

To the Graduate Council:

I am submitting herewith a thesis written by David Julian Bryant entitled "Evaluation of Edge Operators Using Relative and Absolute Grading." I recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Engineering, with a major in Electrical Engineering.

Don Bouldin

Don W. Bouldin, Major Professor

We have read this thesis and
recommend its acceptance:

R. C. Gonzalez

muSoogl

Accepted for the Council:

LE Evans

Vice Chancellor

Graduate Studies and Research

Thesis
78
B787
cop. 2

EVALUATION OF EDGE OPERATORS
USING RELATIVE AND
ABSOLUTE GRADING

A Thesis
Presented for the
Master of Engineering
Degree
The University of Tennessee, Knoxville

David Julian Bryant

December 1978

1371458

ACKNOWLEDGEMENTS

The author is very grateful for the invaluable guidance, patience and helpful advice of Dr. Don W. Bouldin, committee chairman, as well as committee members Dr. Ernest L. Hall and Dr. R. C. Gonzalez.

Additional special recognition is given to my parents for their support and encouragement throughout my education.

Appreciation is also expressed to those fellow students who have supplied additional assistance, especially Elaine Pressly, Gene Crossley and Richard Bryan, Gale Slutzky for graphics assistance, and the staff of the Image and Pattern Analysis Laboratory of the University of Tennessee.

ABSTRACT

The purpose of this study was to develop means for numerically evaluating edge detectors. One approach, termed relative grading, involves comparison of the operator's output to the consensus decision of other operators. Relative grading offers a method of eliminating operator and thresholding noise, as well as determining the operator with the least noise content.

The second approach, named absolute grading, compares any operator to a manually constructed key or target scene. This key is structured to represent the principle edge or edges in the original scene. Absolute grading scores operators on their ability to detect as much of this key edge and as little of the rest of the scene as possible.

Both these techniques provide feedback information that can be used to improve threshold levels, system parameters, and guide operator improvement. They exhibit great potential as valuable tools in any edge detection application.

Six operators were tested by both grading schemes. They were the Sobel, Robert's gradient, Robert's square root gradient, gradient, range and Frei-Chen operators. Relative grading revealed the superiority of the Sobel operator in various detection environments. Several operator improvement schemes, including single point elimination, association and linking demonstrated their ability to enhance operator output. Additionally, absolute grading, coupled with a specially selected image test set indicated overall excellence by the Robert's square root gradient in locating selected key edges.

TABLE OF CONTENTS

CHAPTER	PAGE
I. INTRODUCTION	1
II. FUNDAMENTALS OF EDGE DETECTION	3
Gradient Operators	6
Three-By-Three Operators	9
Obtaining Directional Information	18
Grading	20
III. RELATIVE GRADING	24
Thresholding	24
Relative Grading	27
IV. ABSOLUTE GRADING	33
Operator Improvement	40
V. TYPICAL IMPLEMENTATION AND RESULTS	48
Relative Grading Results	51
Absolute Grading Results	62
VI. CONCLUSIONS	84
LIST OF REFERENCES	85
APPENDIX	88
VITA	121

LIST OF TABLES

TABLE	PAGE
1. Relative Grading Results. "No" Vote Threshold = 2 Threshold Level = 10%	53
2. Relative Grading Results. "No" Vote Threshold = 3 Threshold Level = 10%	56
3. Relative Grading Results. "No" Vote Threshold = 2 Threshold Level = 15%	58
4. Absolute Grade of Operator Test Set	65
5. Absolute Grade of Sobel Operator vs. Threshold Level	66
6. Absolute Grade of Sobel Operator vs. Threshold Level for Relative Change Transformation	69
7. Results of Single Point Elimination	71
8. Association Improvement of Sobel Operator	73
9. Linking Improvement of Sobel Operator	75
10. Absolute Grade of Operator Test Set "Kodak Girl"	81
11. Absolute Grade of Operator Test Set Building Scene	82
12. Average Absolute Grade of Operator Test Set	83

