operators Applied to Rectangles Image



(a) $\sigma=12$ tuned operators



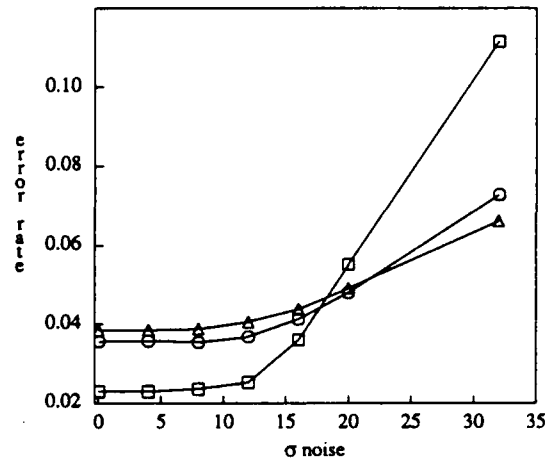
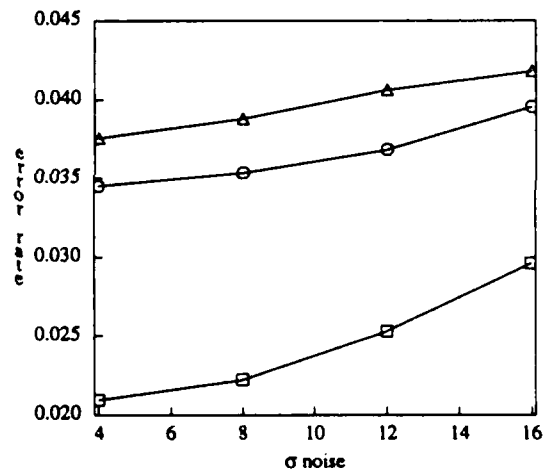
(b) operators tuned to the noise level in the image

squares 5x5 Operator

circles 7x7 Operator

triangles 9x9 Operator

Figure 30 Total Error Rate Charts for Operators Applied to Combined Image

(a) $\sigma=12$ tuned operators

(b) operators tuned to the noise level in the image

squares 5x5 Operator

circles 7x7 Operator

triangles 9x9 Operator

Figure 31 Total Error Rate Charts for Operators Applied to Artificial Images

Note that the 5x5 operator actually improves as the noise level grows in figure 23 b, f, and j. Thus the operators here are tuned to $\sigma=12$ noise and too little or too much noise reduces its accuracy. This effect is much less pronounced in the larger operators because they are much less sensitive to noise. The statistics charted in figure 28a also show that the $\sigma=12$ operator has minimal error at $\sigma=12$ noise (at least for the 5x5 and 7x7 operators).

However figure 31a does not have the error rate for the $\sigma=12$ operator minimized at $\sigma=12$. There are two effects competing here. The operator expects $\sigma=12$ noise. However images with less noise are easier to interpret even for tuned operators. In the circles and rectangles images there are a large boundaries between objects with close gray-levels, thus the tuning effect dominated and the error rate was minimized for $\sigma=12$. In the combined image there are few weak edges so the error rate was minimized for the minimal noise image. Thus for images with large boundaries between objects with close gray-levels the tuning of these operators is important. For images without such the operator's errors increase monotonically with the noise level.

5.2. Results with Laboratory Images

Artificial images are fine for taking statistics. However the statistics computed from artificial images are relevant to the operation of the operator in so far that the artificial images accurately reflect properties of real images. For example artificial images popularly used to test edge detectors often have objects with few evenly spaced gray-levels in them (Lunscher and Beddoes 1986b) (Lunscher and Beddoes 1986a). As an example a common artificial image for testing edge detectors is a checkerboard with noise added to it (Zhou, Chellappa, and Venkateswar 1986). In this image objects have only two gray-levels. All the rest of the gray-levels result from noise. Often operators that work for such images fail

when presented with the multiple reflectances available in real imagery (Pavlidis 1987).

While the artificial images used here do not suffer from that problem they do lack other characteristics of natural imagery such as smooth variance in intensity before noise. The likelihood generators presented here have also been applied to images taken by cameras in a laboratory environment.

Such images are more realistic but there are problems using them. Finding the positions of the boundary pixels in the image can not be done automatically. Thus the image and the thresholded outputs of operators is presented here. The reader must decide for himself about the efficacy of the operators.

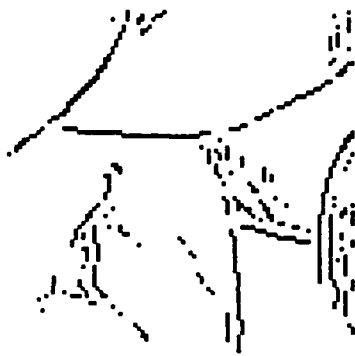
The two laboratory images I am using are shown in figures 32a and 33a. These figures are built of Play Doh™, a children's modeling clay. The main advantage of Play Doh™ is that there are no highlights on Play Doh™ figures and that Play Doh™ comes in many colors and can be molded to any required shape. These images were taken with a MTI Series 68 vidicon camera. This camera has been found to add approximately Gaussian noise with standard deviation between 2 and 6 to images it takes (Sanchis 1986).

Note that there are characteristics to these images that the artificial image lacks. The shading of objects in these images varies smoothly. The objects have dark marks on them. Thus this picture is difficult for a boundary pixel detector. The gnome picture is a bit blurred too.

Figure 32 shows the result of applying the standard deviation 4, 5x5, 7x7 and 9x9 likelihood generators to the blobs image. Figure 33 shows the result of applying the standard deviation 4, 5x5, 7x7 and 9x9 likelihood generators to the gnome image.



a: Image



b: 5x5 operator



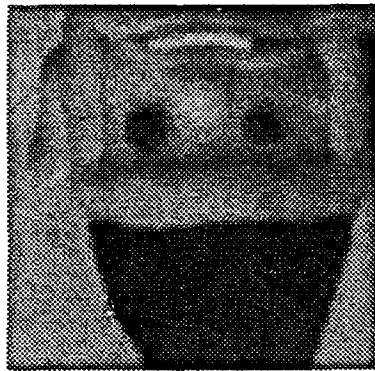
c: 7x7 operator



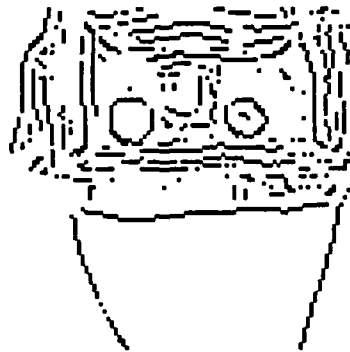
d: 9x9 operator

Figure 32 Application of Likelihood Generators to Play Doh™ Blobs

Note that the operators respond to the shadow boundary. Since these are occlusion boundary detectors these responses are false positives. The model for this operator assumes that the lighting varies smoothly across the image. This assumption is manifestly false at a shadow. A more sophisticated operator can be constructed by combining these detectors with a detector for shadow boundaries using the theory in section 5. Then combination of the shadow boundary detector and the occlusion boundary detector would be able to detect and label both shadow and occlusion boundaries.



a: Image



b: 5x5 operator



c: 7x7 operator



d: 9x9 operator

Figure 33 Application of Likelihood Generators to Play Doh™ Gnome

5.3. Results with Uncontrolled Images

One could object that the images of section 3.5.2 are images taken under controlled circumstances. Because the substances and lighting conditions of a laboratory image are highly controlled the results of these tests are not reliable indicators of the effectiveness of these techniques under uncontrolled circumstances. Two images taken under uncontrolled circumstances are presented in figures 34 and 36a. Figure 34 is an image of my friend Carlos Calderon. However his image requires too much memory in the current implementation of the likelihood generator.

Hence, the operators are applied to his ear (figure 35a). There is no control over the substances my friend is made of or his reflectance functions. Figure 36a is an aerial image of a sewage treatment plant.



Figure 34 Image of Carlos Calderon

Figure 35 shows the result of applying the standard deviation 4, 5x5, 7x7 and 9x9 likelihood generators to Carlos' ear. Figure 36 shows the result of applying the standard deviation 4, 5x5, 7x7 and 9x9 likelihood generators to the sewage treatment plant.



a: Image



b: 5x5 operator



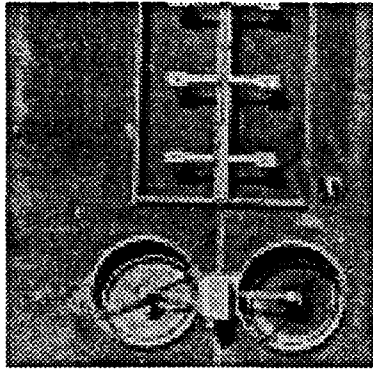
c: 7x7 operator



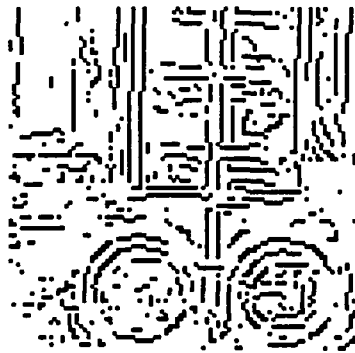
e: 9x9 operator

Figure 35 Application of Likelihood Generators to Carlos' ear

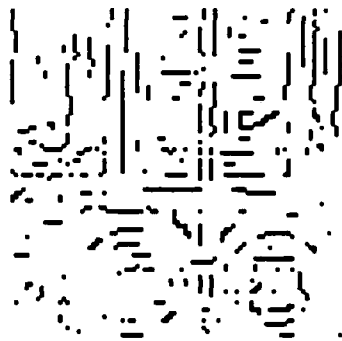
The texture on Carlos' face is too difficult for these operators and they fail. An operator that expects correlated noise might succeed on these images. However the operators did pick out the boundaries that fit their models well, such as his shirt and the edge of his hair. The aerial image in figure 36 is a better fit for the step edge model and the results from using these operators are better.



a: Image



b: 5x5 operator



c: 7x7 operator



d: 9x9 operator

Figure 36 Application of Likelihood Generators to the Sewage Treatment Plant

5.4. Comparisons with Established Techniques

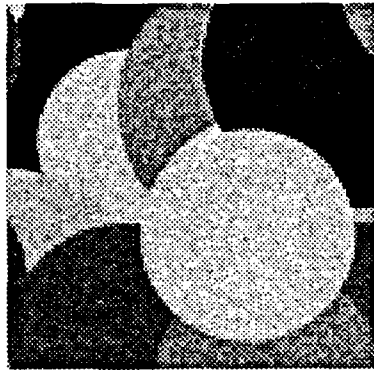
This section shows how the results from using a state of the art edge detector and a detector designed in the early 70's compare to the results in sections 3.5.1, 3.5.2 and 3.5.3. The early 70's edge detector is the Sobel edge detector (Ballard and Brown 1982a). This edge detector is used when an easily coded fast edge detector is required for a low-level vision system. The state of the art edge detector used here was designed by V. Nalwa (Nalwa and Binford 1986). It has a reputation as being the best general purpose edge detector extant (Feldman 1987).

Thresholding the Sobel operator at 220 minimizes the error rate of the Sobel operator on the image in figure 18. Thus the results presented here are with the Sobel operator thresholded at 220.

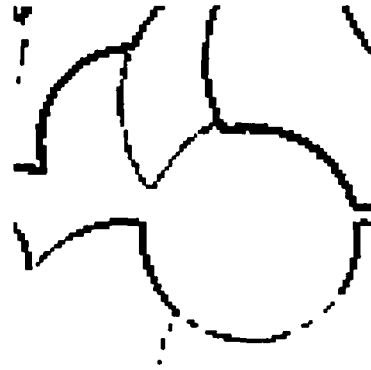
Code for Nalwa's operator was ported to the University of Rochester from Stanford University. It was found that the errors Nalwa's operator made were minimized when only edges less than .3 from the center of the window were accepted and the coefficient of tanh was thresholded at 1.2 the standard deviation of the noise. For a complete description of what these thresholds mean see Nalwa and Binfords PAMI paper (Nalwa and Binford 1986). The results presented here follow this regime.

5.4.1. Results For Artificial Images

Figure 37 is the thresholded output of the Sobel operator applied to the image from figure 18 with standard deviation 12 independent Gaussian additive noise. Figure 38 is the thresholded output of Nalwa's operator applied to the image from figure 18 with standard deviation 12 independent Gaussian additive noise.

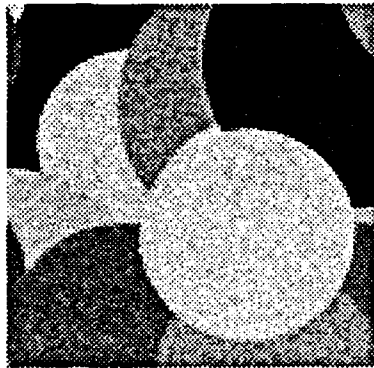


a: Circles Image



b: Sobel Output

Figure 37 Application of Sobel to Artificial Images



a: Circles Image

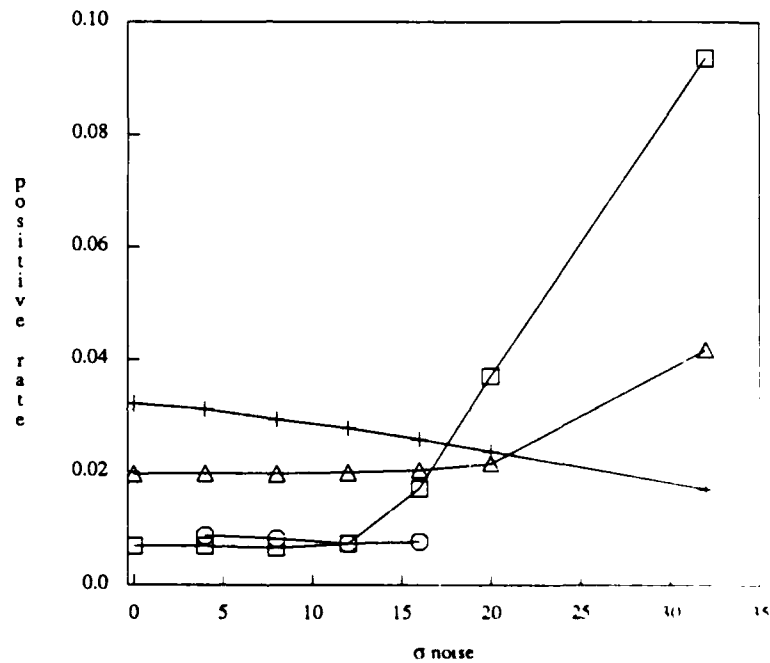


b: Nalwa Output

Figure 38 Application of Sobel to Artificial Images

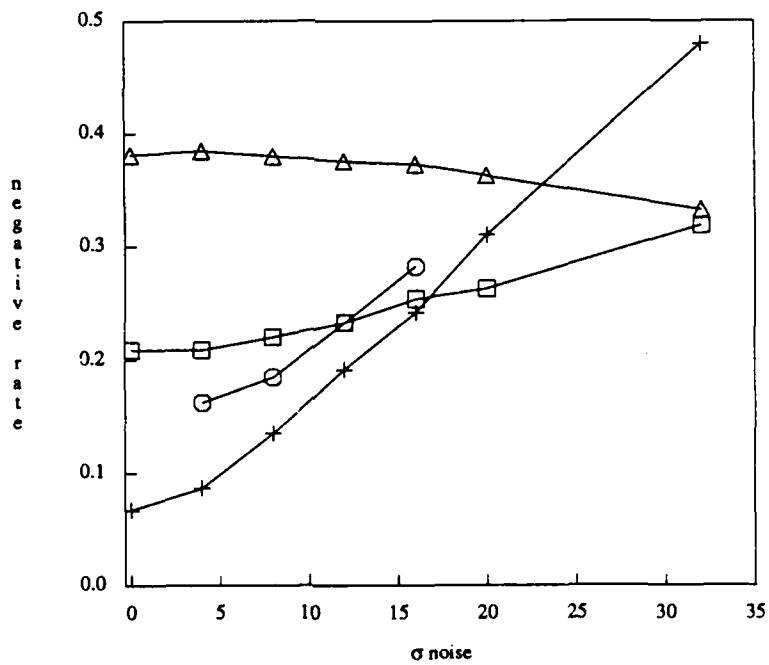
Neither Sobel's nor Nalwa's operator finds the low contrast boundary on the left side of the image. However neither find many false positives away from the boundaries. Sobel's operator returns thick boundaries that cause false positives. Nalwa's operator does not have this problem though. Nalwa's operator handles corners well too.

Figure 39 compares the false positive rates for the Sobel operator, Nalwa's operator and the 5x5 likelihood generator on the artificial images. Figure 40 compares the false negative rates for the Sobel operator, Nalwa's operator and the 5x5 likelihood generator on the artificial images. Figure 41 compares the total error rates for the Sobel operator, Nalwa's operator and the 5x5 likelihood generator on the artificial images.



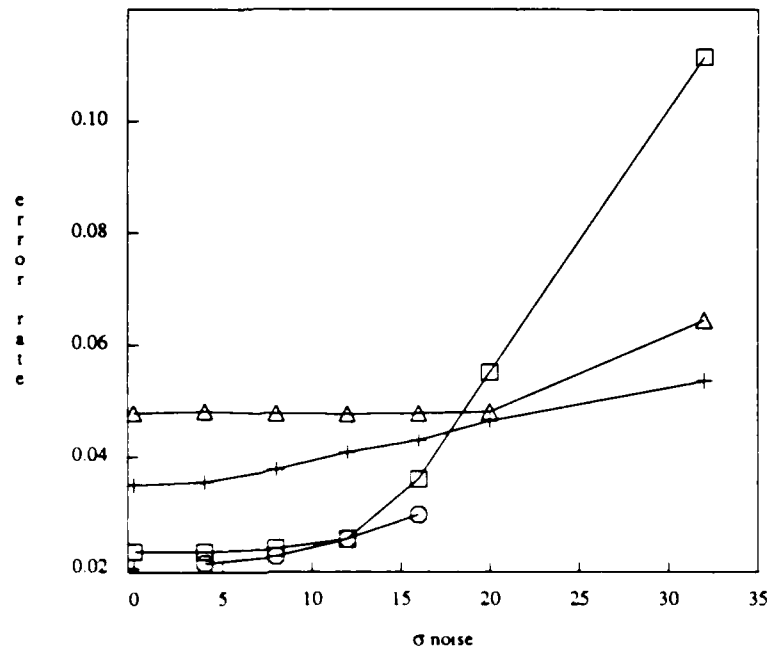
Squares $\sigma=12$ tuned operator
 Circles operator tuned to noise level
 Triangles Sobel operator
 Crosses Nalwa's operator

Figure 39 False Positive Rate Comparison Chart



Squares $\sigma=12$ tuned operator
 Circles operator tuned to noise level
 Triangles Sobel operator
 Crosses Nalwa's operator

Figure 40 False Negative Rate Comparison Chart



Squares $\sigma=12$ tuned operator
 Circles operator tuned to noise level
 Triangles Sobel operator
 Crosses Nalwa's operator

Figure 41 Total Error Rate Comparison Chart

Nalwa's operator has a higher error rate than the 5x5 likelihood generator. But the artificial images used to test the operators fit the model of the likelihood generators better than Nalwa's model. Also the error cost used fits the model of the likelihood generators. Hence these statistics are biased for the likelihood generators. However the bias is not great since the images here are fairly realistic and his operator is tuned to a facet model. Hence these statistics are evidence that the boundary detectors developed here are on par with the best extant edge detector, Nalwa's operator. Likelihood generators can be developed for Nalwa's model and

AD-A189 041

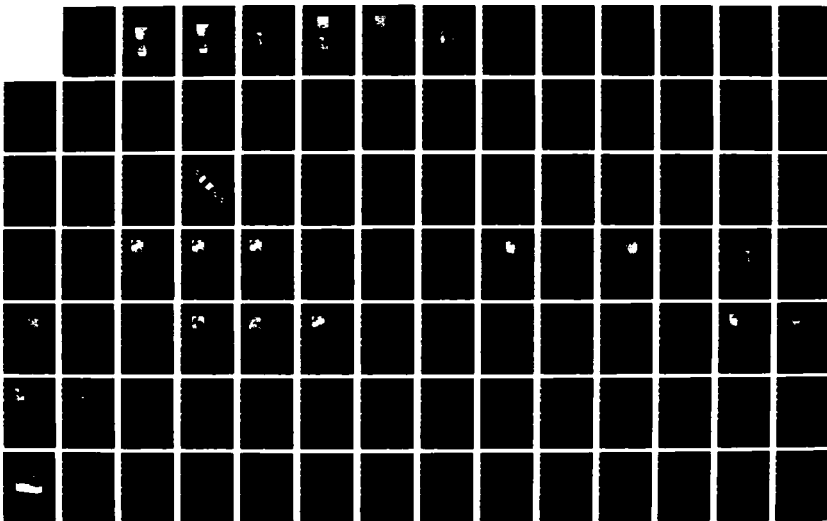
A PROBABILISTIC APPROACH TO LOW-LEVEL VISION (U)
ROCHESTER UNIV NY DEPT OF COMPUTER SCIENCE D B SNER
OCT 87 TR-232 N00014-82-K-0193

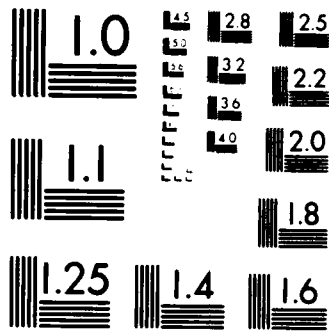
273

UNCLASSIFIED

F/G 23/3

NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963-A

the resulting operator would do better than Nalwa's on images that fit his model.

5.4.2. Results For Laboratory Images and Real Images

Figures 42 and 43 is the output of the thresholded Sobel for the Blob and Gnome image respectively. Figures 44 and 45 is the output of Nalwa's operator thresholded for standard deviation 4 noise on the Blob and Gnome image respectively.



a: Image

b: Sobel Output

Figure 42 Application of the Sobel to the Play Doh™ Blob Image

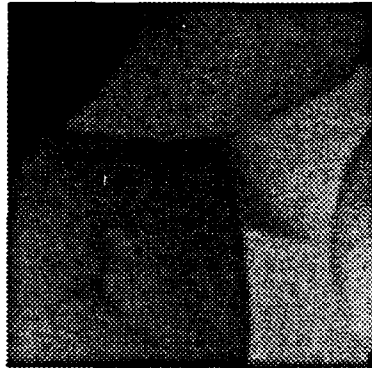


a: Image

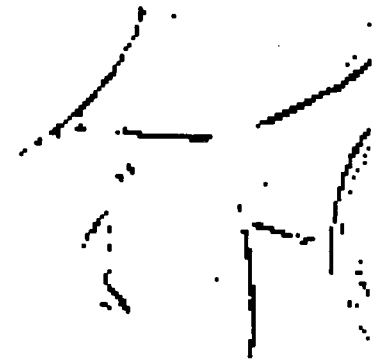
b: Sobel Output

Figure 43 Application of the Sobel to the Play Doh™ Gnome Image

These images are too dim for the Sobel to be thresholded at 220. Thus Sobel's operator only responds to the strongest gradients. The sensitivity of Sobel's operator the brightness of the image is one of the problems using it. However Nalwa's operator has better results.