LIST OF FIGURES

FIGURE	PAGE
1. Ideal Edge Crossections	4
2. A Corrupted Edge Superimposed Over an Ideal Step Edge	5
3. Directional Differences	8
4. True Laplacian Mask	11
5. Lowpass Masks	11
6. Typical Directional Masks	13
7. Examples of Specially Designed Templates	15
8. Frei-Chen Orthogonal Templates	16
9. The Two Gradient Masks Used in the Sobel Operator	17
10. Arrangement of Pixels for Davies Operator	19
11. Confusion Matrix for Edge Detection Process	22
12. Typical Directional Pattern of an Edge	42
13. Test Scene	49
14. Operator Test Set Output	50
15. Threshold at 10%	52
16. Relative Grading Results With "No" Vote Threshold of Two and Threshold Level at 10%	55
17. Relative Grading Results With "No" Vote Threshold of Three and Threshold Level at 10%	57
18. Relative Grading Results With "No" Vote Threshold of Two and Threshold Level at 15%	59
19. ROC Curves for Relative Grading Systems of Tables 1, 2, and 3	61
20. Sample Keys	63

FIGURE	PAGE
21. Absolute Grade of Sobel Operator Vs. Threshold Level	67
22. Relative Change Transformation of Sobel Operator	70
23. Results of Association	74
24. Linking Results	76
25. Plot of Absolute Grade Vs. Length Threshold of Linking Process for Data in Table 9	77
26. Additional Test Scenes	79
A-1. Alignment of Pixels and Labelling Scheme of Davies Method, After Necessary Rotation	89

CHAPTER 1

INTRODUCTION

Digital image processing involves computer manipulation of specially coded versions of physical scenes, pictures or images. Most frequently, picture processing methods are directed in either of two directions: improvement of pictorial information for human interpretation, often termed image enhancement, and processing of scene data for automated machine perception.

Edge detection processes fall within the realm of the latter area, namely image understanding. Location and identification of objects within images is often carried out by searching for their boundaries or edges. Often images are more concisely represented by or more simply handled from their edges. In multispectral imagery, such as joint studies of optical and radar views of identical scenes, edge detection is frequently the only means of locating coincident objects of interest.

For these and many other applications, it is necessary to be able to accurately detect edges. This is accomplished by employing some edge detection scheme, usually a mathematical operator.

The problems of edge operator selection and performance evaluation are often encountered in digital image processing research. This thesis seeks to derive and develop methods for solving these and other related dilemmas.

Chapter 2 contains a background survey of edge detection fundamentals and outlines several commonly encountered edge operators. Chapter 3

discusses relative grading, one solution to these problems, while Chapter 4 covers absolute grading, a second approach.

Chapter 5 provides examples of the use of those techniques developed in Chapters 3 and 4, and illustrates some of the many applications of relative and absolute grading.

CHAPTER 2

FUNDAMENTALS OF EDGE DETECTION

The human observer is the ultimate standard of comparison for many image processing systems. A human observer seems to rely more upon edges and boundaries rather than individual pixel amplitudes for visual recognition and interpretation [1]. For this reason, it is important to be able to extract, with as much accuracy as possible, the edge and boundary information from a scene in the same manner as humans if image understanding systems are to produce usable and meaningful results. Consequently, the definition, detection and assimilation of edges and boundaries is of ever increasing interest to image processors, and the subjects of many theoretical investigations [2,3,4,5,6,7].

Edges can be defined as amplitude discontinuities in some image attribute, usually brightness. In some cases these variations are due to textural variations within an image region, and are termed texture edges. Throughout this thesis, the term edge will refer to object, or nontexture edges. Boundaries are generally assumed to be the borders of certain particular objects of interest, also exhibiting amplitude discontinuities, and, as such, are edges as well. The problem of edge and boundary detection is, in essence, that of detecting these discontinuities.

While this might, upon first glance, appear to be a rather straightforward problem, a more detailed study of the literature reveals the many complexities involved and exposes the great dilemmas which face those whose task it is to deal with any of the various aspects of edge detection.

Ideal edges found in images of homogenous objects are generally of the types shown in Figure 1, depending upon the contents of the picture being analyzed. By far the most common is the step edge (Figure 1a), while roof edges (Figure 1b) make up the majority of those edges remaining.