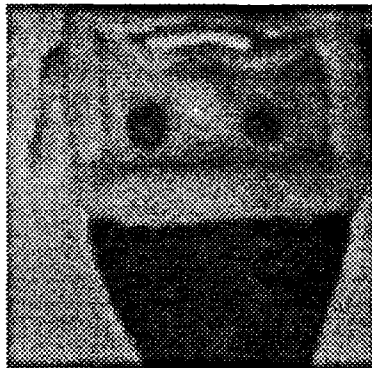


a: Image



b: Nalwa Output

Figure 44 Application of Nalwa's Operator to the Play Doh™ Blob Image



a: Image



b: Nalwa Output

Figure 45 Application of Nalwa's Operator to the Play Doh™ Gnome Image

Nalwa's operator does very well indeed on these images. Nalwa's operator has a higher degree facet model than the likelihood generators tested. Nalwa's model is a

better fit for these clay images than the step edge model. Therefore naturally Nalwa's operator returns superior results. However a likelihood generator can be supplied for his model and such an operator would do even better on these images. A likelihood generator can also be tuned to boundaries between Lambertian surfaces. Such an operator would be superior to any extant edge detector on Lambertian imagery.

Figures 46 and 47 is the output of the thresholded Sobel for Carlos' ear's and the sewage plant's image respectively. Figures 48 and 49 is the output of Nalwa's operator thresholded for standard deviation 4 noise on Carlos' ear's and the sewage plant's image respectively.



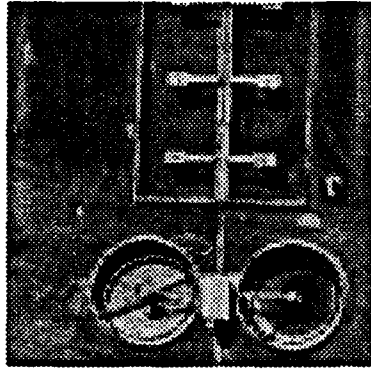
a: Image



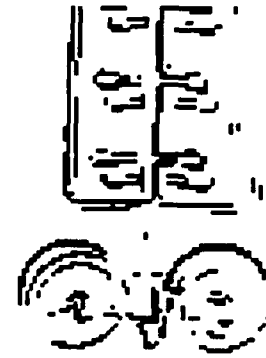
b: Sobel Output

Figure 46 Application of the Sobel to Carlos' ear

In some sense the Sobel does best on this image by ignoring the irrelevant texture in Carlos' face. Thus every operator has an image where it does well.



a: Image



b: Sobel Output

Figure 47 Application of the Sobel to the Sewage Treatment Plant

Here again Sobel's operator does not respond to enough boundaries. Those boundaries it responds to, it responds to too many times.

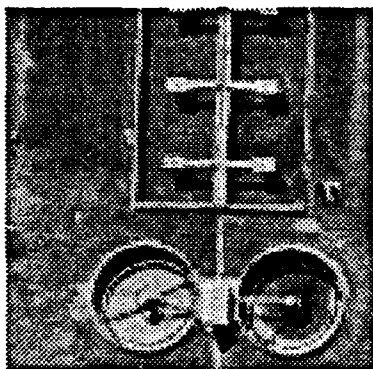


a: Image

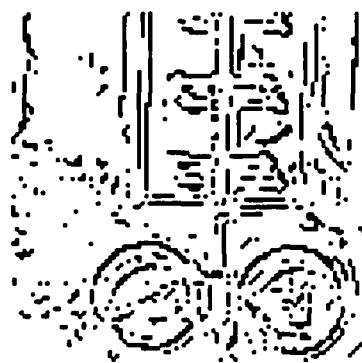


b: Nalwa Output

Figure 48 Application of Nalwa's Operator to the Carlos' ear



a: Image

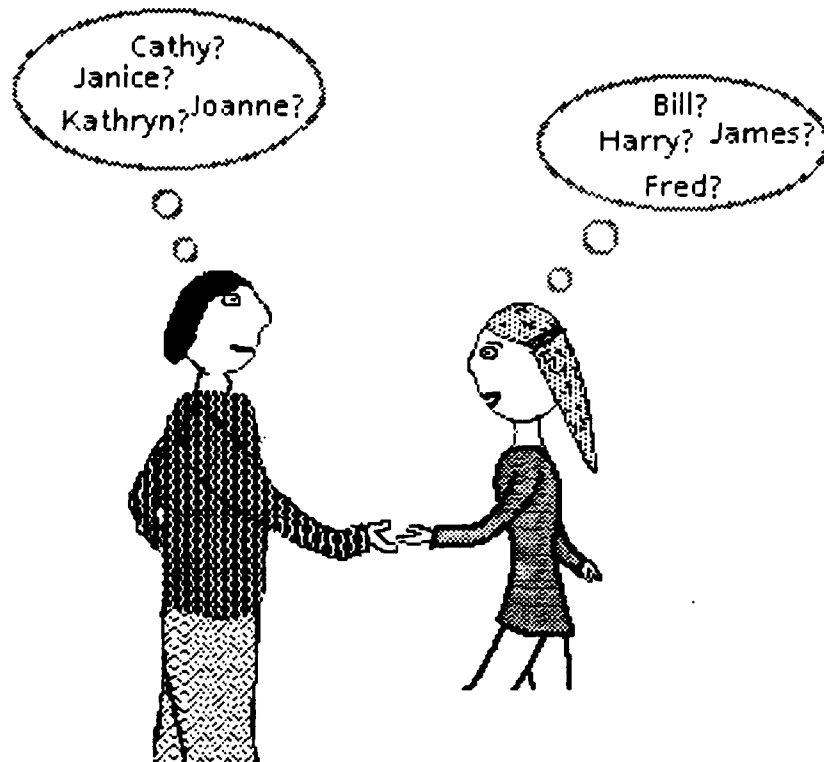


b: Nalwa Output

Figure 49 Application of Nalwa's Operator to the Sewage Treatment Plant

On the aerial image Nalwa's operator's output is similar to that of the 5x5 likelihood generator. Nalwa's operator may be slightly superior on figure 49 though it is difficult to judge. Its superiority can be attributed to its superior model.

Chapter 4



John and Sue Meet Again

Template Matching

CHAPTER 4

Template Matching

This chapter yields a probabilistic interpretation for the medium level vision task of template matching. It is assumed here that template matching is being used for recognizing objects, though this analysis may be used for other types of template matching. Template matching receives input from a low level feature detector such as an edge detector. This chapter assumes that the input is binary. Thus for the set of points in the visual field features are either present or absent at each point. A template matcher looks at the subset of the visual field where it expects the features to be present. It outputs how many of the features actually are present. If the object represented by the template is present in the image the template matcher should return a high output.

Consider a template for a square in the middle of the visual field. Such a template is in figure 50.

0	0	0	0	0	0
0	1	1	1	1	0
0	1	0	0	1	0
0	1	0	0	1	0
0	1	1	1	1	0
0	0	0	0	0	0

Figure 50 Square Template

The output of the feature detector could be figure 51.

1	1	1	1	0	0
1	0	0	<i>1</i>	<i>1</i>	0
1	0	1	0	<i>1</i>	0
1	0	1	0	<i>1</i>	0
1	<i>1</i>	<i>1</i>	0	<i>1</i>	0
1	1	0	0	0	0

Figure 51 Output of Feature Detector

The pixels of figure 51 that are matched by the template of figure 50 are in *italics*. The total number of matches is 7. Section 4.2 shows how to take an output like 7 and derive a probability.

This definition for template matching is a simplified version of the commonly used definition. A more sophisticated version of template matching and some algorithms to implement it are described in (Sher 1985b). Section 4.4 extends the work here to more complex forms of template matching.

A common implementation of this kind of template matching uses the Hough transformation. However other techniques such as convolution with the fast Fourier transformation are available and sometimes more efficient (Sher 1985b). The implementation of the template matcher is irrelevant to this chapter.

Often some parts of a template are considered more important than others. For example often edge detectors become unreliable at corners. Thus the results of matching the corners of the square in figure 50 should be less important than the results of matching the sides. One can represent the difference in importance by weighting the template. Consider figure 52. Here the members of the sides have a weight of 4 while the corner members have a weight of 1. Thus each side member contributes 4 times as much as each corner member in computing the total match.

0	0	0	0	0	0
0	1	4	4	1	0
0	4	0	0	4	0
0	4	0	0	4	0
0	1	4	4	1	0
0	0	0	0	0	0

Figure 52 Weighted Template

When matching with the template in figure 52 a side matching (such as in position (2,3) of figure 51) adds 4 to the total while a corner matching (such as in position (2,2) of figure 52) adds 1 to the total. Weighted matching was used by Brown and Sher (Brown and Sher 1982a) (Brown, Curtis, and Sher 1983a) to improve the performance of Hough transform line finders. Clearly unweighted template matching is a special case of weighted template matching.

1. Models for Template Matching

The models described in sections 3.1 and 2.3 yield prior probabilities for the desired features, namely boundary pixels, and likelihoods for the observed image given the presence of this feature. The observed data in template matching is the output of the low-level feature detector. The feature being labeled by the template matcher is the presence of an object at a particular point p in the image and the labels of the feature are 1 - *present* and 2 - *absent*. Thus the model must supply prior probabilities for objects being present or absent at the points in the image, and the probabilities of observed output from a low-level feature detector given the presence or absence of objects at positions in space.

Given positional isotropy the prior probability of an object being at a position is the expected number of objects in the image divided by the number of positions. The expected number of objects in an image is a primitive statistic easily elicited from a user. Thus given the simplifying assumption of positional isotropy, the prior

probability of an object being at a position can be derived from a primitive statistic.

When object O is present at p , at each element of the template $T(O,p)$ there should be a 1 in the feature detector output. If the input to the template matcher has a 0 at $T(O,p)[i]$ when O is present at p then the input has an error. Also, if O is absent from p , $T(O,p)[i]$ should be 0 or else it is an error. Figure 53 illustrates when an error occurs in the input to the template matcher.

1	1	1	1
1	0	0	1
1	0	0	1
1	1	1	1

TEMPLATE

1	0	1	0
0	1	0	1
1	0	1	0
0	1	0	1

errors if object present

1	0	1	0
0	1	0	1
1	0	1	0
0	1	0	1

errors if object absent


 placed around errors

Figure 53 Errors in Input to a Template Matcher

The likelihood of an observed input from the low-level feature detector given an

object is present is the probability that the misses in the template match would occur. The likelihood of an input from the low-level feature detector given the object is absent is the probability that the erroneous hits would occur.

1.1. Independence Assumption

To simplify the mathematics, errors in the input are assumed to be independent. This assumption is relaxed in section 4.6. From this independence assumption section 4.2 derives likelihoods of objects being present and absent as a function of weighted template matches.

If errors in the input are correlated between pixels then taking a pixel-wise sum (as template matching does) is bound to lose information because it is not clear whether several highly correlated pixels caused the N matches or several uncorrelated pixels caused the N matches. Several highly correlated pixels should not count as highly as several uncorrelated pixels. (To understand why correlated pixels should count for less consider the case of two perfectly correlated pixels. When ever one reports a feature so does the other. There is only one pixel worth of information in this pair.) Thus a template match is only sufficient if errors are independent.

For example consider a two pixel template. It contains position one and two as in figure 54.



Figure 54 Two adjacent pixels

The independence condition applied in this chapter is if O is true (the object is present) the presence of a 1 (in the input) at P1 is independent of the presence of a 1 at P2. Thus if the probability of 1 at P1 is p_1 and the probability of 1 at P2 is p_2 the probability that both are 1 is p_1p_2 . When O is false the same condition applies. This

property is sometimes called conditional independence and is a common assumption for research in evidence combination for vision (Bolles 1976) (Pearl 1985b).

Thus this chapter assumes that errors are uncorrelated between pixels. However this assumption is rarely satisfied since most feature detectors' behavior is locally highly correlated. Future work will concentrate on developing algorithms for object detection that take these correlations into account.

1.2. Parameters of the Model

Given conditional independence the feature detector (for example edge detector) can be modeled separately at each pixel. A template matcher is trying to determine if object O is present at position p in an image. If O is present the feature detector should return 1 at the pixels being matched. Two reasons for the feature detector to return 0 are a *false negative* reported by the feature detector (the feature (eg. edge) is present in the image but because of noise it was not detected) and *occlusion* (the feature is not present in the image because of an occluding object). When the object is not present the feature detector should return 0 at the matched pixels. There are two reasons for it to return 1. It could return a *false positive* (the feature detector was deceived into reporting a feature by noise) and an *extraneous detection* (another object could present the feature to the detector even though O_0 is absent).

Section 4.2 calculates likelihoods for the object being present and absent from this model. Thus if the user supplies the expected number of objects in the image and the probabilities of false positives, false negatives, occlusions, and extraneous detections then she has supplied enough primitive statistics to generate an object detector. Reasoning similar to that of section 2.3 can be used to calculate occlusion and extraneous object rates from the expected digitized areas and perimeters of objects. False positive rates and false negative rates are functions of the input feature detector supplied by the user.

There has been some work studying the effect of other sources of errors. (Shapiro and Iannino 1979) studied the effect of quantization noise on line detection through template matching. (Maitre 1986) studied the effect of uniformly distributed false positives on line detection and showed that the effect is not uniform over the output array.

2. Likelihoods for Objects

Consider, on a pixel by pixel basis, the likelihood of the input pixels when the object is present. Call the probability of a false negative at pixel i , p_i^n and the probability of an occlusion, p_i^o . Call the total probability that pixel i is 0 when it is supposed to be 1, $\beta_i = p_i^n + p_i^o$. Let S_1 be the set of pixels matched by the template that are 1, and S_0 be the set of pixels matched by the template that are 0. The likelihood of the input given O is described by expression 14.

$$\left[\prod_{i \in S_0} (\beta_i) \right] \left[\prod_{i \in S_1} (1 - \beta_i) \right] \quad (14)$$

Similarly the likelihood of the object being absent is calculated from the probability of a false positive at pixel i , p_i^p and the probability of an extraneous event at i , p_i^e in expression 15. Call the total probability that pixel i is 1 when it is supposed to be 0 $\alpha_i = p_i^p + p_i^e$.

$$\left[\prod_{i \in S_1} (\alpha_i) \right] \left[\prod_{i \in S_0} (1 - \alpha_i) \right] \quad (15)$$

Taking the logs of the two likelihoods thus getting a sum in equations 16 and 17 shows that these likelihoods are functions of a template matching operation. (S is the set of all pixels matched by the template.)

$$\begin{aligned} & \left[\sum_{i \in S_0} \ln(\beta_i) \right] + \left[\sum_{i \in S_1} \ln(1-\beta_i) \right] \\ &= \left[\sum_{i \in S_1} \ln(1-\beta_i) - \ln(\beta_i) \right] + \left[\sum_{i \in S} \ln(\beta_i) \right] \end{aligned} \quad (16)$$

$$\begin{aligned} & \left[\sum_{i \in S_1} \ln(\alpha_i) \right] + \left[\sum_{i \in S_0} \ln(1-\alpha_i) \right] \\ &= \left[\sum_{i \in S_1} \ln(\alpha_i) - \ln(1-\alpha_i) \right] + \left[\sum_{i \in S} \ln(1-\alpha_i) \right] \end{aligned} \quad (17)$$

Both equations 16 and 17 show that the likelihoods for the object being present and absent are functions of weighted template matches. Thus the application of two templates to an image one template having pixel i weighted with $\ln(1-\beta_i)-\ln(\beta_i)$ and the other template weighted with $\ln(\alpha_i)-\ln(1-\alpha_i)$ yields the likelihoods of the object being present and absent. In the special case that all the pixels have the same probabilities of false positives, negatives, occlusions, and extraneous detections, all the weights are the same and the likelihoods can be calculated directly from the simpler kind of template matching described at the beginning of this chapter.

For many purposes only the ratio of the two likelihoods is required. In particular the probability of the object being present (computed using Bayes' rule (equation 2)) is a function of the ratio of these two likelihoods. Also Markov random fields model algorithms use likelihood ratios. The likelihood ratio is a function of the difference of the two log likelihoods described in expression 16 and 17. Equation 18 computes it.

$$\left[\sum_{i \in S_1} \ln(1-\beta_i) - \ln(\beta_i) - \ln(\alpha_i) + \ln(1-\alpha_i) \right] \quad (18)$$

$$+ \left[\sum_{i \in S} \ln(\beta_i) - \ln(1-\alpha_i) \right]$$

This statistic can be calculated using a template whose weights are of the form in expression 19.

$$\ln(1-\beta_i) - \ln(\beta_i) - \ln(\alpha_i) + \ln(1-\alpha_i) \quad (19)$$

If all the probabilities are the same at each pixel then the weighted template matches reduces to an unweighted template match.

Call the log likelihood ratio computed in equation 18, L . The probability of the object being present is described by equation 20.

$$\frac{1}{1 + e^{-L} \frac{P(\neg O)}{P(O)}} \quad (20)$$

Thus given the independence assumption a template match with weights as in equation 19 yields the probability of an object being present. When the probabilities of errors are the same at each pixel one can calculate the probability of the presence of an object from an unweighted template match.

For example consider the case of the template in figure 50. An edge detector makes more errors at corners than other places. Otherwise all the pixels are the same. p^p and p^e are the same for all pixels since they concern the case where the object is absent. p^o is the same for all pixels since corners or sides should be equally subject to occlusion. Thus the only difference between the pixels of the template is that p^n is larger at corners than at sides. Because p^n is larger at the corners the weights assigned to the corners is smaller than the weights assigned to sides. In particular if the probability of occlusion is 0.1 the probability of extraneous edges is 0.2 the

probability of a false positive is 0.1 and the probability of a false negative is 0.1 at sides and 0.4 at corners the weight assigned to sides is 2.23 and the weight assigned to corners is 0.85, resulting in a template shown in figure 55

0	0	0	0	0	0
0	0.85	2.23	2.23	0.85	0
0	2.23	0	0	2.23	0
0	2.23	0	0	2.23	0
0	0.85	2.23	2.23	0.85	0
0	0	0	0	0	0

Figure 55 Computed Template

3. Thresholds

Thresholds are used to create the input and to analyze the output of template matching. The output of a template matcher is often thresholded¹ to make decisions about the presence or absence of objects. Determining such a threshold is discussed in section 4.3.1. The output of a feature detector is often thresholded to get an array of 1's and 0's. Finding the correct threshold to use in a feature detector is discussed in section 4.3.2.

3.1. Thresholding the Matcher

This section considers how to pick a threshold for a template matcher that minimizes the posterior cost of errors. This section describes a simple application of decision theory (Berger 1980a). This chapter assumes that there is zero cost to reporting the object is present when it is present and reporting the object is absent when it is absent. Let the cost of reporting an object when it is absent (a false positive for the matcher) be c_p , and the cost for failing to report a present object (a false

¹Every member of the output less than the threshold is set to 0. Every member of the output greater than the threshold is set to 1.

negative for the matcher) be c_n .

Assume that M is the set of matches from a template match. The expected cost of declaring the object is present is the probability of the object being absent and M matches $P(\neg O \& M)$ times the cost of a false positive c_p . The expected cost of declaring the absence of the object is the probability of the object being present $P(O \& M)$ multiplied by the cost of a false negative c_n . Thus one should report an object if the inequality in equation (21) is satisfied.

$$\frac{c_p}{c_n} < \frac{P(O \& M)}{P(\neg O \& M)} \quad (21)$$

The inequality of equation (21) is transformed into a likelihood ratio test of classical statistics in equation (22).

$$\frac{c_p P(\neg O)}{c_n P(O)} < \frac{P(M|O)}{P(M|\neg O)} \quad (22)$$

Thus by picking the correct threshold one can maximize the power of this algorithm for a specified size². However, the rest of this chapter takes a decision theoretic approach (minimizing posterior expected costs) suggested by (Berger 1980a).

Section 4.2 shows that the likelihood ratio is a monotonic function of a weighted template match with the weights chosen according to equation 19. Thus with N the result of the weighted match equation 23 substitutes in equation 18 for the likelihood ratio.

²This sentence uses the statistical definitions for size and power

$$\frac{c_p P(-O)}{c_n P(O)} < e^{N + \sum_{i \in S} \ln(\beta_i) - \ln(1-\alpha_i)} \quad (23)$$

$$\ln \left[\frac{c_p P(-O)}{c_n P(O)} \right] - \sum_{i \in S} \ln(\beta_i) - \ln(1-\alpha_i) < N$$

Thus there is a clear derivation of a threshold for a weighted template match. If the error probabilities are the same at each pixel then all the weights are the same for each pixel. Thus one can use unweighted template matching. Equation 24 is the threshold derived for the unweighted match.

$$\frac{\ln \left[\frac{c_p P(-O)}{c_n P(O)} \right] - |S| \left[\ln(\beta) - \ln(1-\alpha) \right]}{\ln(1-\beta) - \ln(\beta) - \ln(\alpha) + \ln(1-\alpha)} \quad (24)$$

If more matches than equation 24 are found then the expected cost is less when reporting an object present, otherwise the expected cost is minimized by reporting the object is absent.

3.2. Thresholding the Detector

Only unweighted template matching is considered in this section though the results generalize easily enough to weighted template matching.

Often a feature detector is a two step algorithm. The first step is to calculate a confidence or strength that the feature is present at each point considered. The second step thresholds the confidences to make binary decisions at each point. The positional isotropy assumption yields that the threshold is the same for each pixel. The p^p and p^n are functions of this threshold t . This section assumes the functions $p^p(t)$ and $p^n(t)$ are known and from this calculates a formula for optimal selection of t .

Since t must be chosen before running the matcher, in this section the prior expected cost is minimized over choices of t . The prior expected cost is described in equation (25).

$$P(N \leq T(t) \& \neg O)c_p + P(N > T(t) \& O)c_n \quad (25)$$

Here $T(t)$ is the optimal threshold derived in section 4.3.1. Equation 25 is a function of $p^p(t)$ and $p^n(t)$. Thus the prior cost of using a threshold t on the matcher depends only on the false positive and negative rates of the matcher as a function of t .

Thus all the terms of equation 25 can be derived in terms of t . Because $p^p(t)$ and $p^n(t)$ and $T(t)$ are all monotonic in t the prior cost as a function of t has a unique local minimum. This minimum can be found using binary search or Newtons' method to a sufficient precision. At the moment the code to determine the prior cost minimizing threshold is under construction.

Measuring $p^p(t)$ and $p^n(t)$ for detectors is a difficult task. However code to take statistics on these functions and approximate them and their derivatives is available for boundary point detectors. Results from using some of this code appears in (Sher 1987a) and sections 3.5 and 5.8.

4. Multiple Label Feature Detectors

The results of sections 4.2 and 4.3 apply to a restricted form of template matching described at the beginning of this chapter. However more sophisticated problems can be transformed into equivalent problems in this restricted form.

The beginning of this chapter assumes that the feature detector returns binary decisions about the appearance or absence of a feature. Thus an edge detector returns 1 or 0 depending on whether it thinks there is an edge present. Often feature detectors would return a variety of labels. An edge detector often returns an orientation, an optical flow detector usually returns a velocity. Formally such

detectors return one label l of a set of labels L or possibly 0 (no label) at each point.

Consider a feature detector that has $L = \leftarrow, \downarrow, \rightarrow, \uparrow$. A template matcher that accepts input from such a detector has a template that looks like figure 56.

0	0	0	0	0	0
0	\leftarrow	\uparrow	\uparrow	\uparrow	0
0	\leftarrow	0	0	\rightarrow	0
0	\leftarrow	0	0	\rightarrow	0
0	\downarrow	\downarrow	\downarrow	\rightarrow	0
0	0	0	0	0	0

Figure 56 Template with many Labels

The input to the template matcher can look like figure 57.

\leftarrow	\uparrow	\uparrow	\uparrow	0	0
\leftarrow	0	0	\rightarrow	\uparrow	0
\leftarrow	0	\leftarrow	0	\leftarrow	0
\leftarrow	0	\rightarrow	0	\rightarrow	0
\leftarrow	\rightarrow	\downarrow	0	\rightarrow	0
\leftarrow	\rightarrow	0	0	0	0

Figure 57 Output of Multiple Label Feature Detector

A template matcher counts the number of non zero points in the template where the label in the template matches the corresponding label in the feature detector output. For figures 56 and 57 the result of template matching is 4. This type of template matching is described in (Sher 1985b).

There is a transformation of the problem of template matching with multiple labels to template matching with a single label. Thus for every multiple label template matching problem there is a single label template that when matched to a translated input returns the same number.

Assume there are $|L|$ labels. Then replace each entry of the template with $|L|$ entries. Each of the $|L|$ entries corresponds to an element of L . If the entry of the

template is labeled l the entry corresponding to l in the new template is set to 1 and the other entries are set to 0. Table 58 describes how each label is translated into a set of entries for the example above and the third column of the template of figure 57 is translated into a table in figure 59.

label	set of entries			
0	0	0	0	0
↑	1	0	0	0
↓	0	1	0	0
→	0	0	1	0
←	0	0	0	1

Figure 58 Correspondence of Labels to Rows

3rd Column	Translation			
↑	0	0	0	1
0	0	0	0	0
↑	1	0	0	0
↓	0	0	1	0
↓	0	1	0	0
0	0	0	0	0

Figure 59 Translation of Third Column of figure 57

If this translation is applied to both the template and the output of the feature detector then applying template matching for 1's and 0's on the translated output has the same result as the template match with multiple labels.

5. Feature Detectors that Return Probabilities

This dissertation motivates feature detectors that output probabilities rather than feature labels. However the analysis here assumes that feature labels are output by the low-level detector. This section analyzes using the probability distribution output by a low-level vision operator that returns probabilities.

A possibility is to report as the output of the template matcher the expected weighted sum of votes. Summing the probabilities at each point to be matched yields the expected weighted sum of votes. However, it is not clear that the analyses in sections 4.2 and 4.3 are still correct when using expected weighted sums of votes.

For an example using the template of figure 50 (Reprised).

0	0	0	0	0	0
0	1	1	1	1	0
0	1	0	0	1	0
0	1	0	0	1	0
0	1	1	1	1	0
0	0	0	0	0	0

Figure 50 Square Template (Reprised)

Figure 60 is the output of a feature detector that returns probabilities.

0.8	0.97	0.56	0.33	0.23	0.01
0.75	0.25	0.21	0.44	0.34	0.2
0.56	0.15	0.53	0.01	0.89	0.01
0.63	0.43	0.72	0.32	0.77	0.12
0.89	0.66	0.61	0.72	0.43	0.1
0.71	0.12	0.21	0.11	0.32	0.12

Figure 60 Output of a Probabilistic Feature Detector

The expected weighted sum of votes from figure 60 is 5.90. Using a threshold of .5 probability to decide which points are features gives results in 5 votes.

Intuitively the expected weighted sum of votes should be a more useful statistic. For example consider figure 61.

0	0	0	0	0	0
0	0.49	0.49	0.49	0.49	0
0	0.49	0	0	0.49	0
0	0.49	0	0	0.49	0
0	0.49	0.49	0.49	0.49	0
0	0	0	0	0	0

Figure 61 Output of a Probabilistic Feature Detector

If one used a threshold of 0.5 probability then 0 votes would be reported by the matcher when matching the template in figure 50. However the expected weighted sum of votes from this output is 5.88. 5.88 is a better representation of the evidence for the existence of an object than 0.

6. Modeling Multiple Objects

The previous sections of this chapter consider the problem of using a single template at a single position. However usually template matching is used to find many objects that can occur at a variety of positions. Thus template matching returns an array of results from matching a template over a grid of positions. Also results from matching several templates (corresponding to several objects) can be available. This section discusses object recognition using an array of results from template matches.

Assume that the input to the template matcher comes from an edge detector. Thus a template of an object is a set of pixels on the boundary of the object. Assume that there is a set of objects, O (such sets are { square, triangle, circle } and { wrench, screwdriver, hammer }) and a set of positions for objects, Q such that given object, $o_i \in O$ and position, $q_j \in Q$, $o_i(q_j)$ means that there is an o_i at position q_j . Note that a shape can occur at many positions so $o_i(q_j) \& o_i(q_k)$ can be true.

For example the template in figure 62 is a template for a square object in the middle of the image.

0	0	0	0	0	0
0	1	1	1	0	0
0	1	0	1	0	0
0	1	1	1	0	0
0	0	0	0	0	0
0	0	0	0	0	0

Figure 62 Small Square Template

The template in figure 63 is a template for a square on the right side of the image.

0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	1	1	1
0	0	0	1	0	1
0	0	0	1	1	1
0	0	0	0	0	0

Figure 63 Small Square on the Right Side Template

The template in figure 64 is the template for a triangle in the upper left corner of the image.

1	1	1	1	0	0
1	0	1	0	0	0
1	1	0	0	0	0
1	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0

Figure 64 Triangle Template

To simplify the problem assume that $o_i(q_j)$ is independent of $o_k(q_t)$. Thus objects are dropped on the world at random with respect to other objects. Also assume that the probabilities of false negatives and positives are independent at each pixel. Thus if $o_i(q_j)$ does not add extrinsic edges or occlude edges from $o_k(q_t)$ then the determination of $o_k(q_t)$ from the input data is independent of $o_i(q_j)$.

Call the set of objects that occlude or add extrinsic edges to the boundary of $o_k(q_i)$, the neighborhood of $o_k(q_i)$, $N(o_k(q_i))$. For example both the objects of figure 63 and figure 64 are in the neighborhood of the object in figure 62 because they overlap it. But the object in figure 64 is not in the neighborhood of the object in figure 63.

If the label of every element $n \in N(o_k(q_i))$ was known then one could compute how likely it is that a pixel of the template of $o_k(q_i)$ would be occluded by a member of $N(o_k(q_i))$ given that $o_k(q_i)$ was present or how likely it is that a pixel of the template has an extrinsic edge added from $N(o_k(q_i))$ when $o_k(q_i)$ is absent.

A system that determines the probabilities of each element of a set from its neighborhood defines a Markov random field on the events $O(Q)$. Recently many algorithms have been proposed for interpreting data modeled by Markov random fields. Causal Markov random fields can be handled as Markov chains (Ronse 1985) (Hansen and Elliot 1982). However the fields described here are unlikely to be causal. Techniques for finding the MAP estimate of a Markov random field using simulated annealing were applied to vision in (Geman and Geman 1984). However the MAP estimate is often inappropriate for vision problems (Sher 1986c) (Marroquin, Mitter, and Poggio 1985). In (Marroquin 1985) techniques are developed for finding probability distributions for each element of a Markov random field. For the fields described here this algorithm yields the probability of each object at each position. An application of that algorithm to template matching works thus.

Calculate the posterior probability for all the objects and positions considered assuming their neighborhoods are in a default state (usually all off). Threshold at 0.5 probability to form an estimate of which objects are at what positions. Now a state of the neighborhood of each object position is available. Then compute the probability of each object being at each position from their neighborhoods (adjusting the p^e and

p^o at each pixel). Call these probabilities $P(o_k(q_t))$. If in the estimate $o_k(q_t)$ is 0 change it to 1 with probability $P(o_k(q_t))$. If $o_k(q_t)$ is 1 in the estimate make it 0 with probability $1-P(o_k(q_t))$. Iterate many times. For each $o_k(q_t)$ the percentage of time $o_k(q_t)$ is true approaches the probability of $o_k(q_t)$ given the Markov model and the observed data.

Faster algorithms for estimating the state of a system modeled by Markov random fields are being examined by Paul Chou at the University of Rochester. Any such algorithms developed could be applied to template matching.

7. Proposed Experiment

This section proposes a simple experiment to test the algorithms proposed in this chapter on artificial and real data. The experiment is to implement the algorithm that finds vertical lines of length n (n is a parameter) in edge detector output. The template for a vertical line is shown in figure 65.

1
1
1
1
1
1

Figure 65 Template for Vertical Line of Length 6

Every point in the template p^P is a constant determined by the edge detector and threshold in use. p^n is larger at the endpoints since if the segment stops at a corner as in figure 66 the edge detector is more likely to fail.

0	0	1	0	0
0	0	1	0	0
0	0	1	0	0
0	0	1	0	0
0	0	1	0	0
0	0	1	0	0
1	1	1	1	1

Figure 66 Picture of corner in image

However the percentage of line segments of length n that happen to end in corners may be small enough that p^n is practically the same for all the pixels of the template. The rest of this section considers p^n to be constant for all the pixels of the template.

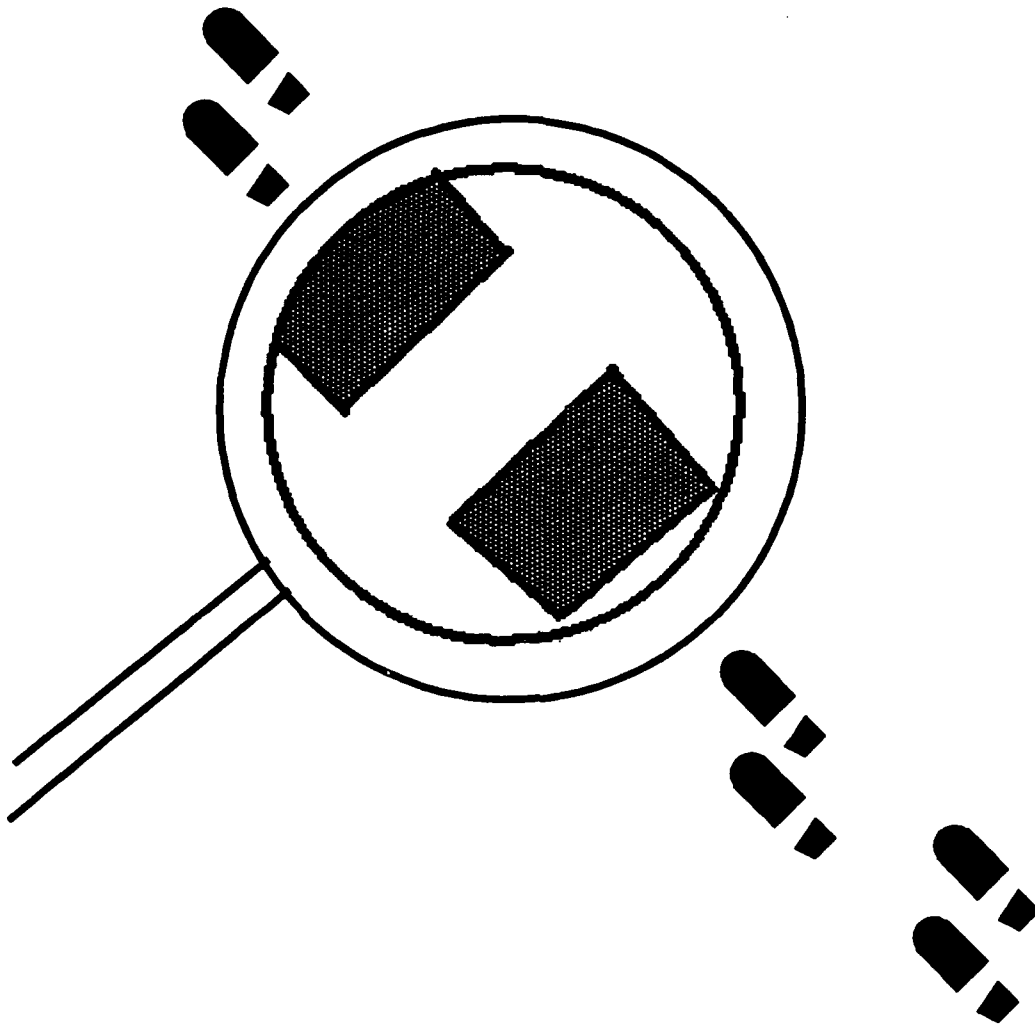
Initially p^o and p^e are considered constant too. p^n and p^p are derived from experiments with the input edge detector. p^o and p^e are derived from a model of the image ensemble. Given these values the template can be applied to the n by 1 windows of an image and for each window a probability that a vertical line segment is there can be recovered using the mathematics derived in section 4.2. Thus an initial estimate of the positions of line segments in the image is derived.

Next the algorithm from section 4.6 is applied to the problem. For the preliminary experiments one may assume that the segments do not occlude each other but only add extrinsic edges. The neighborhood of a line segment is the line segments that intersect it. Assume that the probability of line segment L_0 is being determined and an intersecting line segment, L_1 , has been turned on. Then for each pixel in $L_0 \cap L_1$ $p^e + p^p$ should be set to $1 - p^n - p^o$ since we have determined that the pixel should be on even if L_0 is not in the image. Thus the pixels in the intersection of L_0 and L_1 do not contribute to computing the probability of L_0 .

The algorithm in 4.6 yields the probabilities for each line segment. After applying that algorithm one can threshold these probabilities and test the percentage of the time they were computed correctly. This experiment can yield preliminary

results that demonstrate the effectiveness of the algorithms in this chapter. The results of section 4.3 can also be tested if this experiment is implemented.

Chapter 5



Evidence Combination

CHAPTER 5

Evidence Combination

Chapters 3 and 4 contain techniques for using a domain model, M , to generate probability distributions for feature labels given the observed data. Likelihoods, $P(O | f=l \& M)$, were computed as an intermediate step towards calculating these probabilities.

This chapter discusses merging several competing models, $M_1, M_2, M_3 \dots$, that have likelihood generators. Each model has a probability of accurately modeling any section of an image. Their likelihood generators yield a probability distribution over labels for a feature. This chapter shows how each model's likelihood generator contribute to the probability distribution over feature labels proportionately to the probability of the model's assumptions being correct.

Thus one does not need a boundary pixel detector that also understands corners. One can construct a likelihood generator for a model that understands boundaries and a likelihood generator for a model that understands corners. Near a corner the results from the corner model are used. Far from a corner the boundary model is used.

This section takes a Bayesian approach to evidence combination related to the work of Good (Good 1950), (Good 1983b), and Bolles (Bolles 1976). Neither Good nor Bolles have approached the problem of combining computations based on several models of the same data. Wesley and Hanson (Wesley and Hanson 1982) discuss using Dempster-Shafer theory for this kind of evidence combination for

high-level vision. Reynolds, Strahman and Lehrer (Reynolds, Strahman, and Lehrer 1985) discuss how low-level vision operators can generate input to a Dempster-Shafer theory. However Dempster-Shafer evidence theory has problems. It has been reduced to an interval Bayesian evidence theory in (Kyburg 1987), (Hummel and Landy 1985) and (Grosz 1985). The Bayesian system that yields the Dempster-Shafer theory contains an implicit independence assumption. All assumptions are made explicit in this dissertation. In particular, there is no independence assumption necessary for the evidence combination rules. All the rules derive from identities of probability theory.

1. Desiderata for Evidence Combination

The rules for combining evidence from several domain models were originally developed with certain desiderata in mind. These desiderata are as follows.

- (1) The combined detector takes *a priori* knowledge about the reliability of the detectors' models into account. (Dempster-Shafer combination rules do not.) This desideratum is important because often a specialized model is known to be *a priori* unlikely. For example a corner pixel is a rare phenomenon. However such a model can be important because the features it describes are important.
- (2) The combined detector takes *a posteriori* knowledge about the reliability of the detectors' models into account. If evidence in the image indicates that a specialized detector's model does not apply then the result of that detector does not affect the output of the combined detector.
- (3) The time required to execute the combined detector is linear in the times required to execute the specialized detectors.

The first two desiderata are achieved by using the techniques developed in sections 5.2 and 5.3. The last desideratum is achieved because the disjunction of two models is a model. The combination of two detectors built from models M_1 and M_2 is a detector built from the model $M_1 \vee M_2$. Thus the cost of constructing the combination of detectors built from models M_1 , M_2 , and M_3 is the cost of combining M_1 and M_2 plus the cost of combining $M_1 \vee M_2$ with M_3 . Thus if the cost of combining two detectors is constant the cost of combining n detectors is linear in n .

2. Combining Priors

Assume there are two domain models, M_1 and M_2 , either of which may accurately model the image. A feature f with labels $l \in L$ can be detected using M_1 or M_2 . Following the approach of this dissertation, the feature detectors are built from a prior probability distribution and a likelihood generator. This section shows how to combine the prior probability distribution from M_1 and M_2 .

The combined prior is $P(f=l | M_1 \vee M_2)$. Equation 26 uses probability theory to express the combined prior in terms of individual priors and *a priori* probabilities of the models applying.

$$\begin{aligned}
 P(f=l | M_1 \vee M_2) &= \frac{P(f=l \& M_1 \vee M_2)}{P(M_1 \vee M_2)} \\
 &= \frac{P(f=l \& M_1) + P(f=l \& M_2) - P(f=l \& M_1 \& M_2)}{P(M_1) + P(M_2) - P(M_1 \& M_2)} \quad (26) \\
 &= \frac{P(f=l | M_1)P(M_1) + P(f=l | M_2)P(M_2) - P(f=l | M_1 \& M_2)P(M_1 \& M_2)}{P(M_1) + P(M_2) - P(M_1 \& M_2)}
 \end{aligned}$$

The combined prior is expressed in terms of:

- (1) the priors from the two models

- (2) the *a priori* probabilities of the two models
- (3) the prior of the conjunction of the two models
- (4) the *a priori* probability of the conjunction of the two models

The tricky part of this rule is the conjunction of two models. The conjunction of two models apply when two models simultaneously apply. Often the conjunction has 0 probability. For example if M_1 assumes that the noise in the image has $\sigma=4\pm\epsilon$ and M_2 assumes that the noise in the image has $\sigma=8\pm\epsilon$ then they both can not be simultaneously true and $P(M_1\&M_2)=0$. Two such models are *disjoint* models. In this dissertation only disjoint models are combined. Equation 27 shows the rule for combining priors from two such models. Equation 27 is derived by setting $P(M_1\&M_2)=0$ in equation 26.

$$P(f=l|M_1\vee M_2)=\frac{P(f=l|M_1)P(M_1)+P(f=l|M_2)P(M_2)}{P(M_1)+P(M_2)} \quad (27)$$

3. Combining Likelihoods

This section shows how to take two likelihood generators from domain models M_1 and M_2 and get a likelihood generator for $M_1\vee M_2$. For this derivation in equation 28 prior probabilities for the feature labels under each model ($P(f=l|M_1)$ and $P(f=l|M_2)$) are necessary. If $P(M_1\&M_2)\neq 0$ then the prior probability of the feature label under the conjunction of M_1 and M_2 ($P(f=l|M_1\&M_2)$) and the output of a likelihood generator for the conjunction of the two models ($P(O|f=l\&(M_1\&M_2))$) are needed. Equation 28 uses probability theory to derive the rule for combining likelihoods.

$$\begin{aligned}
 P(O \mid f=l \& (M_1 \vee M_2)) &= \frac{P(O \& f=l \& (M_1 \vee M_2))}{P(f=l \& (M_1 \vee M_2))} \\
 &= \frac{\left[\begin{array}{c} P(O \& f=l \& M_1) \\ + \\ P(O \& f=l \& M_2) \\ - \\ P(O \& f=l \& (M_1 \& M_2)) \end{array} \right]}{\left[\begin{array}{c} P(f=l \& M_1) \\ + \\ P(f=l \& M_2) \\ - \\ P(f=l \& (M_1 \& M_2)) \end{array} \right]} \quad (28) \\
 &= \frac{\left[\begin{array}{c} P(O \mid f=l \& M_1)P(f=l \mid M_1)P(M_1) \\ + \\ P(O \mid f=l \& M_2)P(f=l \mid M_2)P(M_2) \\ - \\ P(O \mid f=l \& (M_1 \& M_2))P(f=l \mid (M_1 \& M_2))P(M_1 \& M_2) \end{array} \right]}{\left[\begin{array}{c} P(f=l \mid M_1)P(M_1) \\ + \\ P(f=l \mid M_2)P(M_2) \\ - \\ P(f=l \mid (M_1 \& M_2))P(M_1 \& M_2) \end{array} \right]}
 \end{aligned}$$

Equation 28 can be simplified if $P(M_1 \& M_2) = 0$. The simplified version of equation 28 is equation 29.

$$\begin{aligned}
 P(O \mid f=l \& (M_1 \vee M_2)) &= \frac{\left[\begin{array}{c} P(O \mid f=l \& M_1)P(f=l \mid M_1)P(M_1) \\ + \\ P(O \mid f=l \& M_2)P(f=l \mid M_2)P(M_2) \end{array} \right]}{\left[\begin{array}{c} P(f=l \mid M_1)P(M_1) \\ + \\ P(f=l \mid M_2)P(M_2) \end{array} \right]} \quad (29)
 \end{aligned}$$

When a model's assumptions fit the observed data well, the likelihoods generated from that model are large. If the fit is poor, the likelihoods are small.

Assume that M_1 yields large likelihoods and M_2 yields small ones. Also assume that the prior probabilities are uniform. Equation 29 then reduces to taking the average of the two likelihoods weighted by the prior probabilities of the two models. To simplify further assume that M_1 and M_2 are equally *a priori* probable.

When equation 29 is used to combine their likelihoods the large likelihoods from M_1 dominate the average. Thus the combined likelihoods are similar to those generated with M_1 . Applying Bayes' law (equation 2) to the combined likelihoods yields probabilities similar to those generated by M_1 . Thus when M_1 fits and M_2 does not the combined probability distribution over feature labels is similar to M_1 's distribution. If the likelihoods from M_1 and M_2 are equal the combined probability distribution is the average of the probabilities computed from M_1 and M_2 . Thus the evidence combination rule in this section takes the fit of the model to the observed data into account when combining evidence. Section 5.4 demonstrates this robustness with a simple example of evidence combination.

4. A Simple Example of Evidence Combination

This section presents a simple example of evidence combination. Assume that a coin is being flipped. There is a 50-50 chance that the coin is biased so that heads come up 90% of the time. If the coin is not biased then it is a fair coin and heads come up 50% of the time. Thus there are two models of the coin, M_b models the biased coin and M_f models the fair coin. Coin flips are independent events.

The task is to determine the probability that the fifth flip of the coin is heads after seeing the first four flips. The feature is the fifth flip. There are two labels for the feature, H and T. The observed data is the first 4 flips. There are sixteen possible observations in figure 67.

HHHH	HHHT	HHTH	HHTT
HHTH	HTHT	HTTH	HTTT
THHH	THHT	THTH	THTT
TTHH	TTHT	TTTH	TTTT

Figure 67 Coin Flip Observations

The prior probability of H from M_b is 0.9. The prior from M_f is 0.5. The coin can not be simultaneously fair and biased so $P(M_b \& M_f) = 0$. Hence equation 27 is used to compute the prior for the combination of the biased and fair model. This equation takes the average of the two priors to result in a prior probability of 0.7. Thus the prior probability of heads is 0.7.

Figure 68 is a table of the probabilities for the five distinct cases (since coin flips are independent only the total number of heads counts).