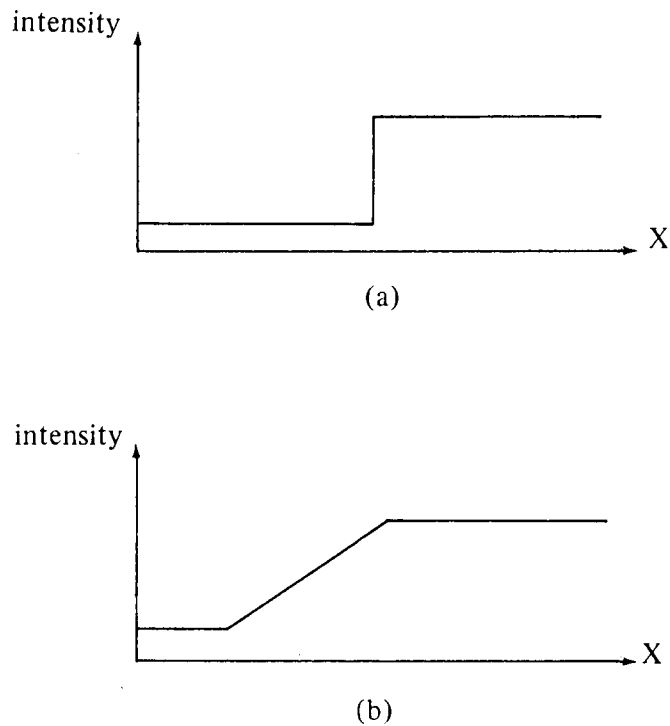


Figure 1. Ideal edge crosssections. (b) characterizes solid objects with surface intersections.

General limitations of the digitization process as well as physical factors tend to degrade these ideal edges [2]. The causes include:

1. photon noise (quantum effects)
2. blurring or defocusing

3. irregularities of the surface structure of the objects
(texture)

4. sensor quantizing (resolution)

The degree to which each of these factors contribute to the overall deterioration of edges varies widely with the application, but they are almost universally encountered to some noticeable extent.

The net effect on the ideal step edge is to broaden the transition region and reduce the intensity discontinuity, through a convolution process, as shown in Figure 2 [2].

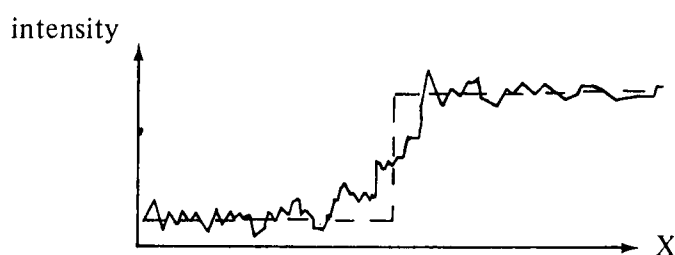


Figure 2. A corrupted edge superimposed on an ideal step edge.

The edge detection process is greatly burdened by these corrupting factors, often to the point where visually apparent edges cannot be discriminated within a small image subarea or, conversely, a transition which seems to be an edge within the subarea might actually be due to the texture of a somewhat larger region.

The general solution to edge and boundary detection, and the one which is recognized as the best approach, involves the utilization of a relatively simple (mandated from a computational cost standpoint) local edge detector, followed by some form of thinning and linking [7].

To achieve this purpose, many fast numerical techniques for intensity edge extraction have been devised. For the most part, each of these operators relies on the same general principles, as dictated by the nature of the problem, while incorporating some additional innovations which depend greatly upon the specific application for which they are devised. Unfortunately, this has resulted in an overabundance of potential edge detectors and causes many complications in the process of selecting a particular operator for a certain new application.

Of the many operators, a few have shown themselves to be acceptable over a fairly wide variety of imagery, and are the most commonly used by image processors today [6,7]. These operators are: gradient [3], Robert's gradient [4], Sobel's [4], range [6], Frei-Chen [7] and Robert's square root gradient [3]. Each approach is slightly different, being custom tailored to avoid certain pitfalls which are encountered when analyzing edge information, while attempting to utilize as many of the general principles of edge detection as possible, in order to insure maximal edge extraction and minimum "noise."

An examination of these six operators gives a general idea of the various types of solution schemes employed to locate principle edges as well as providing some insight into the scope of the often encountered but seldom understood problem of boundary and edge detection.

Gradient Operators

Locating abrupt intensity changes in a two-dimensional image is analogous to finding sharp transitions in a one-dimensional waveform. The most commonly used method of locating such transitions is by

observing the value of the waveform's derivative and noting where it becomes large. This approach can be used for images if the differentiation is extended into two dimensions. The magnitude of the two-dimensional derivative reflects the steepness of any change in gray level and can be used to identify an edge.