Number Heads	M_b Likelihood	M_f Likelihood	Comb. H Likelihood	Comb. T Likelihood	Comb. H Probability
0 (TTTT)	0.0001	0.0625	0.0224	0.0521	0.5006
1 (HTTT)	0.0009	0.0625	0.0229	0.0522	0.5057
2 (HHTT)	0.0081	0.0625	0.0275	0.0534	0.5459
3 (HHHT)	0.0729	0.0625	0.0692	0.0642	0.7154
4 (HHHH)	0.6561	0.0625	0.4441	0.1614	0.8652

Figure 68 Table of Probabilities Computed for Coins

Note that when there are 0 heads in the observed data the likelihoods generated by M_b are small and the probability of the combination is close to that yielded from M_f . When there are 4 heads the likelihoods generated by M_b are large and the probability of the combination is close to that yielded from M_b . When there are 3 heads the likelihood from M_b and that from M_f are near each other and the probability is approximately that of the average.

5. Maximum Entropy

The previous chapters computed probabilities conditioned on some model. Thus the algorithms yield the probability distribution over feature labels conditional upon the assumptions of a domain model M being met and the observed data. The techniques in the previous chapter compute probabilities conditional upon the disjunction of several models. However the user requires the probability distribution

of feature labels conditioned only on the observed data. Such a probability is often called a *physical probability distribution*.

Approximating the physical probability distribution of feature labels requires considering that every one of the domain models can fail. Given a "model" for all the models failing, the evidence combination techniques in sections 5.2 and 5.3 can be used to combine the probabilities predicated on the models working with the probabilities predicated on the models failing to get physical probabilities. This section uses maximum entropy techniques to "model" the domain models failing.

This section assumes that a uniform distribution over events has the least information in it. This assumption is derived from the *maximum entropy* approach to uncertainty (Jaynes 1985) (Grandy 1985). (Loui 1984) describes some problems with this approach. The main problem with using a uniform distribution to express ignorance is the distribution changing when the labeling of a feature changes. For example chapter 3 contains two different labelings for boundary pixels. One is in equation 5 (reprinted here) the other is in equation 6 (reprinted here).

$$L = \{ \textit{boundary}, \neg \textit{boundary} \} \quad (5)$$

$$L = A \cup \{ \neg \textit{boundary} \} \quad (6)$$

A uniform distribution over L from equation 5 places a probability of 0.5 on $\neg \textit{boundary}$. A uniform distribution over L from equation 6 places a probability less than 0.5 on $\neg \textit{boundary}$. But in both cases the same event is expressed by $\neg \textit{boundary}$. However some technique need be used to express ignorance. Maximum entropy is a popular technique and so this section accepts it in spite of its problems. Some others who have taken this approach in image processing are (Andrews and Hunt 1977c) (Skilling and Gull 1985) (Frieden 1985) (Herman 1985).

Using this principle a uniform distribution is the prior distribution for all domain models failing. The likelihood generator for all domain models failing is a constant function. The constant is 1 over the cardinality of the set of possible observations. Thus all observations are equally probable when all domain models fail.

Note that using maximum entropy likelihood generator in Bayes' law (equation 2 (reprinted here)) generates posterior probabilities equal to the prior probabilities entered.

$$P_f(l|a\&D) = \frac{L_f(a|l\&D)prior_f(l)}{\sum_{l' \in V} L_f(a|l'\&D)prior_f(l')} \quad (2)$$

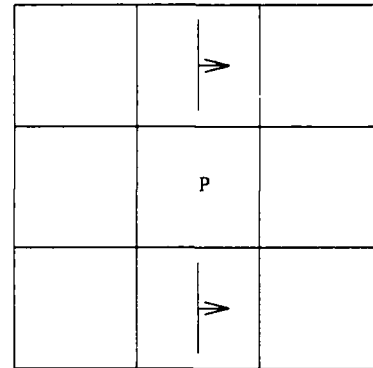
Thus these likelihoods add no information to the priors. This technique is sometimes referred to as minimizing cross entropy (Johnson and Shore 1985).

Thus a prior and likelihood generator exists for the case of all models failing.

6. Maximum Entropy and Markov Random Fields

The maximum entropy likelihood generator can be useful with Markov Random field models. Markov random field models supply prior probabilities that encode properties of neighborhoods of features (Geman and Geman 1984) (Marroquin, Mitter, and Poggio 1985). Markov random fields encode such facts as that boundaries are continuous and in general smooth, and surfaces are smooth. Using a Markov random field with a maximum entropy likelihood generator gets the constraints of the field ignoring the data in the image. Combining the maximum entropy likelihood generator with the likelihood generator from a model using the technique from section 5.3 yields an algorithm that uses likelihoods from the model when the model fits the image well and uses maximum entropy when the model fails. Using such a combination with a Markov random field allows a field that applies the neighborhood constraints more strongly when the model doesn't fit than when it does.

For example a common use for Markov Random fields is edge linking. Assume that the maximum entropy likelihood generator generates $2.328E-10$ ($1/256^{**4}$) for its likelihood. Consider pixel P in figure 69, a tentative labeling of an image modeled with a Markov random field.



|
→ indicates a boundary pixel

Figure 69 Pixels in a Markov Random Field

A Markov random field would place a high prior probability that P is a boundary pixel. If the likelihood generator for M returns 10^{-14} likelihood of no boundary and 10^{-17} likelihood of a boundary then combining the Markov random field with M would label P as no boundary because the likelihood ratio for no boundary is 1000. However combining M with the maximum entropy prior reduces the likelihood ratio to near 1. Thus the prior from the Markov random field labels P as a boundary. This is a good idea because a likelihood of 10^{-14} indicates that M does not fit the data and thus the likelihood ratio generated by M is garbage. On the other hand if M generates a likelihood of 10^{-4} for no boundary and 10^{-7} for having a boundary then even with the maximum entropy prior the likelihood ratio is 1000 against an edge. M clearly

fits the image well and should dominate the results from the Markov random field.

In the near future Paul Chou and I will test using Markov random fields with maximum entropy for segmentation

7. Operators with Several Window Sizes

One application of the combination rule of section 5.3 is to combine operators with different window sizes. Operators with large window sizes are less susceptible to noise than small windowed operators because of the law of large numbers. However when a corner where 3 objects meet occurs in a window large operators tend to fail (see section 3.5.1.2 for a clear example of this). The larger the window of an operator the more such corners are seen by the operator. Also the larger operators are sensitive to orientations that are not in the model (see section 3.5.1.1). Thus there are cases where the larger operator is more effective and there are cases where the smaller operator is more effective. Thus a combined operator should be constructed that has the good properties of the large and small operators.

Equation 28 assumes that the observed data is the same for the two operators being combined. However a 5x5 operator sees 25 pixels while a 9x9 operator sees 81 pixels. The proposed solution is to extend the 5x5 operator into a 9x9 operator. The technique that extends the 5x5 operator uses a maximum entropy assumption for the 56 unseen pixels. All possible assignments of intensities to the 56 pixels are assigned the same probability. Thus the likelihood generated by the 5x5 operator is multiplied by a constant to generate a 9x9 operator. The two 9x9 operators can then be combined by equation 29.

The model for the extended 5x5 operator is that the model fails for pixels outside its 5x5 window. The prior probability of this model applying is the probability that the model fails in at least one of the 56 unseen pixels. If f is the

probability that a model fails in a pixel then $1-(1-f)^{56}$ is the probability that f fails in at least one of the 56 pixels. The 9x9 operator assumes that its model succeeds for all 81 pixels. Thus near corners the extended 5x5 operator dominates the combination because the model (that assumes no corners) fails in the 56 pixels. Away from corners and other anomalies the 9x9 operator dominates.

8. Experimental Results

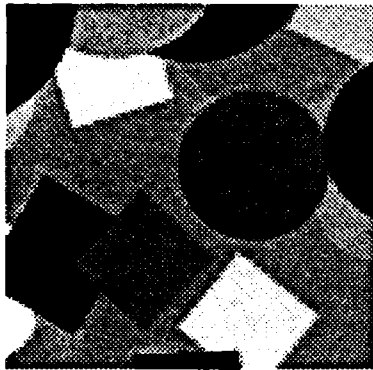
Two different forms of evidence combination have been tested on the likelihood generators developed in chapter 3. Operators tuned to several levels of noise in the image have been combined to develop an operator that behaves well for all the levels of noise tuned to. Operators with different window sizes have been combined to get an operator with the advantages of both small and large window sizes.

8.1. Experiments Combining Noise Levels

This section shows the results of experiments with combinations of operators tuned to several noise levels. Section 5.8.1.1 shows the effect of combining noise levels on artificial data. Section 5.8.1.2 shows the effect of combining noise levels on camera data.

8.1.1. Combining Noise with Artificial Images

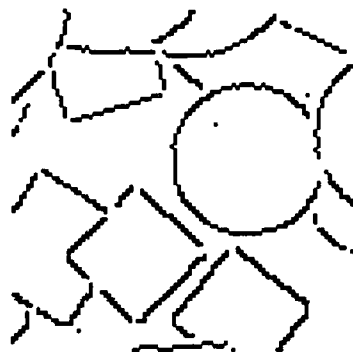
The artificial images described in section 3.5.1 figures 18, 19 and 20 are used for collecting quantitative data on the effectiveness of the evidence combination techniques. The same 5x5 operator is applied to these images as the one in section 3.5. The operators tuned to noise $\sigma=4,8,12,16$ is compared to the combination of these operators on the images with $\sigma=4,8,12,16$. Figure 70 compares the tuned to the noise level operator with the combined operator and the tuned $\sigma=12$ operator on the image from figures 18 with $\sigma=4,12,20$.



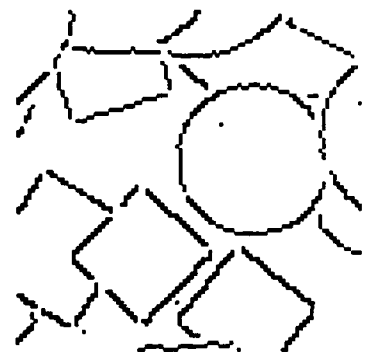
a: $\sigma=4$ Image



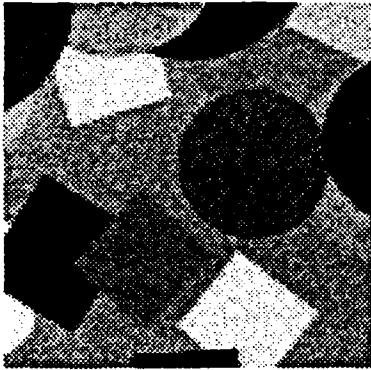
b: $\sigma=12$ Operator



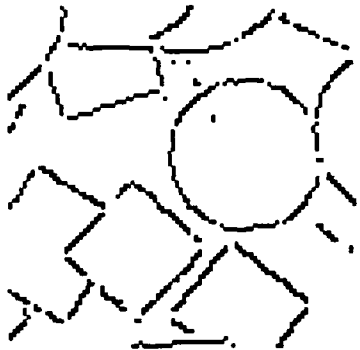
c: Combined Operator



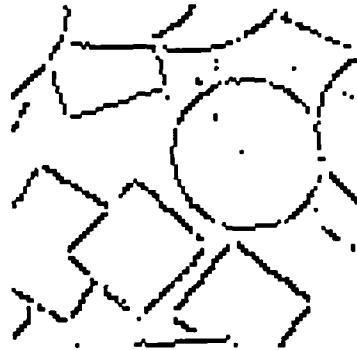
d: Tuned Operator



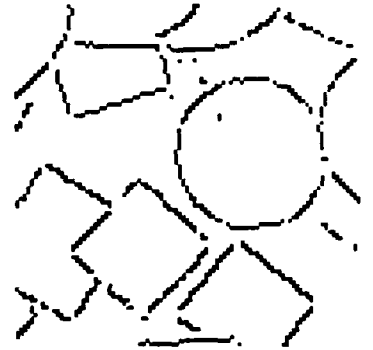
e: $\sigma=12$ Image



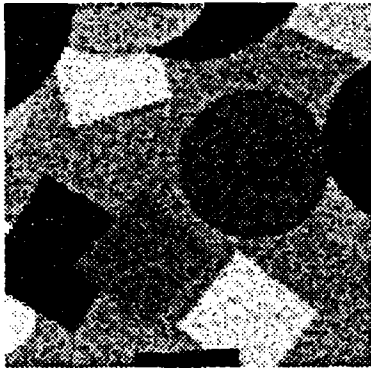
f: $\sigma=12$ Operator



g: Combined Operator



h: Tuned Operator

i: $\sigma=20$ Imagej: $\sigma=12$ Operator

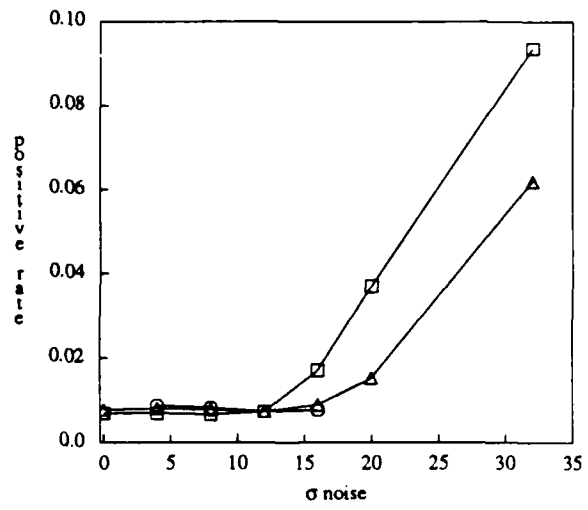
k: Combined Operator

l: $\sigma=16$ Operator

Figure 70 Combination of Noise Levels On Artificial Images

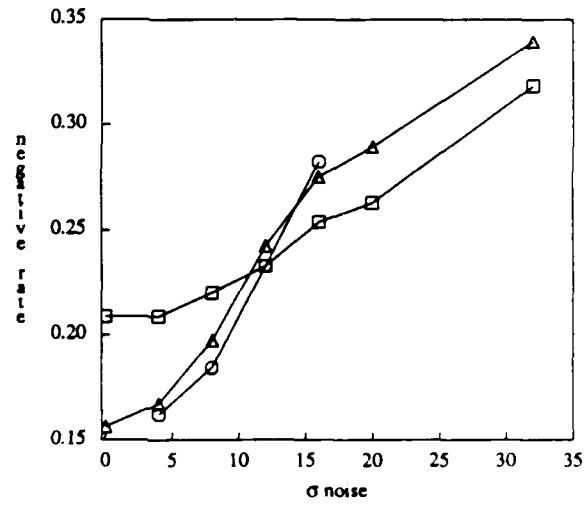
For these images the combined operator is only slightly worse than the tuned operator. Since the tuned operator is being combined with three worse operators in equal proportion these results demonstrate the robustness of the evidence combination rules.

The false positive, false negative and total error rates are shown in figures 71, 72, and 73 as a function of the standard deviation of the noise for these 4 operators.



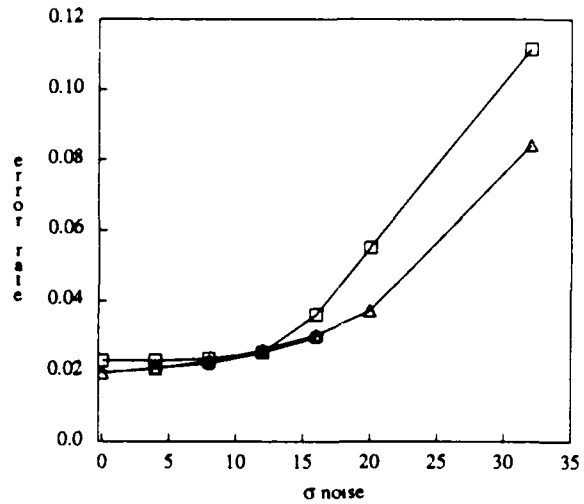
squares $\sigma=12$ 5x5 operator
circles 5x5 operator tuned to σ
triangles Combined operator

Figure 71 False Positive Rate for the Combined σ Operator



squares $\sigma=12$ 5x5 operator
circles 5x5 operator tuned to σ
triangles Combined operator

Figure 72 False Negative Rate for the Combined σ Operator



squares $\sigma=12$ 5x5 operator
 circles 5x5 operator tuned to σ
 triangles Combined operator

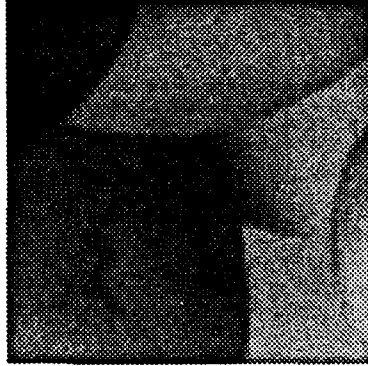
Figure 73 Total Error Rate for the Combined σ Operator

Note that the error rate of the combined operator is close to that of the operator tuned to the exact noise level. Thus the combined operator has the low error rate of a tuned operator but is more flexible than any of the tuned operators.

8.1.2. Combining Noise with Camera Images

This section shows the results from experiments combining operators tuned to several noise levels with camera imagery. The camera that took these pictures was a Vidicon Camera with $2 < \sigma < 6$. Pictures were taken of two scenes constructed from Play Doh™, a children's toy. In figure 74 a vidicon image of Play Doh™ blobs has the $\sigma=1,4,8$ and the combination thereof applied to it. Figure 75 has the same operators applied to vidicon the Play Doh™ gnome image.

(a)



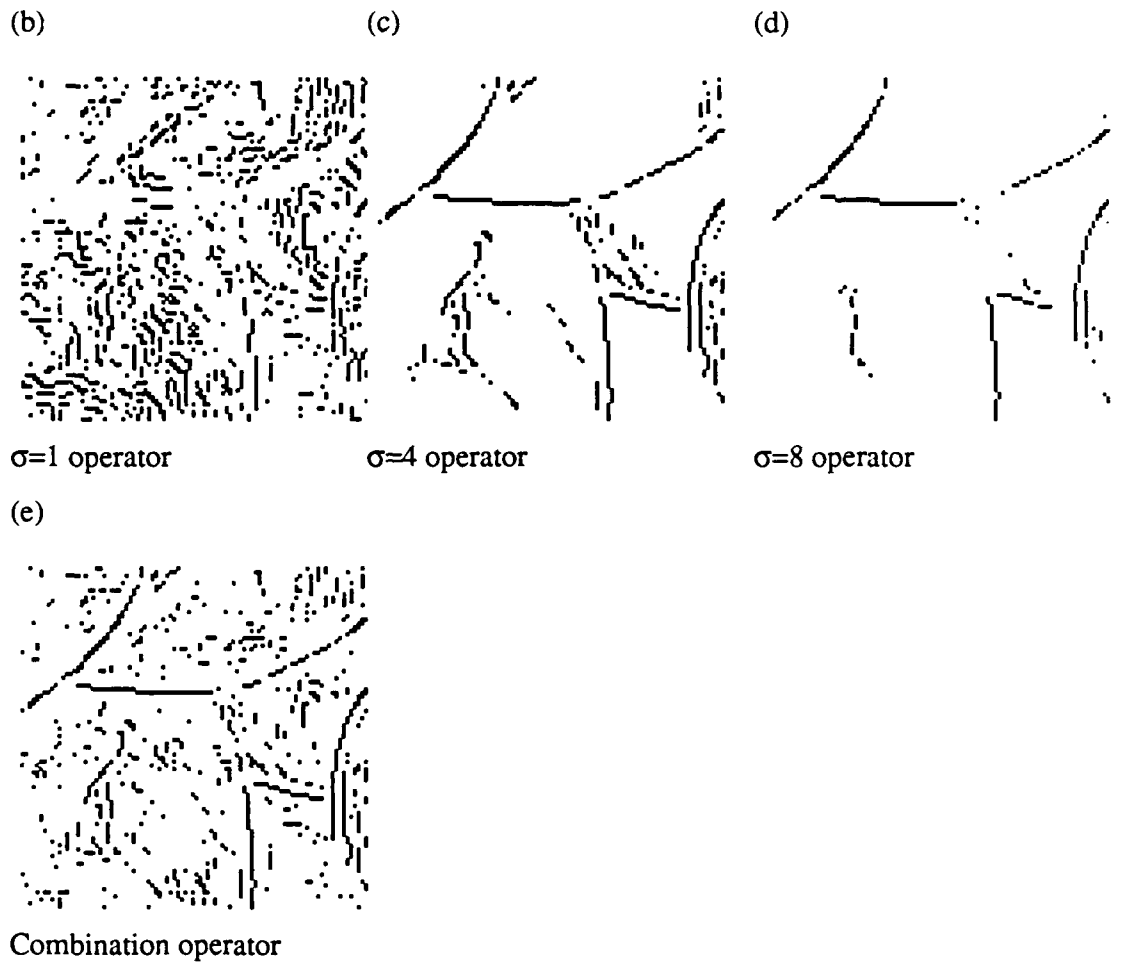
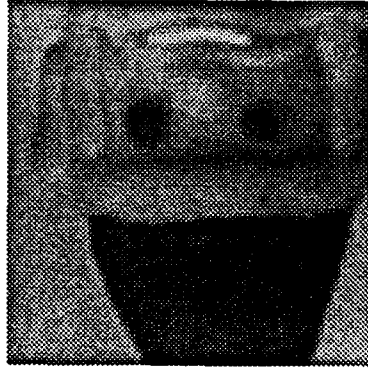


Figure 74 Evidence Combination of Noise Levels Applied to Vidicon Blobs Image

(a)



original image

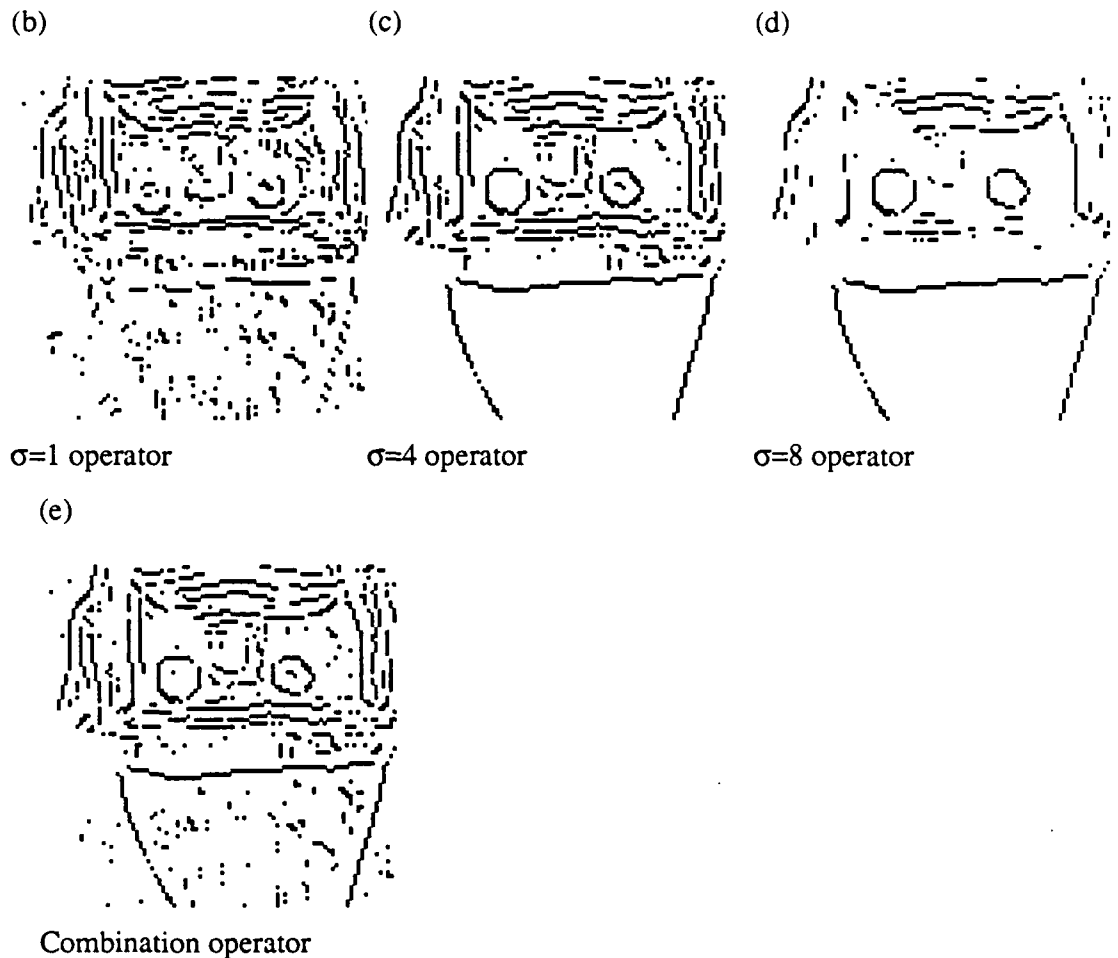


Figure 75 Evidence Combination of Noise Levels Applied to Vidicon Gnome Image

The $\sigma=1$ operator generates many false positives on these images. Some of these false positives actually represent structures in the image with low contrast such as small brightness gradients from shading. Since these structures are real the $\sigma=1$ operator has high confidence for some of these false positives and they occur in the combination. Two techniques can be used to prevent these false positives. Since they result from the $\sigma=1$ operator a low prior probability for the $\sigma=1$ operator would prevent them. A higher degree facet model would consider windows with small

gradients as regions without boundaries. A Markov random field would remove most of the false positives from the $\sigma=1$ operator too.

The combined operator has also been tested on uncontrolled imagery. Figure 76 shows the result of the same combination on Carlos' ear. Figure 77 shows the application of the combination to the aerial image.

(a)



original image

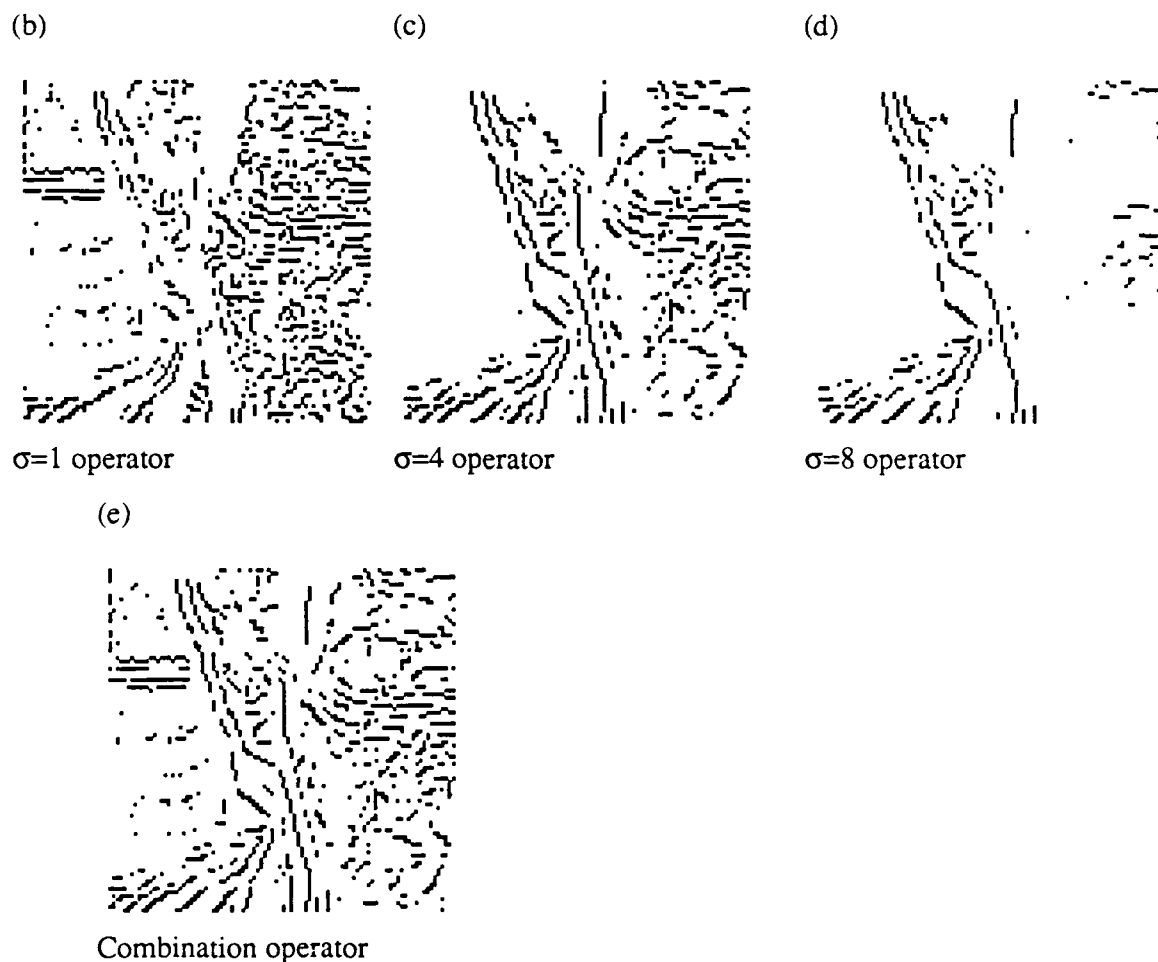
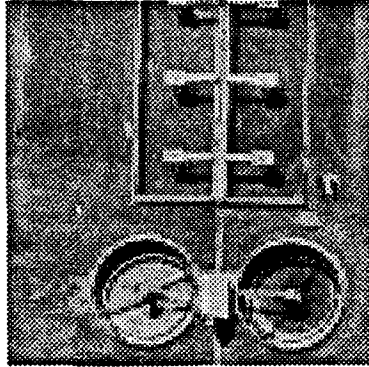


Figure 76 Evidence Combination of Noise Levels Applied to Carlos' ear

On Carlos' ear much of the fine structure in the image was reported falsely as boundaries. However on the sewage treatment plant image the combination is indeed superior to the $\sigma=4$ operator. This improvement occurs because the aerial image fits the step edge model better than the other camera imagery.

(a)



original image

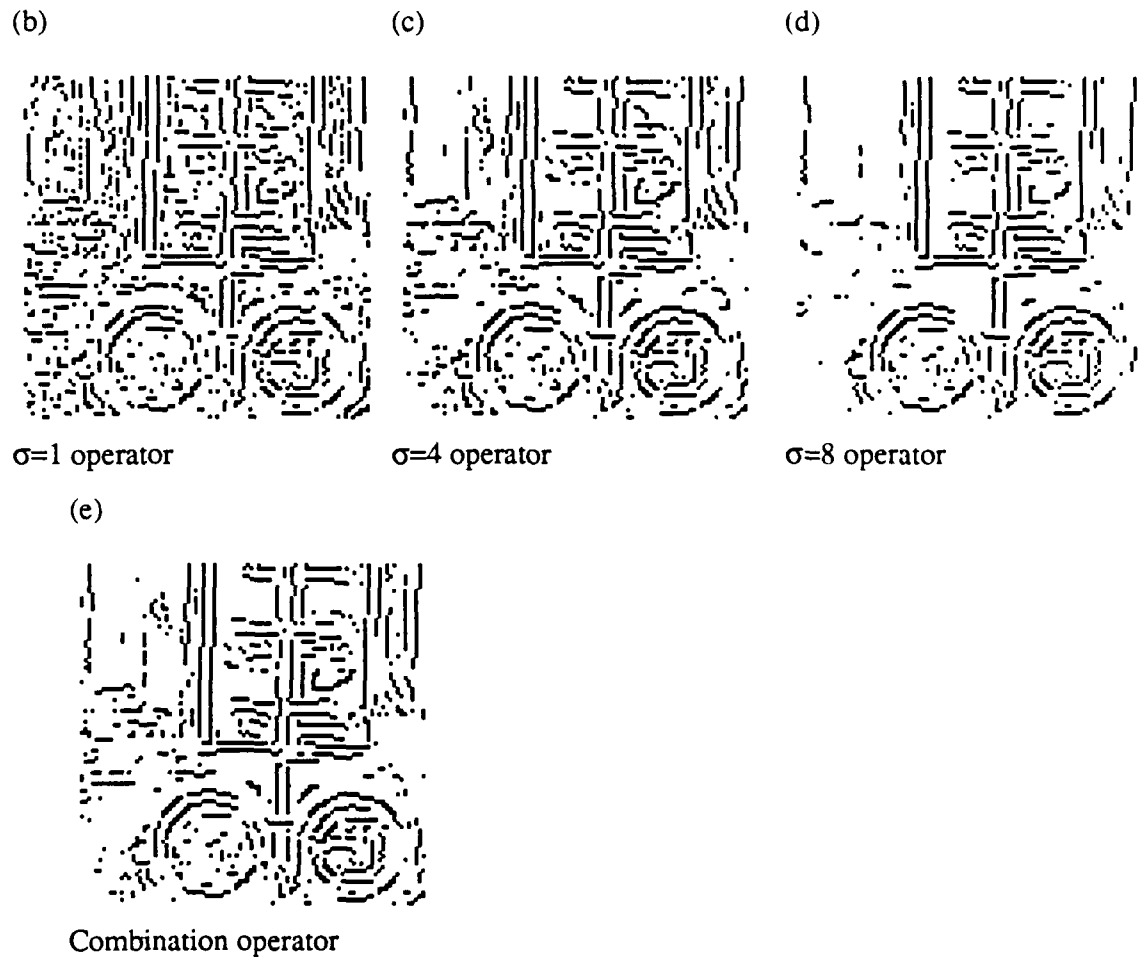


Figure 77 Evidence Combination of Noise Levels Applied to Aerial Image

8.2. Experiments Combining Sizes

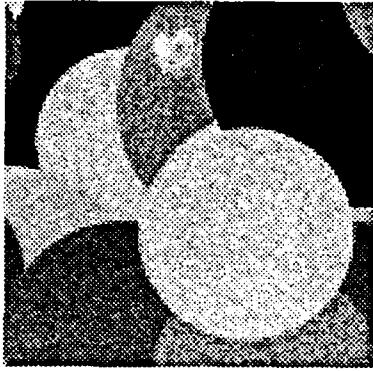
This section shows the results of experiments combining operators with several window sizes. Section 3.5 describes operators with 5x5, 7x7, and 9x9 window sizes. The 5x5 operators are sensitive to the level of the noise in the image. The larger operators are sensitive to corners and unmodeled boundary angles in the image. The combination of the 3 operators is shown here to be robust against all these sources of errors. It is shown to be at least as good as the best of the 3 operators.

Section 5.8.2.1 uses artificial data to get quantitative results for the combination. Section 5.8.2.2 shows the results of the combination on camera imagery.

8.2.1. Combining Window Sizes with Artificial Images

The results of applying the combined 5x5, 7x7 and 9x9 operator to the artificial images from section 3.5.1. The results of using this operator on the images in figures 18, 19 and 20 appears in figure 78. In figure 78 the combined $\sigma=12$ operator is compared to the 5x5 and 9x9 $\sigma=12$ operators.

(a)

 $\sigma=12$ noise level circles picture

Image



5x5 Operator

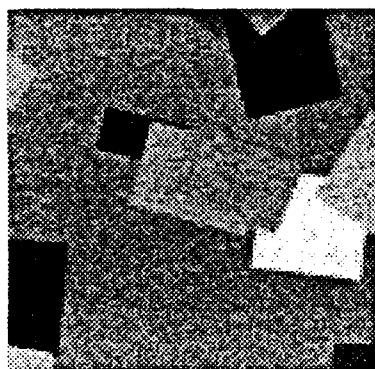


Combined Operator



9x9 Operator

(b)

 $\sigma=12$ noise level rectangles picture

Image



5x5 Operator

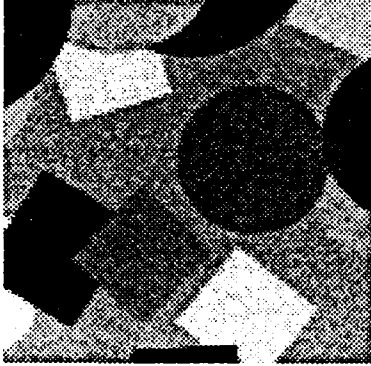


Combined Operator

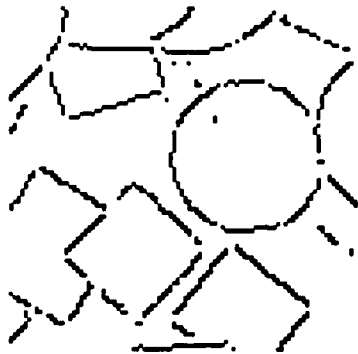


9x9 Operator

(c)

 $\sigma=12$ noise level combined circles and rectangles picture

Image



5x5 Operator



Combined Operator



9x9 Operator

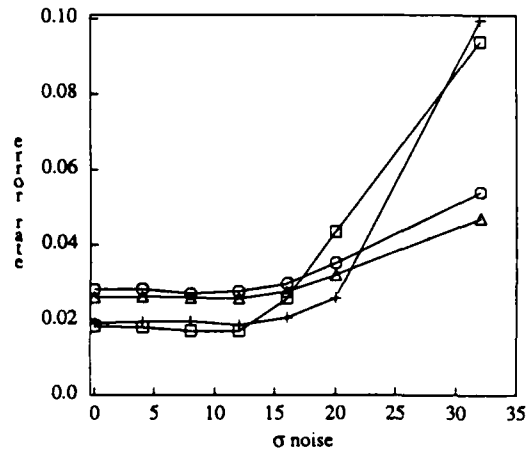
Figure 78 Combination of Noise Levels On Artificial Images

The combined operator always has better results than the 9x9 operator. In figure 78c there are less false positives from the combined operator than from the the 5x5 operator but less false negatives than the 9x9 operator. For all these pictures there is little degradation from the 5x5 operator to the combined one.

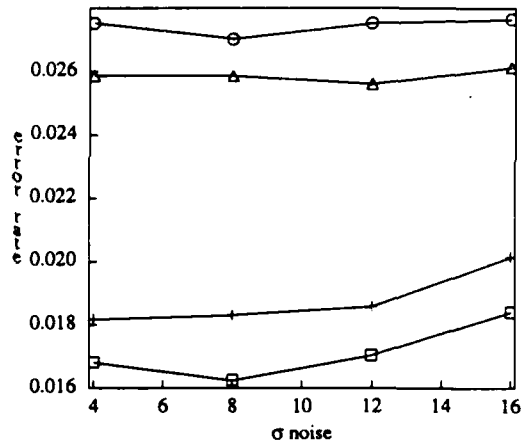
Figures 79, 80, 81, and 82 shows the effect of combing the 3 $\sigma=12$ operators compared to the 3 operators as a function of the noise level of the image. These figures show that the combined operator has the robustness of an operator with a

large window size but the performance of an operator with a small window size.

(a) $\sigma=12$ tuned operator



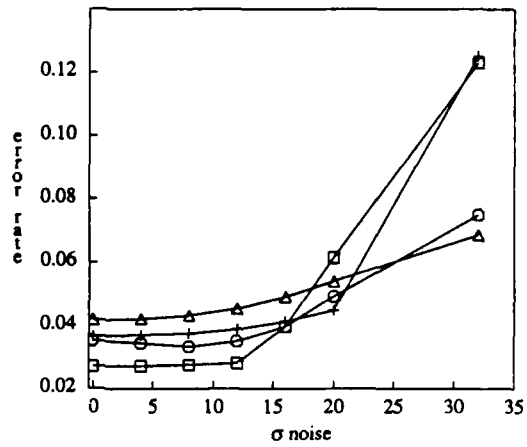
(b) operator tuned to σ noise



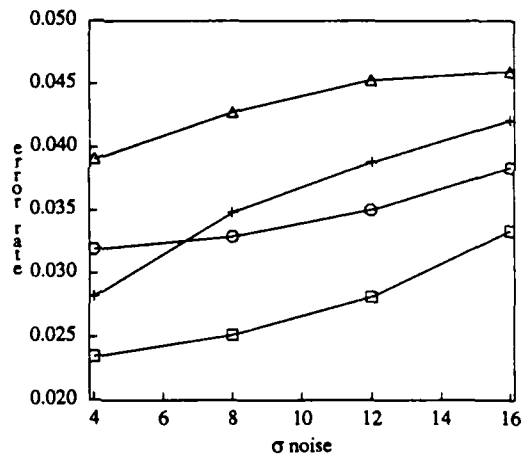
boxes 5x5 operator
 circles 7x7 operator
 triangles 9x9 operator
 +'s combined operator

Figure 79 The combination of operators with several window sizes on circles image

(a) $\sigma=12$ tuned operator



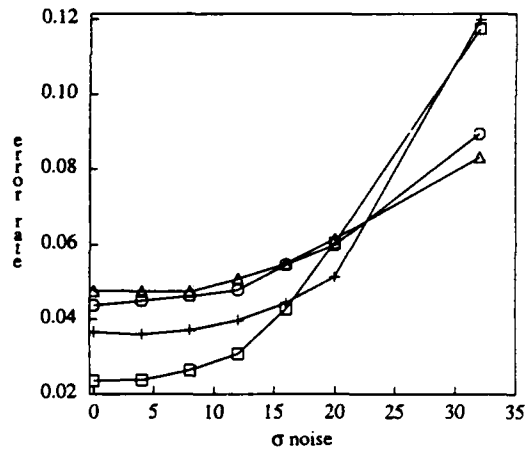
(b) operator tuned to σ of noise



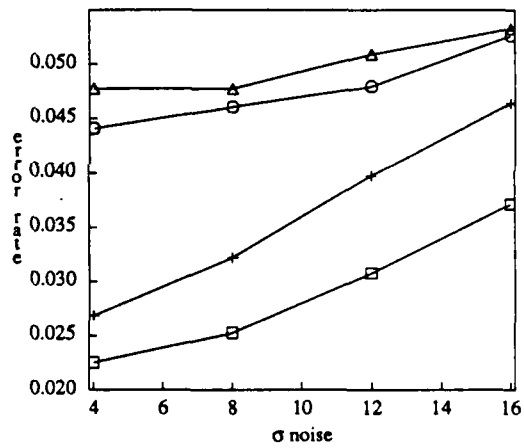
boxes 5x5 operator
 circles 7x7 operator
 triangles 9x9 operator
 +'s combined operator

Figure 80 The combination of operators with several window sizes on rectangles image

(a) operator tuned to $\sigma=12$ noise



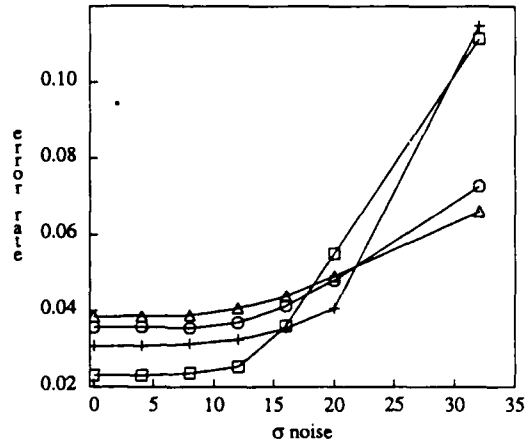
(b) operator tuned to σ of noise in image



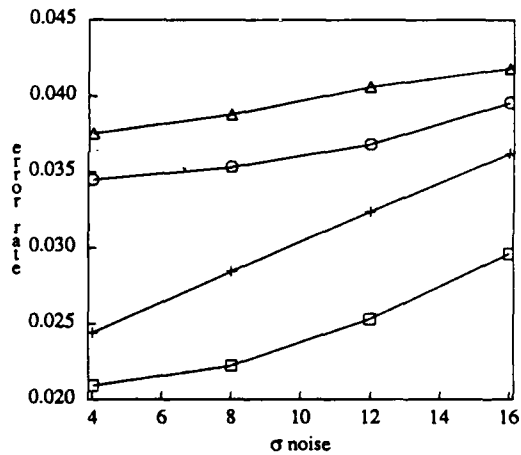
boxes 5x5 operator
 circles 7x7 operator
 triangles 9x9 operator
 +'s combined operator

Figure 81 The combination of operators with several window sizes on combined image

(a) operator tuned to $\sigma=12$ noise



(b) operator tuned the noise σ of the image

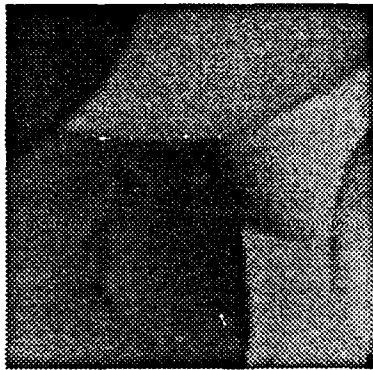


boxes 5x5 operator
 circles 7x7 operator
 triangles 9x9 operator
 +'s combined operator

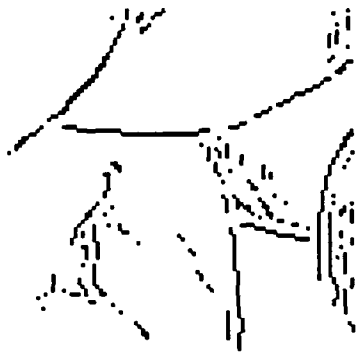
Figure 82 The combination of operators with several window sizes on all images

8.2.2. Combining Window Sizes with Camera Images

This section shows how the combination of sizes works in real life. The following figures show the combination of the $\sigma=1$ and $\sigma=4$, 5×5 , 7×7 , and 9×9 operators applied to 3 pictures taken with the CCD camera with $1 < \sigma < 4$. Figure 83 is the application to the blobs image of this operator. It is compared there with the 5×5 and the 9×9 operators. A similar comparison for the gnome image is in figure 84. A similar comparison is applied to Carlos Calderon in figure 85 and the aerial image in 86.



a: Image



b: 5x5 operator

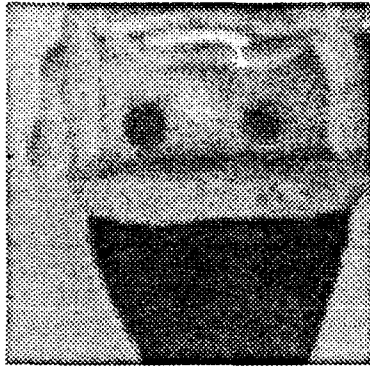


c: Combination operator

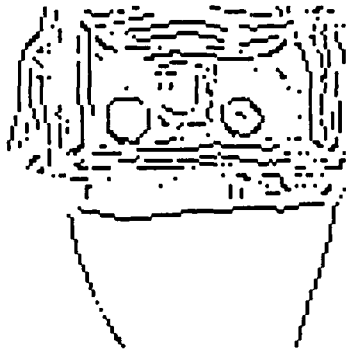


e: 9x9 operator

Figure 83 Combination of Different Window Sizes on Blobs



a: Image



b: 5x5 operator



c: Combination operator



d: 9x9 operator

Figure 84 Combination of Different Window Sizes on the Gnome

On the camera imagery the combination of the different sized operators does a better job of finding occlusion boundaries than either the 5x5 or the 9x9 operator. The combined operator picks out the important boundaries while ignoring unimportant details found by the 5x5 operator.