Known as the gradient, this two-dimensional differentiator can be formulated as:

$$\vec{G} [f(x,y)] = \begin{bmatrix} \partial f / \partial x \\ \partial f / \partial y \end{bmatrix} \quad (1)$$

It is important to note that $\vec{G}$ is a vector which points in the direction of the maximum rate of increase of the function $f(x,y)$.

For a digital image, the derivatives in equation (1) can be approximated by differences computed in the appropriate direction. This formulation gives the gradient magnitude as

$$G[f(x,y)] \cong \sqrt{[(f(x,y) - f(x+1,y))^2 + (f(x,y) - f(x,y+1))^2]} \quad (2)$$

which is often simplified, in the interest of computation expediency to

$$G[f(x,y)] \cong |f(x,y) - f(x+1,y)| + |f(x,y) - f(x,y+1)| \quad (3)$$

The directional relationship between pixels shown in Figure 3a. The inclusion of pixels $(x+1,y)$ and $(x,y+1)$ in the computation of the gradient at (x,y) means it is not possible to take the gradient for points which lie in the N^{th} row or N^{th} column of an $N \times N$ image.

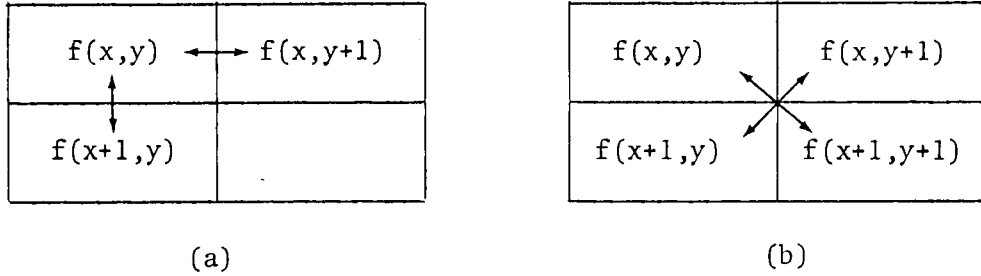


Figure 3. Directional differences. (a) Gradient, (b) Robert's gradient.

Other forms of the gradient were proposed by Roberts [4] in an attempt to utilize all the directional information present in the 2x2 neighborhood of (x,y) . One type, known as Robert's square root gradient [4] uses cross-differences, as shown in Figure 3b, in the form

$$G[f(x,y)] = \sqrt{(f(x,y) - f(x+1,y+1))^2 + (f(x+1,y) - f(x,y+1))^2} \quad (4)$$

As with the gradient, equation (4) can be simplified into a form, known as Robert's magnitude gradient, [4] using magnitudes instead of powers:

$$G[f(x,y)] = |f(x,y) - f(x+1,y+1)| + |f(x+1,y) - f(x,y+1)| \quad (5)$$

Basically, all of these various forms determine the directional derivatives in the x and y directions at point $f(x,y)$. The magnitude of the gradient reflects the degree of abruptness of any changes in gray level within the four pixel neighborhood of $f(x,y)$, and is a reasonable measure of the presence of an edge in that area. They are not, however, without their drawbacks. There is some question as to the minimum value

of the gradient which indicates an edge and how much, if any this level changes throughout the image. Additionally, there are computational deficiencies which can result in erroneous values for the gradient in certain cases. Consider the case for which $f(x,y) = f(x+1,y+1)$ and $f(x+1,y) = f(x,y+1)$ and $f(x,y) \neq f(x+1,y)$. All of the gradient operators yield results of zero despite the fact that 2 narrow, parallel edges pass through the neighborhood. (A single edge could not produce such a pixel pattern.)

To overcome some of these drawbacks, an operator which is gradient-like in structure has been developed [6]. Known as the maximum difference or range operator, this edge detector measures the difference between the maximum and minimum gray levels in the neighborhood.

$$g(f(x,y)) = \text{Max} (f(x,y), f(x+1,y), f(x,y+1), f(x+1,y+1)) \\ - \text{Min} (f(x,y), f(x+1,y), f(x,y+1), f(x+1,y+1)) \quad (6)$$

For the particular case mentioned above, the range operator would yield a nonzero value indicating the presence of some discontinuity.

Additionally, the range operator has the ability to produce more continuous edges. This is due to its increased sensitivity to closely packed parallel edges which often confuse gradient operators [6].