a: Image



b: 5x5 operator

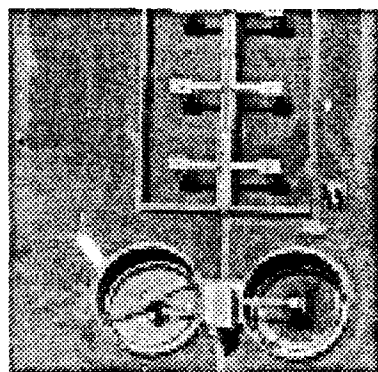


c: Combination operator

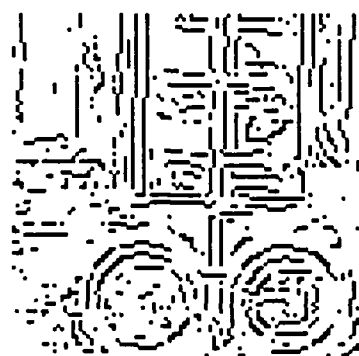


e: 9x9 operator

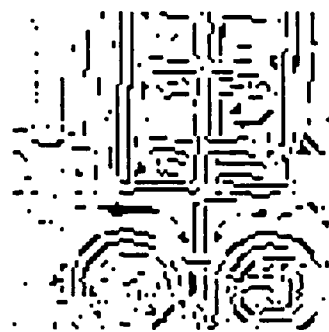
Figure 85 Combination of Different Window Sizes on Carlos' ear



a: Image



b: 5x5 operator



c: Combination operator

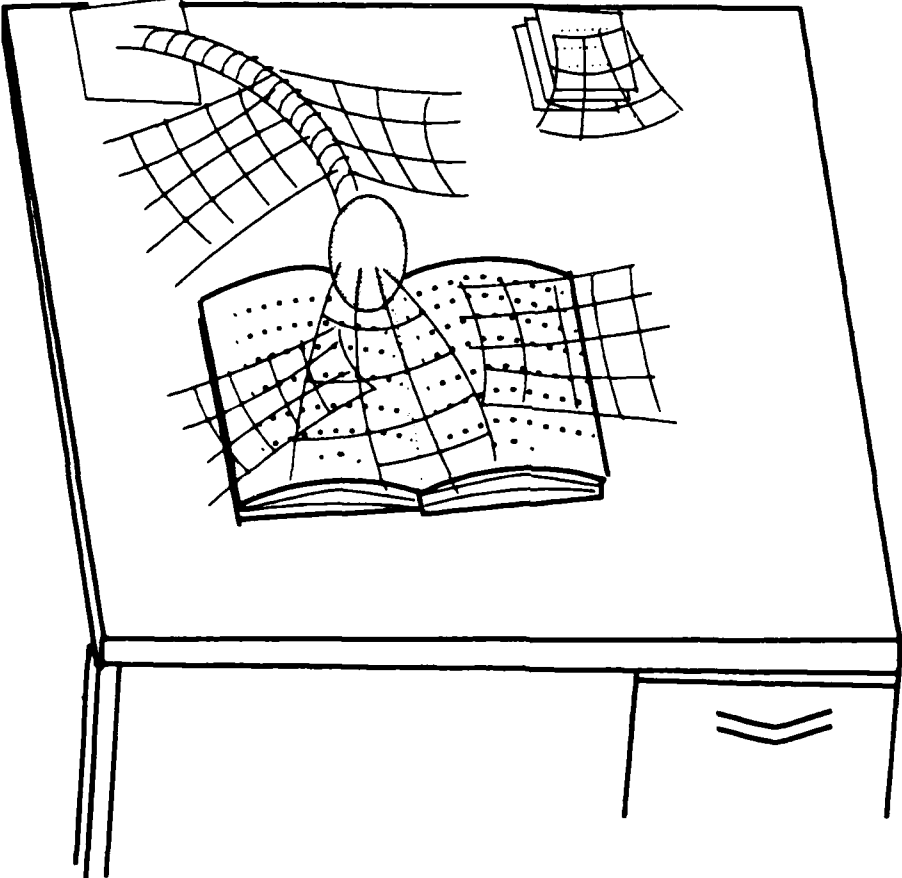


d: 9x9 operator

Figure 86 Combination of Different Window Sizes on the Sewage Treatment Plant

Because tuned operators are being combined the advantages of robustness from the combined operator are not illustrated here. On the aerial image the results from the combined operator is better than that of the 9x9 or 7x7 operator but not better than the results from the 5x5 operator. One reason for this is the lack of sophistication in the step edge model. Considering the 5x5 operator is combined using the maximum entropy assumption with the 9x9 and the 7x7 operator it is good that the important features from the 5x5 operator are preserved.

Chapter 6



Previous Work

CHAPTER 6

Previous Work

This section discusses previous work in low-level vision and evidence combination and its relationship to this dissertation. Section 6.1 discusses earlier work on modeling vision problems and its relationship to chapter 2. Section 6.2 discusses work on edge detection and its relationship to chapter 3. Section 6.3 discusses work on template matching and the Hough transform and its relationship to chapter 4. Section 6.4 discusses work on evidence combination and its relationship to chapter 5. Since the operators developed in this dissertation can be fed into a Markov random field to achieve results similar to probabilistic relaxation, section 6.5 discusses work on probabilistic relaxation and Markov random fields.

1. Previous Work on Modeling

Using models as a mathematical bridge between the data supplied by the users of a system and the data required by the designer of a system is not discussed in other works. Most probabilistic approaches to computer vision assume that required statistics such as prior probabilities are directly available to the system designers (Feldman and Yakimovsky 1974) (Bolles 1976). However Witken (Witkin 1981) derives priors for edge directions on the image assuming a uniform distribution of edge directions in the texture on the objects in the scene. Much work has been done on direct elicitation of priors for simple events (Berger 1980b).

Much of the work on edge detection has used the step edge model. Some work has been done on developing sophisticated models for low-level vision by Binford (Binford 1981). He develops a model from 3 dimensional descriptions of objects.

Haralick describes a model with piecewise polynomial surfaces with noise added and discretized (Haralick 1980). Haralick (Haralick 1984) (Haralick 1983) uses a low level image model based on image surface that are cubic polynomials in x (the horizontal distance in the image) and y (the vertical distance in the image). Haralick also has used a step edge model (Haralick 1986b). In general his models are referred to as *facet models*. Section 3.1.2 yields a facet model by simplifying a more realistic model. Section 3.2 derives a likelihood generator for boundary pixel detection from a simplified facet model. However the algorithm derived in section 3.2 can be used with a more sophisticated model of imagery such as that described in 3.4 similar to the Binfords model (Binford 1981).

Models of the imaging system (the camera) and the noise within it occur are developed by Horn (Horn 1986) Shen (Shen and Castan 1986) and Andrews and Hunt (Andrews and Hunt 1977b). These analysis suggest that a correlated Gaussian model for internal camera noise suffices.

Davis, Rosenfeld and Zucker (Davis, Rosenfeld, and Zucker 1975) discuss developing models for visual tasks that are not well specified. For doing work in shape from shading Horn (Horn 1970) develops a model of how certain kinds of surfaces appear given orthographic projections and specified reflectance functions. (Witkin 1981) develops a model of how textured surfaces appear when projected. He uses an isotropy assumption for edge orientations to do this.

A good survey of work on modeling for computer vision was done by Ahuja and Schachter (Ahuja and Schachter 1981). This work focuses on modeling textured images with autocorrelation matrices but many other models are mentioned. Rama

Chellappa in (Chellappa 1981b) discusses autocorrelation texture models and using Markov random fields to classify texture. Markov Random Fields are themselves low level vision models. They are usually derived from ad hoc criteria or analogical reasoning. Currently Paul Chou (Chou 1987) is examining these issues at the University of Rochester.

2. Previous Work on Edge Detection

Edge detection has been one of the earliest and most important tasks attempted by computer vision systems. Often edge detection is described as a problem in image reconstruction. *Edge detection* discovers the contrast in a region of the ideal image when there is an edge between two constant intensity regions of the ideal image (Haralick 1986b). Since I am not motivated by image reconstruction this task is not of particular interest for me. However often edge detection algorithms are used for boundary detection. The idea is to accept as boundaries the windows that the detector considers to have high contrast.

On one dimensional data the detectors presented in chapter 3 when thresholded are equivalent to a thinned contrast detector thresholded by contrast. Thus ignoring issues of *a priori* threshold selection and evidence combination, the techniques for boundary pixel detection are functionally equivalent to matched filter techniques (Boie, Cox, and Rehak 1986). Indeed an optimal operator for 1 dimensional boundary pixel detection for the step-edge model is linear (Shen and Castan 1986). However this equivalence fails to hold on 2 dimensional data. Two dimensional data requires a non-linear operator for optimal step edge prediction.

Hueckel (Hueckel 1971) considers linear operators over a disc on the image. He models edges as step edges with 0 curvature uniformly distributed about a disk. Figure 87 demonstrates the possible regions in this model.

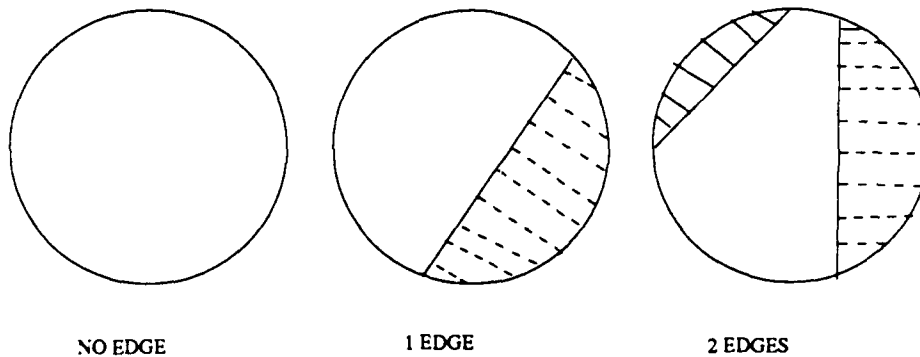


Figure 87 Hueckel's model

His functions only use certain specific Bessel coordinates (of an integral Fourier transform) that he determined are useful for edge detection. He takes into account somewhat the possibility of two randomly distributed edges in the region. A later work generalized this model to handle a thin region with a different graytone between two regions that meet (Hueckel 1973) as in figure 88.

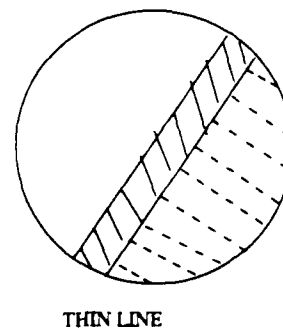


Figure 88 Hueckel's 2nd model

Thus lines on paper and the boundaries of Lambertian surfaces are modeled.

At MIT starting with Marr (Marr 1982) there has been concentration on zero crossing edge detection. The edge detectors they locate edges at zero crossings of a Laplacian of a Gaussian. Poggio, (Torre and Poggio 1986) and Beddoes and

Lunscher (Lunscher and Beddoes 1986b) (Lunscher and Beddoes 1986a) show that such an edge detector is an approximation to a spline operator that has locally maximal output at edges. Such an operator has been shown always to create closed contours. Such operators are related to optimal linear operators developed by Canny (Canny 1986) and Boie, Cox and Rehab (Boie, Cox, and Rehak 1986). Since the objective of these operators is biological verisimilitude while the objective of the operators in this dissertation is low error rates and flexibility of use it is hard to compare zero-crossing operators with those in this paper. Clearly zero-crossing operators make more mistakes than the ones presented in this paper. However those mistakes result in closed contours, if far too many of them and not at the boundaries (Canny 1983). For some applications this may outweigh the advantages of accuracy.

Canny (Canny 1986) studies linear operators for edge detection. Like many who work on linear operators he considered an edge detector to be good if it reported strongly when there was an edge there and *did not report* when there was no edge. He also wanted a detector that only reported once for an edge. He found that these constraints conflict when one is limited to convolution edge detectors (such behavior arises naturally for the boundary detectors in this paper). His primary work on this topic was with a one dimensional step edge model. He derived a convolution operator that was similar to a zero crossing operator. He also discusses extending the operators defined for one dimensional images to two dimensional images, and when oriented operators are desirable. These operators have had notable success on real images. Chapter 3 section 3.2 derives boundary detectors for step edges but does not constrain the functional form of the edge detector. Thus the edge detectors from that chapter should perform at least as well as Canny's detector.

Haralick has developed a body of low-level vision work using his facet model. Haralick (Haralick 1980) describes a likelihood generator for detecting non-edges

similar to the early likelihood generator developed by Sher (Sher 1985a). Later work (Haralick 1984) (Huertas and Medioni 1986) approximates a window on the image with a cubic function. Least squares are used to fit the image with the best cubic. Then zero crossings in the second derivative of the cubic that correspond to large first derivatives of the cubic are reported as edges. This work corresponds to a large class of work on edge detection that use surface fitting.

The operators in this dissertation consider all the possible surfaces rather than the best fit. However surfaces far from the best fit contribute little to the estimate of the likelihood of an edge. Surfaces close to the best fit return results similar to that of the best fit. Thus only considering the best fit may be an effective numerical simplifying assumption for approximating the likelihood. More work can be done on the relationship between surface fitting and likelihood generation.

Nalwa (Nalwa and Binford 1986) fits regions in the intensity image with surfaces that are planar, cubic two dimensional polynomials in the horizontal and vertical directions, and $intensity = \tanh^{-1}(x)$ rotated in the image coordinates. Step edges are modeled with tanh surfaces. The attempts to fit are ordered by increasing computational complexity. For real images his operator is considered the best state of the art edge detector extant (Feldman 1987). Nalwa's operator was used for comparison with the ones presented here.

There have been some sophisticated models for edge detection that model the image as a Markov chain with parameters varying as a Markov chain. A filter that removes noise with such a model is a Kalman filter. Chellapa (Zhou, Chellappa, and Venkateswar 1986) detects edges using such a filter on 1 dimensional slices of the image.

Another approach to edge detection is to simulate parts of the human early visual system. Zero crossing operators were originally motivated by this argument

since it was found that there were cells in the human early visual system that compute zero crossings at various frequencies (Marr 1982). Other work that seriously studies the human early visual system was by Fleet (Fleet 1984) on the spatio-temporal properties of center-surrounds operators in the early human visual system. This dissertation is not concerned with the structure of the early human visual system since its goals are to perform a task best as possible rather than as human-like as possible.

3. Previous Work on Template Matching

There has been some theoretical work on template matching. Ballard (Ballard 1981) introduces the Hough transform for template matching of the form described here. Li and Dubes (Li and Dubes 1985) analyzes a problem similar to chapter 4's problem and generates a likelihood ratio test for deciding whether a template is matched. Their domain is recognizing structures in thresholded LANDSAT data. Thus both 0's and 1's are considered significant. Their operators could distinguish matched 0's from matched 1's and count 0 and 1 matches separately. They analyzed this problem using classical hypothesis testing and measure the power of their tests empirically.

Brown (Brown 1982) discusses matches generated by objects that are near the object looked for (extraneous matches in my notation). It suggests a technique for minimizing these matches. Experimental work on this topic is discussed by Brown, Curtis and Sher (Brown, Curtis, and Sher 1983b). Other work on the accuracy of template matching (implemented by the Hough transform) for line finding is available. (Shapiro and Iannino 1979) (Maitre 1986). These works use a model of discretization noise to determine how the space of possible lines in the image can be discretized to get consistent results from the template matching. Isberg and Morris (Isberg and Morris 1986) and Wernick and Morris (Wernick and Morris 1986) do a

classical statistical analysis of template matching when intensities are being matched under low light levels. Their problem is to decide how many photons should be observed before the decision is made.

4. Previous Work on Decision Theory and Artificial Intelligence

Low-level vision presents decision problems whereby a choice must be made among several possibilities from observed evidence. Classical statistics does not address this kind of problem.

Classical hypothesis testing involves constructing decision operators that are guaranteed to have less than a specified false positive rate. Such operators are useful when applied to scientific investigation since a false positive in a scientific investigation is a disaster. The false positive rate is described in statistics as the *size* of the operator. Once the size is set statisticians go about minimizing the false negative rate. It is not easy to apply this work to multiple hypothesis decision theory, though some work has been done in this direction (Ferguson 1967). Thus classical hypothesis testing is not an appropriate theory for most of low-level vision.

A Bayesian approach is taken in this thesis. Bayesian decision theory allows one to make decisions between many hypotheses taking into account the cost of mistakes. Bayesian theories require a large amount of information to be used. The questions about the applicability of Bayesian theories revolve about the availability of this information. When the information required by a Bayesian theory is available any rational person would agree that a Bayesian theory should be used (Good 1983b). In low-level vision large amounts of data and intricate models are available. Hence a Bayesian theory, such as the one presented in this paper, is a plausible decision theory for low-level vision.

For many applications maximum entropy is used to approximate exact probabilities that can not be supplied by the user (Barth and Norton 1986). Here isotropy assumptions equivalent to maximum entropy are used to simplify overly complex models into tractable ones and to model the scenes for which all the available models fail. Maximum entropy has been used to simplify image analysis in (Andrews and Hunt 1977a), (Grandy 1985), (Skilling and Gull 1985), (Frieden 1985), (Herman 1985), and (Jaynes 1985).

Outside artificial intelligence applications there seems always to be a preferred model for interpreting observed data. Thus the problem of combining evidence generated by several models does not come up. The two most popular techniques for combining information from several models are the Bayesian techniques (Levitt 1985) (Pearl 1985a) and techniques that use Dempster-Shafer statistics (Lowrance and Garvey 1983) (Wesley and Hanson 1982) (Reynolds, Strahman, and Lehrer 1985). Dempster-Shafer techniques for evidence combination fail to satisfy the desiderata in section 5.1. (Kyburg 1987) (Grosz 1985), and (Hummel and Landy 1985) show that Dempster-Shafer reasoning is a restriction on convex Bayesian interval reasoning. That restriction results from an independence assumption encoded in the combination rules. This independence assumption allows a Dempster-Shafer system to combine information from two models without any prior knowledge of the probability of the two models. However Dempster-Shafer reasoning leads to incorrect conclusions in domains where this assumption does not hold. Thus Dempster-Shafer reasoning was rejected as inadequate for my research project.

5. Previous Work on Probabilistic Relaxation

The algorithms developed in chapters 3 and 4 generate input for higher level vision algorithms. Often the higher level vision algorithms use probabilistic

relaxation or neighborhood constraints. One such class of algorithms is Markov random field analysis algorithms. Such algorithms use likelihood ratios to describe the observed data. For example the algorithm in section 3.2 is used in (Chou 1987). Section 4.6 generates a Markov random field for object recognition from boundary pixels. Section 5.6 shows how maximum entropy and the evidence combination rules might improve the performance of a Markov random field algorithm.

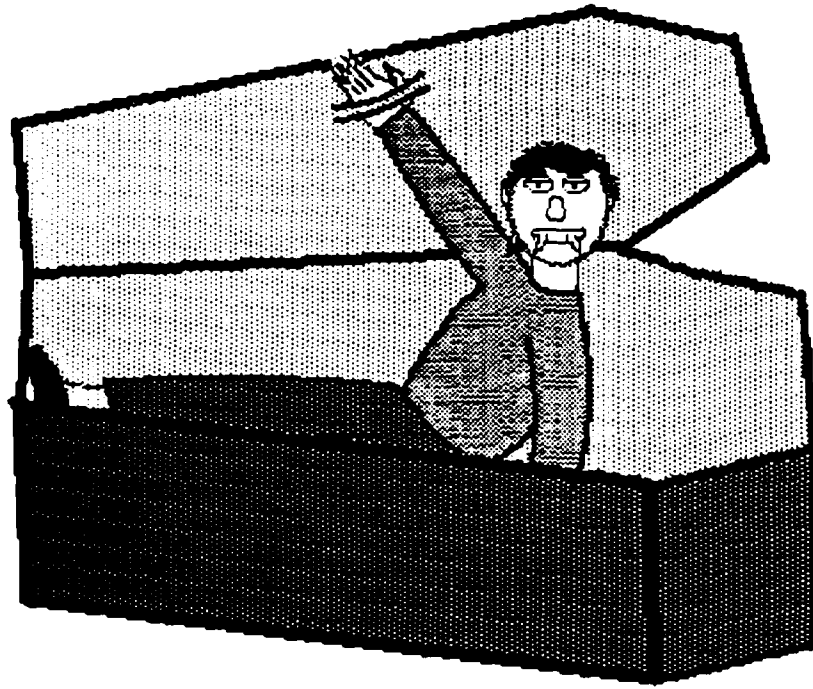
There has been great interest in using Markov random fields for low level vision. Markov random fields are used for image reconstruction in (Derin et al. 1984) (Geman and Geman 1984) and (Marroquin 1985). Markov random fields are used for segmentation in (Hansen and Elliot 1982) (Chou 1987) (Derin and Elliott 1987) (Cohen and Cooper 1987) Markov random fields are a popular model for textured images (Ahuja and Schachter 1981) (Chellappa 1981b) (Derin and Elliott 1987).

Probabilistic relaxation was an early approach to taking knowledge about the structure of neighborhoods of features into account. Smoothness and thinness of boundaries were often encoded as parameters to a probabilistic relaxation. Probabilistic relaxation accepted probability distributions over feature labels as input.

Some probabilistic approaches to relaxation have been discussed in (Zucker, Hummel, and Rosenfeld 1975) (Rosenfeld, Hummel, and Zucker 1975) (Hanson, Riseman, and Glazer 1980). None of these approaches have been particularly successful. One problem with these approaches is that their purpose was to derive an estimate rather than probability distributions. Another problem is that the source of probability distributions was not well designed. Because this work was not successful Zucker and Hummel, and Hanson, Riseman and Glazer abandoned probabilistic approaches to relaxation (Hummel and Zucker 1980) (Hanson, Riseman, and Glazer 1980). However for the high level vision task of labeling

segments probabilistic relaxation has been successful. (Feldman and Yakimovsky 1974) used Bayesian reasoning to constrain the labeling of segments. Probabilistic relaxation has been subsumed by the work on Markov random field models.

Chapter 7



Final Thoughts

CHAPTER 7

Final Thoughts

This chapter concludes my dissertation. It finishes with a summary of what has been shown in the preceding chapters and a description of how this work can be extended in the future. Thus this chapter acts as an overview of the ongoing research program one of whose fruits is this dissertation.

1. Summary

This dissertation presents a paradigm for designing low-level vision operators and systems. First a low-level vision task such as boundary pixel detection or template matching is chosen. Once the task is chosen there is an array of features each with a set of labels one of which is the value of the feature

Chapter 1 shows that for a feature labeling algorithm to be flexible it must return a probability distribution over the feature labels for each feature. These probability distributions can be used by many higher level algorithms that differ in error sensitivities.

Section 1.3 motivates algorithms that yield likelihoods instead of probabilities for feature labels. The advantages of likelihoods are that

- (1) Given priors, Bayes' law yields probabilities for feature labels
- (2) Because likelihoods do not sum to unity they contain one more degree of freedom than probabilities. This degree of freedom is the degree of fit on the

image for the domain model (used to get the likelihoods). This information allows the development of a robust evidence combination theory from probability theory in chapter 5.

- 3) Efficient algorithms are developed that accurately approximate likelihoods. Such algorithms are developed in chapters 3 and 4.
- 4) Markov random field models of vision tasks use likelihoods to represent observed data.

Chapter 2 describes the first step in developing a low-level vision algorithm: modeling the domain. Modeling is described there as making the mathematical connection between the statistics desired by the designer of the feature detector and the statistics supplied by the user. Simplifying assumptions are necessary for constructing tractable models and efficient algorithms. Types of simplifications that are often useful in low-level computer vision are listed in section 2.2. Using some of these simplifications, section 2.3 develops a domain model that yields prior probabilities for boundary pixels given expected values for the areas and perimeters of the object silhouettes.

Chapter 3 develops a likelihood generator for boundary pixel detection. In section 3.1, simplifications from section 2.2 are applied to a realistic imaging and scene model. The resulting model is a facet model (Haralick, 1980) with correlated Gaussian additive noise and linear shift invariant blur. Given this simplified model, section 3.2 derives a class of algorithms for boundary pixel detection. This class of algorithms can be implemented efficiently in hardware that convolves images fast.

The boundary pixel detectors are tuned to a particular level of noise, blur and expected image histogram. When applied to images with these characteristics these detectors error rates are smaller than the error rates of state-of-the-art edge detectors (section 3.5). explains experiments that yield these results on artificial images.

Section 3.5.3 applies the probabilistic boundary pixel detectors to laboratory and aerial images. Section 3.5.4 compares these results with the result of applying established edge detectors to the artificial, the laboratory and the aerial images.

Chapter 4 derives likelihood generators for another task (besides boundary pixel detection). Likelihood generators are derived for template matching. Template matching takes the thresholded output of a feature detector as input and returns probabilities for object instances at positions. To develop a likelihood generator for template matching an error model for the input feature detector is required. Simplifications from section 2.2 are used by section 4.1 to generate a tractable error model for the feature detector. With this model, section 4.2 describes a likelihood generator that is a nonlinear function of a weighted template match. Sections 4.4 and 4.5 show the analysis in chapter 4 applies to more sophisticated forms of template matching. Section 4.6 uses Markov random fields to model overlapping objects and the contribution of object instances to neighboring objects. Thus Markov random field algorithms can be used to yield probabilities for objects.

Chapter 5 shows how specialized likelihood generators such as those designed in chapters 3 and 4 can be combined into more robust likelihood generators. The combination rules satisfy the following desiderata.

- (1) The combined detector takes *a priori* knowledge about the reliability of detectors into account. (Dempster-Shafer combination rules do not.)
- (2) The combined detector takes *a posteriori* knowledge about the reliability of the detectors into account. Thus if evidence in the image indicates that a specialized detector's model does not apply then the result of that detector does not affect the output of the combined detector.
- (3) The time required to execute the combined detector is linear in the times required to execute the specialized detectors.

Section 5.3 uses an identity of probability theory, equation 28, for combining likelihoods conditioned on two different models. If all the elements of this equation are supplied, likelihoods of feature values given the disjunction of two models can be calculated. Thus from likelihood generators for two models a likelihood generator can be derived that works when either of the domain models applies.

Evidence combination using likelihood generators has been applied to the problem of boundary pixel detection in two ways. These results show the effectiveness of evidence combination using likelihoods.

Section 5.8 shows the result of combining detectors derived and tested in chapter 3. Detectors tuned to noise levels of $\sigma=1,4,8,12,16$ are available. On artificial data with noise level σ the combination of the operators tuned to $\sigma=4,8,12,16$ is shown to be nearly as good as the tuned operator. On laboratory images that experiments have shown to have noise with $2<\sigma<6$, the combination of operators with $\sigma=1,4,8$ seems to return better results than any of the individual operators.

The combination of operators with window sizes 5 by 5, 7 by 7, and 9 by 9 was tested and found to be superior to any of the individual operators. The operator tuned to $\sigma=12$ noise in each of the window sizes and the combination thereof was applied to artificial data. The combination retained the low sensitivity to extraneous phenomena such as corners that the 5 by 5 operator has but retained the low false positive rate of the larger operators. The combined noise level operators for the three sizes and the combination thereof was applied to the laboratory images and the combination has superior results to any of the individual operators.

In conclusion, this thesis shows that it is useful to have low-level vision operators that return probabilities. Such operators can be built for several important low-level vision problems and efficiently implemented. Robust operators can be

constructed out of several specialized operators using this approach. This work is upheld by experiments using artificial and laboratory images.

2. Future Directions

Given the results summarized in section 7.1, what further work needs to be done on this approach?

There are many statistics of interest to vision system designers. Most of these statistics are not in a form to be easily supplied by a user of such systems. Examples of these are expected distribution of surface orientations, expected autocorrelation of images, and probability of pixel measuring light from a shadowed area. A derivation of these and other such statistics from primitive statistics would be useful.

The theory of constructing mathematical models for vision problems needs more work. The degree of inaccuracy in resulting from many common forms of simplification is not known. Also the algebraic structure of simplifications to domain models should be studied with an eye towards a complete and consistent theory of simplifications. Finally, here, simplifications are made on an *ad hoc* basis. It would be useful to have a theory for choosing an appropriate simplification based on the circumstances and purpose of the simplification.

Boundary pixel detection can benefit from more sophisticated models. The current model does not handle these phenomena.

- (1) Non additive noise sources such as fog, dirt on film, broken sensors in the CCD array, and dirt, paint or camouflage etc. on objects
- (2) Texture on objects
- (3) Discontinuities in the surface orientation of objects such as corners
- (4) Transparent and translucent objects such as water

- (5) Reflectance functions that vary quickly, such as specular reflectance.
- (6) Discontinuities in the lighting such as that caused by shadows and internal reflection.
- (7) Contextual knowledge, such as that boundaries for closed contours.

Phenomenon 2 requires only a basis to describe the texture. Specialized boundary detectors can be constructed to handle the more common cases of phenomena 3 and 6. However solutions to phenomena 5 and 1 promise to be expensive. Phenomenon 4 has not been extensively addressed by vision researchers. Markov random fields somewhat take into account the knowledge in 7 however there is much more work to be done on this topic.

More work can be done comparing the boundary detectors here against the more common edge detectors, such as those described in 6.2. The choices of window sizes made here for the boundary detectors is *ad hoc*. It would be useful to have a theory for choosing support for a feature detector, balancing loss of accuracy with complexity of modeling and computational efficiency.

A more sophisticated error model for the input to template matchers would be useful. In particular the correlation between failures of the feature detector from local bursts of noise or other unmodeled phenomena should be taken into account. A Markov random field error model is one approach to this problem.

The template matching work in chapter 4 should be implemented and tested on artificial and real data. A proposed experiment is supplied in section 4.7. More sophisticated template matching tasks can be tried after this experiment.

Many other applications would benefit from the approach described in this thesis. Likelihood generators are available for tasks such as texture classification (Owen 1984) Other low-level vision problems such as detecting optical flow, surface

orientation, and structure from flow can benefit from the flexibility that results when probabilities are used. Until now approaches in these areas have tried to derive exact answers such as flow fields or shapes (Horn 1970) (Ikeuchi 1980). But recent analyses (Huang and Tsai 1981) have shown that the numbers for flow derived from these approaches are ill-conditioned. Because a probability distribution is a weaker constraint than an exact answer¹ there may be robust ways to generate a probability distribution for these problems even though there are not robust techniques to get exact answers. Another benefit of this approach for these problems is the evidence combination here allows one to break up the problem of detecting, say optical flow, in the general case to a set of problems where optical flow occurs under specific conditions, such as rigid body motion (Ballard and Kimball 1981) or planar surfaces (Aloimonos and Chou 1985).

Many previous approaches to detecting surface orientation and optical flow use constraints such as smoothness of surface orientation and optical flow fields. Since smoothness is a phenomena of local neighborhoods, a natural way to model such constraints is with a Markov random field. One can simply encode in the field that sharply curved or discontinuous neighborhoods are unlikely. Markov random fields fit naturally with the approach here. A Markov random field acts as a source of prior probabilities about arrays of feature labels. Work progresses on efficient algorithms for labeling arrays of features modeled by Markov random fields (Chou 1987) (Derin et al. 1984).

¹ Consider how much more is said by the statement "The coin came up heads" than the statement "The coin has a 1/5 chance of coming up heads."

Bibliography

Narendra Ahuja and B. J. Schachter, *Image Models*, *ACM Computing Surveys* 13,4 (December 1981), 373 - 398, ACM.

John Aloimonos and Paul Chou, *Detection of Surface Orientation and Motion from Texture: I. The Case of Planes*, 161, Computer Science Department, University of Rochester, January 1985.

H. C. Andrews and B. R. Hunt, *Digital Image Restoration*, 147-187 , Prentice-Hall INC., Englewood Cliffs, New Jersey 07632, 1977

H. C. Andrews and B. R. Hunt, *Digital Image Restoration*, 8-26 , Prentice Hall, INC., Englewood Cliffs, New Jersey 07632, 1977

H. C. Andrews and B. R. Hunt, *Digital Image Restoration*, 187-211 , Prentice-Hall, INC., Englewood Cliffs, New Jersey 07632, 1977

H. C. Andrews and B. R. Hunt, *Digital Image Restoration*, 113-126 , Prentice-Hall, INC., Englewood Cliffs, New Jersey 07632, 1977

D. H. Ballard and O. A. Kimball, *Rigid Body Motion from Depth and Optical Flow*, 70, Department of Computer Science, University of Rochester, November 1981 .

D. H. Ballard, Generalizing the Hough Transform to Detect Arbitrary Shapes, *Pattern Recognition* 13,2 (1981), 111-122.

Dana H. Ballard and Christopher M. Brown, *Computer Vision*, 77 , Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1982

Dana H. Ballard and Christopher M. Brown, in *Computer Vision*, Prentice-Hall Inc., Englewood Cliffs, New Jersey, 1982, 24-30.

Dana H. Ballard and Christopher M. Brown, in *Computer Vision*, Prentice-Hall Inc., Englewood Cliffs, New Jersey, 1982, 125.

Knowledge Stephen W. Barth and Steven W. Norton, Knowledge Engineering Within a Generalized Bayesian Framework, *Proceedings of the Workshop on Uncertainty in Artificial Intelligence*, Philadelphia, August 8-10 1986, 7-15.

James O. Berger. *Statistical Decision Theory*. Springer-Verlag., New York Heidelberg Berlin, 1980

James O. Berger, *Statistical Decision Theory*, 61-86 , Springer-Verlag., New York Heidelberg Berlin, 1980

James O. Berger, *Statistical Decision Theory*, 110-112 , Springer-Verlag., New York Heidelberg Berlin, 1980

Thomas O. Binford, Inferring Surfaces from Images, *Artificial Intelligence* 17.1-3 (August 1981), 205-244, North-Holland Publishing Company.

Robert A. Boie, Ingemar J. Cox, and Pavel Rehak, On Optimum Edge Recognition using Matched Filters, *CVPR*, Miami, Florida, June 1986, 100-108.

Robert C. Bolles, *Verification Vision within a Programmable Assembly System*, Thesis Draft Stanford., September 1976

Chris Brown and David Sher, Hough Transformation into Cache Accumulators: Considerations and Simulations, Tech. Rep. 114, Computer Science Department, University of Rochester, August 1982.

Chris Brown and David Sher, Hough Transformation into Cache Accumulators: Considerations and Simulations, *Proc., DARPA Image Understanding Workshop*, Palo Alto California, September 1982.

C. M. Brown, Bias and Noise in the Hough Transform 1: theory, 105, Department of Computer Science, University of Rochester, June 1982.

Chris Brown, Matt B. Curtis, and David Sher, Advanced Hough transform implementations, *Proc., 8th IJCAI*, Karlsruhe, West Germany, August 1983.

Chris Brown, Matt B. Curtis, and David Sher, Advanced Hough transform implementations, Tech. Rep. 113, Computer Science Department, University of Rochester, March 1983.

John Francis Canny, Finding Edges and Lines in Images, 720, MIT Artificial Intelligence Laboratory, June 1983.

John Canny, A computational Approach to Edge Detection. *PAMI* 8,6 (November 1986), 679-698, IEEE Computer Society.

R. Chellappa, Digital Image Restoration using Conditional Markov Models, 1027. University of Maryland, COmputer Vision Laboratory, Computer Science Center. March 1981.

R. Chellappa, Fitting Markov Random Field Models to Images, 994, University of Maryland, COmputer Vision Laboratory, Computer Science Center, January 1981.

Paul Chou, Multi-Modal Segmentation using Markov Random Fields. *IJCAI*, July 1987.

Fernand S. Cohen and David B. Cooper, Simple Parallel Hierarchical and Relaxation Algorithms for Segmenting Noncausal Markovian Random Fields, *IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-9*, NO. 2 (March 1987), 195-219.

Larry S. Davis, Azriel Rosenfeld, and Steven W. Zucker, General Purpose Models: Expectations about the Unexpected, Tech. Rep. -347, University of Maryland, January 1975.

Haluk Derin, Howard Elliott, Roberto Cristi, and Donald Geman, Bayes Smoothing Algorithms for Segmentation of Binary Images Modeled by Markov Random Fields, *PAMI* 6,6 (November 1984), 707-720, IEEE.

Haluk Derin and Howard Elliott, Modeling and Segmentation of Noisy and Textured Images Using Gibbs Random Fields, *PAMI* 9,1 (January 1987), 39-55, IEEE.

Jerome A. Feldman and Yoram Yakimovsky, Decision Theory and Artificial Intelligence: I. A Semantics-Based region Analyzer, *Artificial Intelligence* 5:1974, 349-371, North-Holland Publishing Company.

Jerome Feldman, Naïve's Operator is Best Extant, Personal Communication, May 1987.

Thomas S. Ferguson, *Mathematical Statistics - a Decision Theoretic Approach*, 284-295, Academic Press., 1967

David J. Fleet, The Early Processing of Spatio - Temporal Visual Information, 84-7, University of Toronto, Research in Biological and Computational Vision, September 1984.

B. Roy Frieden, Estimating Occurrence Laws with Maximum Probability, and the Transition to Entropic Estimators, in *Maximum-Entropy and Bayesian Methods in Inverse Problems*, D. Reidel Publishing Company, Lancaster, 1985

M. A. Furst and P. E. Cairns, *Edge Detection with Image Enhancement via Dynamic Programming*, 263-279, Academic Press INC., March 1986

Stuart Geman and Donald Geman, Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images, *PAMI* 6,6 (November 1984), 721-741, IEEE

I. J. Good, *Probability and the Weighing of Evidence*, Hafner Publishing Company, London, New York, 1950

Irving John Good, Some Logic and History of Hypothesis Testing, in *Good Thinking: The Foundations of Probability and its Applications*, University of Minnesota Press, Minneapolis, 1983, 129-148

Irving John Good, Rational Decisions, in *Good Thinking: The Foundations of Probability and its Applications*, University of Minnesota Press, Minneapolis, 1983, 114

Irving John Good, Which Comes First, Probability or Statistics, in *Good Thinking: The Foundations of Probability and its Applications*, University of Minnesota Press, Minneapolis, 1983, 59-62

W. J. Grandy, Incomplete Information and Generalized Inverse Problems, in *Maximum Entropy and Bayesian Methods in Inverse Problems*, D. Reidel Publishing Company, Lancaster, 1985, 1-20

Benjamin N. Grosz, *An Inequality Paradigm for Probabilistic Knowledge*, AAI, RCA, Los Angeles, August 14-16, 1985

E. R. Hansen and H. Elliot, Image Segmentation Using Simple Markov Field Models, *Computer Vision, Graphics and Image Processing* 20,2 (October 1982), 101-132.

Allen R. Hanson, Edward M. Riseman, and Frank C. Glazer, Edge Relaxation and Boundary Continuity, 80-11, University of Massachusetts at Amherst, *Computer and Information Science*, May 1980

Robert M. Haralick, Edge and Region Analysis for Digital Image Data, *Computer*

NO-A189 041

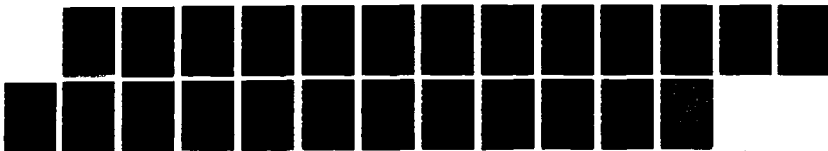
A PROBABILISTIC APPROACH TO LOW-LEVEL VISION(U)
ROCHESTER UNIV NY DEPT OF COMPUTER SCIENCE D B SHER
OCT 87 TR-232 N00014-82-K-0193

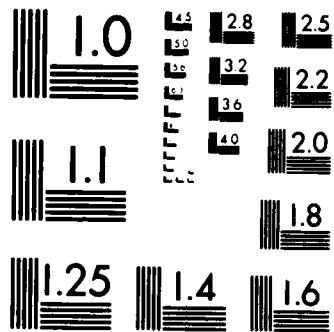
3/3

UNCLASSIFIED

F/G 23/3

NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

Graphics and Image Processing 12(1980), 60-73, Academic Press.

Robert M. Haralick, Ridges and Valleys on Digital Images, *Computer Vision, Graphics, and Image Processing 22*(1983), 28-38, Academic Press.

Robert M. Haralick, Digital Step Edges from Zero Crossing of Second Directional Derivatives, *PAMI 6*,1 (January 1984), 58-68, IEEE.

Robert M. Haralick, Second Directional Derivative Zero Crossing Detector Using the Cubic Facet Model, *Facet Model Image Processing (CVPR Tutorial)*, June 1986.

Robert M. Haralick, The Facet Approach to Gradient Edge Detection, *Tutorial 1 Facet Model Image Processing (CVPR)*, May 1986.

Gabor T. Herman, Application of Maximum Entropy and Bayesian Optimization Methods to Image Reconstruction from Projections, in *Maximum-Entropy and Bayesian Methods in Inverse Problems*, D. Reidel Publishing Company, Lancaster, 1985, 319-337.

Berthold Kaus Paul Horn, *Shape from Shading: A Method for Finding the Shape of a Smooth Opaque Object from One View*, Massachusetts Institute of Technology Department of Electrical Engineering., August 1970

Berthold Kaus Paul Horn, *Robot Vision*, 18-46 , MIT Press., Cambridge Massachusetts, London England, 1986

P. V. C. Hough , Method and Means for Recognizing Complex Patterns, U.S. Patent

3,069,654, 1962.

T. S. Huang and R. Y. Tsai, *Image sequence analysis: Motion estimation*, Springer-Verlag., 1981

Manfred H Hueckel, An Operator Which Locates Edges in Digitized Pictures, *Journal of the Association for Computing Machinery* 18,1 (January 1971), 113-125, ACM.

Manfred H. Hueckel, A Local Visual Operator Which Recognizes Edges and Lines, *J. ACM* 20,4 (October 1973), 634-647, ACM.

Andres Huertas and Gerard Medioni, Detection of Intensity Changes with Subpixel Accuracy Using Laplacian Gaussian Masks, *PAMI* 8,3 (September 1986), 651-664, IEEE.

Robert A. Hummel and Steven W. Zucker, On the Foundations of Relaxation Labeling, Tech. Rep.-80-7, Computer Vision and Graphics Lab, Department of Electrical Engineering, McGill University, July 1980.

Robert A. Hummel and Michael S. Landy, A Statistical Viewpoint on the Theory of Evidence, Tech. Rep.#194 Robotics Report #57, New York University, Courant Institute of Mathematical Sciences, December 1985.

Katsushi Ikeuchi, Shape from Regular Patterns (an Example of Constraint Propagation in Vision), 567, Massachusetts Institute of Technology, Artificial Intelligence Laboratory, March 1980.

Thomas A. Isberg and G. Michael Morris, Rotation-invariant Image Recognition at Low Light Levels, *Journal of the Optical Society of America A* 3(July 1986), 954, Optical Society of America.

E. T. Jaynes, Where do We Go from Here?, in *Maximum-Entropy and Bayesian Methods in Inverse Problems*, D. Reidel Publishing Company, Lancaster, 1985.

Rodney W. Johnson and John E. Shore, *Introduction to Minimum-Cross-Entropy Spectral Analysis of Multiple Signals*, D. Reidel Publishing Company., Lancaster, 1985

Henry E. Kyburg, Bayesian and Non-Bayesian Evidential Updating, *Artificial Intelligence* 31(1987), 271-293, Elsevier Science Publishers B.V. (North-Holland).

Tod S. Levitt, Probabilistic Conflict Resolution in Hierarchical Hypothesis Spaces, *Proceedings: Uncertainty and Probability in Artificial Intelligence*, August 14-16, 1985, 265-272.

Xiaobo Li and Richard C. Dubes, The First Stage in Two-Stage Template Matching, *Pattern Analysis and Machine Intelligence* 7,6 (Nov 1985), 700-707, IEEE.

Ronald Prescott Loui, *A Perspective on Probability in AI*, Internal Memo University of Rochester Departments of Computer Science and Philosophy., 1984

John D. Lowrance and Thomas D. Garvey, Evidential Reasoning: An Implementation for Multisensor Integration, 307, SRI International Artificial Intelligence Center, Computer Science and Technology Division, December 1983.

Wolfram H. H. J. Lunscher and Michael P. Beddoes, Optimal Edge Detector Design II: Coefficient Quantization, *Pattern Analysis and Machine Intelligence* 8,2 (March 1986), 178-187, IEEE.

Wolfram H. H. J. Lunscher and Michael P. Beddoes, Optimal Edge Detector Design I: Parameter Selection and Noise Effects, *Pattern Analysis and Machine Intelligence* 8,2 (March 1986), 164-177, IEEE.

Henri Maitre, Contribution to the Prediction of Performances of the Hough Transform, *PAMI* 8,3 (September 1986), 669-674, IEEE.

David Marr, *Vision*, W. H. Freeman and Company., New York, 1982

Jose Luis Marroquin, Probabilistic Solution of Inverse Problems, Tech. Rep. 860, MIT Artificial Intelligence Laboratory, September 1985.

J. Marroquin, S. Mitter, and T. Poggio, Probabilistic Solution of Ill-Posed Problems in Computational Vision, *Proceedings: Image Understanding Workshop*, December 1985, 293-309. Sponsored by: Information Processing Techniques Office Defence Advanced Research Projects Agency.

Salvatore D. Morgera, Toward a Fundamental Theory of Optimal Feature Selection: Part II-Implementation and Computational Complexity, *PAMI* 9,1 (January 1987), 29-38, IEEE.

Vishvjit S. Nalwa and Thomas O. Binford, *PAMI* 8,6 (November 1986), 679-698, IEEE Computer Society.

Ahmed M Nazif and Martin D. Levine, Low Level Image Segmentation: An Expert System, *Pattern Analysis and Machine Intelligence* 6,6 (Sept 1984), 555-577, IEEE.

Art Owen, A Neighbourhood-based Classifier for LANDSAT Data, *The Canadian Journal of Statistics* 12,3 (September 1984), 191-200, Statistical Society of Canada.

Theodore Pavlidis, Personal Communication, March 1987.

Judea Pearl, Bayesian Networks: A Model of Self-Activated Memory for Evidential Reasoning, Tech. Rep. Computer Science Dpt.-850021R-43, Computer Science Dept, University of California, June 1985.

Judea Pearl, Fusion, Propagation, and Structuring in Bayesian Networks, Computer Science Dpt.-850022 R-42, Cognitive Systems Laboratory, Computer Science Department, University of California, June 1985.

G. Reynolds, D. Strahman, and N. Lehrer, Converting Feature Values to Evidence, *PROCEEDINGS: IMAGE UNDERSTANDING WORKSHOP*, December 1985, 331-339. Sponsored by: Information Processing Techniques Office, Defence Advanced Research Projects Agency.

Christian Ronse, *A General Model for Causal Markov Fields*, Provisional Working Document Philips Research Laboratory Brussels., August 1985

Azriel Rosenfeld, Robert A. Hummel, and Steven W. Zucker, Scene Labelling by Relaxation Operations, Tech. Rep.-379, University of Maryland, Computer Science Department, May, 1975.

Laura Sanchis, Finding Gamma Correction Factors, *Internal Memo*, August 1986.

Stephen D. Shapiro and Anthony Iannino, Geometric Constructions for Predicting Hough Transform Performance, *PAMI 1,3* (July 1979), 319-317, IEEE.

Lokendra Shastri, Evidential Reasoning in Semantic Networks: A Formal Theory and its Parallel Implementations, Tech. Rep. 166, Department of Computer Science, The University of Rochester, Rochester NY, September 1985.

Jun Shen and Serge Castan, An Optimal Linear Operator For Edge Detection, *CVPR*, Miami, Florida, June 1986, 109-114.

David Sher and Avadis Tevanian, The Hough Chip, A Custom Integrated Circuit, Tech. Rep. 144, Computer Science Department, University of Rochester, Rochester, NY, November 1984.

David Sher and Avadis Tevanian, The Hough Chip, A Custom Integrated Circuit, *Proc., IEEE Custom Integrated Circuits Conference*, Rochester, NY, May 1984.

David B. Sher, Evidence Combination for Vision using Likelihood Generators, *Proceedings: Image Understanding Workshop (DARPA)*, Miami, Florida, December 1985, 255-270. Sponsored by: Information Processing Techniques Office Defence Advanced Research Projects Agency.

David Sher, Template Matching on Parallel Architectures, Tech. Rep. 156, Computer Science Department, University of Rochester, July 1985.

David Sher, Optimal Likelihood Detectors for Boundary Detection Under Gaussian Additive Noise, *IEEE Conference on Computer Vision and Pattern Recognition*, Rochester, NY, April 1986.

David Sher, Optimal Likelihood Detectors for Boundary Detection Under Gaussian Additive Noise, *IEEE Conference on Computer Vision and Pattern Recognition*, Miami, Florida, June 1986, 94-99.

David Sher, Appropriate and Inappropriate Estimation Techniques, *AAAI Workshop on Probability and Uncertainty in Artificial Intelligence*, Philadelphia, Pennsylvania, August 1986, 261-266.

David Sher, Expert Systems for Vision Based on Bayes' Rule, *Computer Vision, Graphics and Pattern Recognition*, April 1986, 114-116.

David B. Sher, Advanced Likelihood Generators for Boundary Detection, TR197, University of Rochester Computer Science Department, Rochester NY 14627, January 1987.

David B. Sher, Evidence Combination Based on Likelihood Generators, TR192, University of Rochester Computer Science Department, Rochester NY 14627, January 1987.

David Sher and Myra Van Inwegen, *The See++ Image Processing System*, University of Rochester, Computer Science Department., Rochester NY 14627, December? 1987

Haim Shvaytser and Shmuel Peleg, A New Approach to the Consistent Labeling

Problem, *CVPR 85*, August 1985, 320-326.

John Skilling and S. F. Gull, Algorithms and Applications, in *Maximum-Entropy and Bayesian Methods in Inverse Problems*, D. Reidel Publishing Company, Lancaster, 1985.

Michael J. Swain, Shape From Texture, 926-931, 1985.

Vincent Torre and Tomaso A. Poggio, On Edge Detection, *Pattern Analysis and Machine Intelligence* 8,2 (March 1986), 147-163, IEEE.

Jerry L. Turney, Trevor N. Mudge, and Richard A. Volz, Recognizing Partially Occluded Parts, *PAMI* 7,4 (July, 1985), 410 - 421, IEEE.

Miles N. Wernick and G. Michael Morris, *Image Classification at Low Light Levels*, The Institute of Optics, University of Rochester., Rochester, NY, May 1986

Leonard P. Wesley and Allen R. Hanson, The Use of an Evidential-Based Model for Representing Knowledge and Reasoning about Images in the Visions System, *PAMI* 4,5 (Sept 1982), 14-25, IEEE.

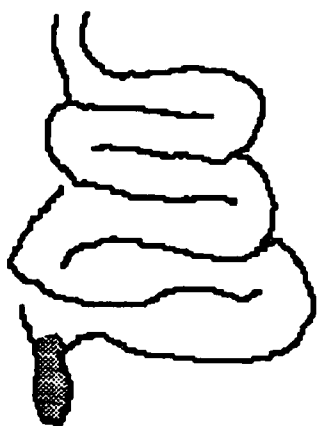
Andrew P. Witkin, Recovering Surface Shape and Orientation From Texture, *Artificial Intelligence* 17,1-3 (August 1981), 17-45, North-Holland Publishing Company.

Y. T. Zhou, R. Chellappa, and V. Venkateswar, Edge Detection Using the Directional Derivatives of a Space Varying Correlated Random Field Model, *CVPR*, Miami,

Florida, June 1986, 115-121.

Steven W. Zucker, Robert A. Hummel, and Azriel Rosenfeld, Applications of Relaxation Labelling, 1: Line and Curve Enhancement, Tech. Rep. 419, University of Maryland, October 1975.

Appendix A



appendix \ə-'pen-diks/ n, pl **-dix-es** or **-dices** \də-'sēz/
[L *appendic-*, *appendix*, fr. *appendere*] (1542) **1 a:** APPENDAGE
b: supplementary material usu. attached at the end of a piece of writing
2: a body outgrowth or process; *specif:* VERIFORM APPENDIX

Glossary

APPENDIX A

Glossary

This work introduces lots of terminology, much of which is new to its expected audience. Hence here is the glossary for quick reference to the more obscure terminology. The definitive references for these terms are in the text and can be found with help from the index of defined terms.

Approximation Assumption

The approximation assumption is a simplifying assumption that allows an algorithm to use an approximation of a function instead of the function itself. Since real-valued functions can not be computed by a computer only approximated, this assumption is very useful. The discretization assumption is a common approximation assumption.

See Also: discretization assumption

Basis

For certain low-level vision problems a label for a feature represents a set of ideal windows that with noise generate the observed data when the label is present. For example in the step edge model the label that represents no boundary passing through a window is the set of constant intensity windows. A basis for a label is a small set of windows linear combinations of which generate the label's windows. A basis for the set of constant intensity

windows is the unit intensity window. Note that this definition of basis is **not** the linear algebra definition of basis. In linear algebra a basis is called a spanning set.

See Also: label

Bayesian Feature Detector

A Bayesian feature detector is an algorithm that takes an observed image and returns a probability distribution for the labels of each feature. Chapters 3 and 4 are about building Bayesian feature detectors.

See Also: feature

Boundary Pixel

A boundary pixel measures light reflected from two objects. A boundary between two objects in an image passes through a boundary pixel.

Corner Pixel

A corner pixel can either be a pixel that measures light from three or more objects or a boundary pixel whose angle is not well defined like the corner of a square. For the most part the corner pixels referred to in this dissertation are pixels that measure light from three or more objects.

Designer

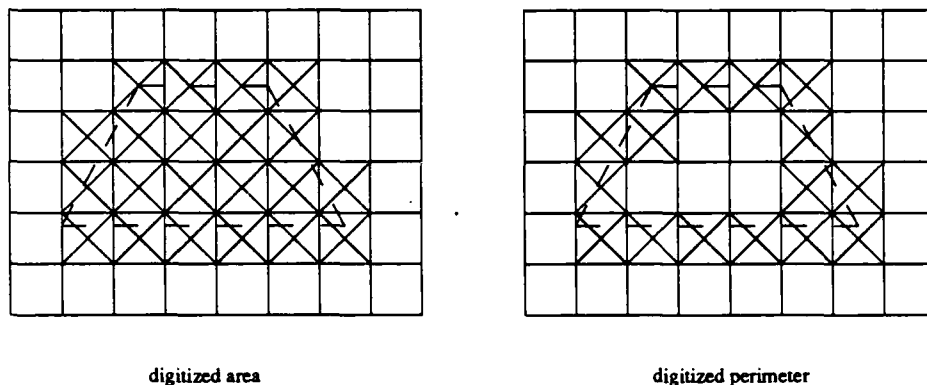
The designer of a low level vision system uses a domain model and primitive statistics from the user to derive required statistics. From required statistics he constructs a Bayesian feature detector.

See Also: domain model, primitive statistic, required statistic, Bayesian feature

detector

Digitized Area

The digitized area of a region on an image is the number of pixels that contain at least part of the region. Figure 10 shows an example of a digitized area.



Object outline is shown by dashed lines.

Crossed boxes are the pixels included in the digitized area and perimeter respectively.

Figure 10 Example of Digitized Area and Perimeter (Reprised)

See Also: digitized perimeter

Digitized Perimeter

The digitized perimeter of a region are the pixels that the boundary of the region passes through. Figure 10 shows an example of a digitized perimeter.

See Also: digitized area

Discretization Assumption

A discretization assumption assumes that events occur only at discrete intervals. Such an assumption is that before noise the gray-level of a pixel is integral. Another such assumption is that a boundary that passes through a

pixel passes through the center of a pixel.

See Also: approximation assumption

Disjoint

Two domain models are disjoint if the probability of all their assumptions being simultaneously true is 0. Examples of disjoint models in this dissertation are a model that assumes $\sigma=4$ and a model that assumes $\sigma=8$, and a model that assumes a corner occurs in a window and a model that assumes no corner occurs in a window.

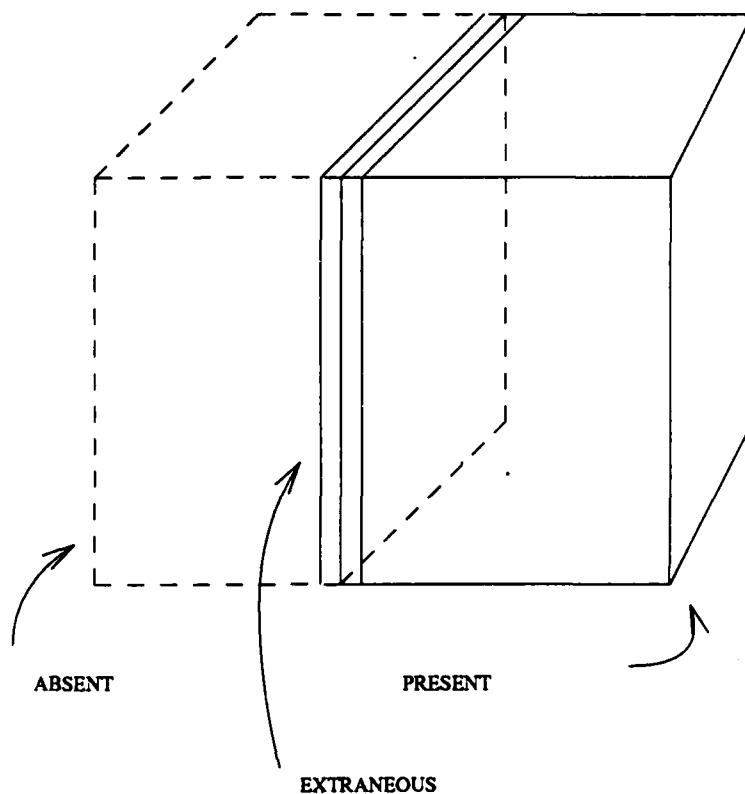
See Also: domain model

Edge Detection

Edge detection is the task of taking a window on an image and estimating what the contrast across a boundary would be if there was a boundary in the image. A constant intensity window could only be a low contrast boundary. A high estimate of contrast is good evidence for the presence of a boundary. Thus edge detectors are sometimes used for boundary detection.

Extraneous Detection

If an object is absent from the scene but another object causes some features to have the same labels as they have when the object is present in the scene, these feature labelings are extraneous detections. Figure 89 presents some extraneous features.



Dashed object is not present

Triple line is the extraneous features
from the object on the right.

Figure 89 Source of Extraneous Features

See Also: scene

Facet Model

A facet model is a model that assumes before noise the image intensities are piecewise polynomial in the horizontal and vertical directions. The degree of a facet model is the maximal acceptable degree for the polynomial pieces. Figure 90 illustrates a piecewise polynomial surface.

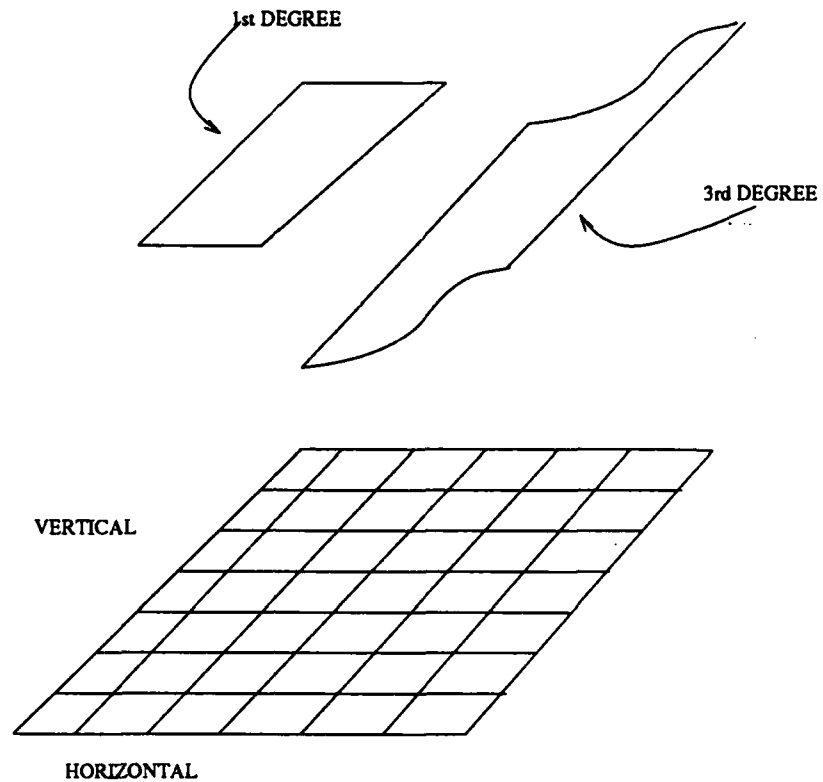


Figure 90 A Piecewise Polynomial Surface

See Also: model

False Negative

If a feature has two labels of the form $\{E, \neg E\}$ then reporting $\neg E$ when E is true is a false negative. For example in boundary pixel detection reporting \neg *boundary* at a pixel when a boundary passes through the pixel is a false negative.

See Also: false positive, feature

False Positive

If a feature has two labels of the form $\{E, -E\}$ then reporting E when $-E$ is true is a false positive. For example in boundary pixel detection reporting *boundary* at a pixel when no boundary passes through the pixel is a false negative.

See Also: false negative, feature

Feature

A feature is a set of labels that a low level vision algorithm must pick from. A feature generally corresponds to a state of nature of interest to the user of the system. Bayesian feature detectors derive probability distributions over the set of labels. An example of a feature is a boundary pixel. In this example the feature is the set of labels, $\{boundary, -boundary\}$, that can be assigned to a pixel. Each pixel has its own feature.

See Also: Bayesian feature detector, boundary pixel, label

Image

The image is the observed data in a vision problem. Usually it is an array of pixels, each measuring the number of photons hitting a light sensitive area. The vision problem can be characterized as an attempt to find out where those photons came from.

Isotropy

Isotropy simplifies a model by assigning equal probabilities to a set of states of nature. As an example section 2.3 assumes that the probability of an object

centered within a pixel is equal for all pixels. Other isotropy assumptions are that the probability of an edge is equal in all directions.

Label

A label is a state of nature. A feature is a set of labels. An example of a label is *boundary*. This label can be assigned to any pixel in the image when doing boundary pixel detection.

See Also: feature

Likelihood

If the observed data is o , the likelihood of a label l for feature f under domain model M , $L_f(o | l \& M)$ is $P(o | f=l \& M)$. Likelihoods and priors taken together determines the probability distribution for f 's labels under M using equation 2 and the probability $P(o | M)$. $P(o | M)$ shows the support in the observed data for M , and thus is useful when deciding between domain models.

See Also: priors

Likelihood Generator

A likelihood generator is an algorithm derived from a domain model that takes observed data o as input and outputs the likelihoods of the labels for a feature f or an array of features f_i . For example a boundary pixel likelihood generator returns likelihoods for *boundary* and \neg *boundary* for the pixels of an image. With priors a likelihood generator can be used by a Bayesian feature detector. Likelihood generators are also used by Markov random fields.

See Also: feature, label, Bayesian feature detector

Marginal Probability Distribution

The task of boundary detection is to find which pixels of an image a boundary passes through. A Bayesian feature detector for this problem returns a

probability distribution over segmentations of an image. If instead a Bayesian feature detector returns a probability distribution for a boundary passing through a single pixel it is reporting a marginal probability distribution because for any particular pixel there are many segmentations that have a boundary passing through it and many segmentations that do not. Knowing many marginal probability distributions does not imply knowing the entire distribution. Thus knowing the probability of a boundary at each pixel is insufficient to determine the probability distribution over segmentations without further assumptions.

See Also: feature

Maximum Entropy

The entropy of a probability distribution, $p(x)$ is shown in expression 30.

$$\int p(x) \log p(x) dx \quad (30)$$

The uniform distribution has the highest entropy. Most of the work on maximum entropy maximizes entropy subject to a constraint. In this paper maximum entropy is used to justify using a uniform distribution.

Domain Model

A domain model is a set of assumptions that derive from primitive statistics the required statistics yielding a Bayesian feature detector. Many of these assumptions are simplifying assumptions that reduce the complexity of the world described by the user.

See Also: Bayesian feature detector, primitive statistics, required statistics

Noise

Noise is a random phenomena that prevents pixels from accurately describing the light reflected off of objects. Noise can result from electronic or quantum mechanical effects. In this dissertation the noise is always modeled by a normally distributed mean 0 variable being added to the reflected light.

Occlusion

Object 1 occludes object 2 when the light reflected from object 2 that would hit the camera is blocked by object 1. An object is completely occluded if none of the light reflected from it gets to the camera. It is partially occluded if only some of the light reaches the camera.

Orthographic Projection

An image is under orthographic projection if it is taken with an ideal camera of infinite focal length. Orthographic projection is approximated by using a long range lens.

Physical Probability Distribution

This distribution is the real world (true) distribution over feature labels. In section 5.5 the principle of maximum entropy in combination with the evidence combination rules developed in sections 5.2 and 5.3 to derive physical probability distributions from probability distributions conditioned on a domain model.

See Also: feature, domain model, maximum entropy

Posterior Probabilities

A posterior distribution is the probability distribution for a feature's labels conditioned on the observed evidence (in vision the image). Bayes' Law (equation 2) uses likelihoods and prior probabilities to derive the posterior distribution.

See Also: prior probabilities, likelihood

Power

The power is 1 minus the false negative rate for a test. Classical hypothesis testing tries to select a test that maximizes power for tests of a specified size.

A test that always returns positive has a power of 1 but a size of 1 too.

See Also: false negative, size

Primitive Statistic

A primitive statistic is a statistic that the user of a low-level vision system can easily supply. Section 2.3 assumes that the user can supply the expected perimeter and area of objects' projections in the scene. Thus the expected perimeter and area of objects' projections are typical primitive statistics.

See Also: required statistic, user

Prior Probability

The prior probability distribution is the probability distribution one has for a feature before one observes any data. Given likelihoods, Bayes' law 2 applied to a prior distribution yields a posterior distribution.

See Also: posterior distribution, likelihood, Bayes' law

Required Statistic

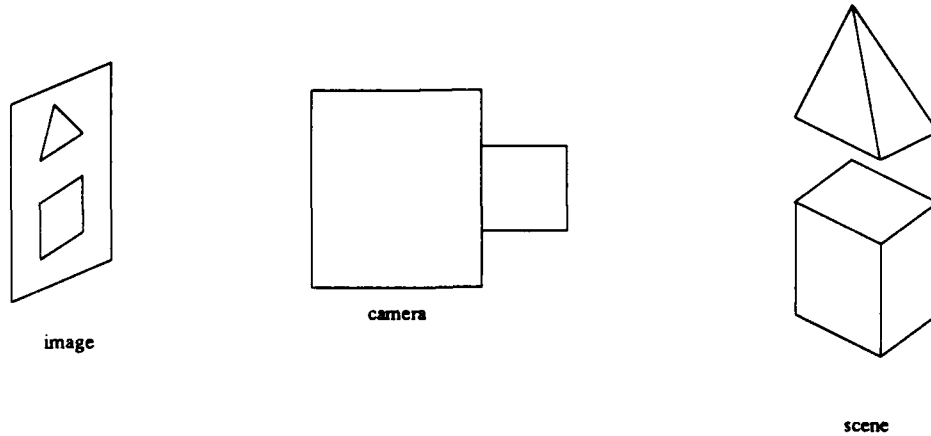
A low-level vision system designer uses a set of required statistics to derive an algorithm for feature detection tuned to the domain model a user supplies. The required statistics are derived from primitive statistics, using the model. In section 3.2 the standard deviation of the noise in the image is a required statistic. In section 4.3.2 the false positive rate as a function of the threshold on a feature detector is a required statistic.

See Also: primitive statistic, designer, user

Scene

The scene is what the camera is pointed at. Figure 5 is an example of a scene.

Figure 5 Definition of Image and Scene (Reprised)



See Also: image

Separable

In the context of this dissertation a function $f(a,b)$ is separable if there are two functions $g(a)$ and $h(b)$ such that $f(a,b)=g(a)\ominus h(b)$ where \ominus is an easy operation such as addition or multiplication. For example if f is described by equation 31 then f is separable.

$$f(a,b)=\int e^{a+b} \partial a \partial b$$

$$f(a,b)=g(a)g(b) \tag{31}$$

$$g(a)=\int e^a \partial a$$

Size

The size of a test is the probability of a false positive from the test. In classical hypothesis testing the size of a test is set to be less than a specified amount such as 0.05 or 0.01. Given a size for a test the power of the test should be maximized. The test that always returns negative has a size of 0.

See Also: false positive, power

User

A user of a low level vision system needs probability distributions for a set of features such as a probabilities of boundaries passing through the pixels of an image. The user supplies primitive statistics and a domain model. From these the designer derives a Bayesian feature detector.

See Also: designer, domain model, primitive statistic, Bayesian feature detector

Window

A window on an image is a subset of the pixels of an image. Usually windows are all the pixels in a rectangle, though some have used other shapes. As an example in a 100 by 100 image the pixels with coordinates (2,3) , (2,4) , (3,3) , and (3,4) is a 2 by 2 window.

See Also: image

END

DÄTE

3-88

DTIC