Three-by-Three Operators

Despite minor variations, all the operators mentioned in the previous section base their edge decisions upon the values of those pixels lying in a two by two neighborhood of $f(x,y)$. It is not unreasonable to expect that more precise measurements can be made when additional information

about the image subarea which contains $f(x,y)$ is available. This increase in information can be brought about by increasing the size of the neighborhood, and many operators rely on just that approach.

Logically, the next step up in neighborhood size would be to a three-by-three. In addition to accessing a larger segment of the scene about any given point, the three-by-three offers more degrees of freedom in the manner in which the additional data can be processed. This has led to a wide variety of operators, differing greatly in approach, but utilizing the greater flexibility of a three-by-three subarea. A majority of these operators fall into one of four basic classes:

1. Mask convolution
2. Template matching
3. Basis vector projection
4. Multi-step computation

Mask convolution involves the convolution of the scene under study with a three-by-three array, or mask, which is constructed to carry out specific mathematical operations. Such an operator is the discrete Laplacian, which calculates the divergence of the gradient of a scalar pixel field [3]. In essence, the Laplacian operators (there are several) use the second-order directional derivatives at a point as the measure of edge presence. The Laplacian of a two-dimensional function, $f(x,y)$ is

$$\nabla^2 f(x,y) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \quad (7)$$

Replacing the derivatives with differences yields the discrete version:

$$\nabla^2 f(x,y) = \left(\frac{\partial f}{\partial x} \right)_{x+1,y} - \left(\frac{\partial f}{\partial x} \right)_{x,y} + \left(\frac{\partial f}{\partial y} \right)_{x,y+1} - \left(\frac{\partial f}{\partial y} \right)_{x,y} \quad (8)$$

where the partial derivatives can be approximated in any of the ways described for the gradient-type operators.

The variety of estimation techniques for the partial derivatives in equation 8 results in many possible masks for the Laplacian operator, one of which is shown in Figure 4.

(a)

1/2	0	1/2
0	-2	0
1/2	0	1/2

(b)

1	0	1
0	-4	0
1	0	1

Figure 4. True Laplacian mask. Multiplication of true Laplacian mask (a) by 2 yields scaled integer mask (b).

The Laplacian is, in essence an omnidirectional derivative and is useful as an edge detector where no preferential edge direction exists. Use of the Laplacian operators is often known as unsharp masking.

Mask convolution can also be used to carry out defocusing operations using lowpass masks such as those shown in Figure 5.

$W_0 = \frac{1}{9}$

1	1	1
1	1	1
1	1	1

(a)

$M_0 = \frac{1}{16}$

1	2	1
2	4	2
1	2	1

(b)

Figure 5. Lowpass masks. (a) Neighborhood averaging
(b) Improved mask with better sidelobe structure.

Template matching is very closely related to mask convolution in that it too involves the convolution of the image with a three-by-three (or larger) array. Instead of involving some mathematical process whose value is indicative of a particular attribute, template matching produces a numerical result which is a measure of the degree-of-resemblance between the template and an image subarea. The most common use of template matching in the area of edge detection involves the use of directional masks. Shown in Figure 6 [8], directional masks are specially designed to produce large values when an edge with a particular orientation is encountered. The complete set of eight masks is capable of detecting straight edges passing through the center of the three-by-three pixel group while also determining edge direction. Some directional masks such as the three and five level simple masks of Figure 6, cover all eight possible directions with only four masks, making use of the sign of the result, as well as the magnitude, to obtain edge orientation. The term "level" is often associated with directional templates to specify the number of different integer values used within the mask. Thus the Prewitt [8] and Kirsch [8] masks are three level masks since the Prewitt set uses the integers -2, -1 and 1, and the Kirsch set uses -3, 0 and 5.

There are several considerations fundamental to the development of templates. While pixel value is not important, it is critical that the sum of all mask elements be zero, in order to insure that no edge is indicated when the image subarea is of uniform gray level. The center element is often forced to zero to suppress jitter on the line where an edge might occur. Additionally, normalization is often called for after

	Prewitt Masks	Kirsch Masks	3-Level Simple	5-Level Simple
NORTH	$\begin{bmatrix} 1 & 1 & 1 \\ 1 & -2 & 1 \\ -1 & -1 & -1 \end{bmatrix}$	$\begin{bmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$	$\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$
NORTHWEST	$\begin{bmatrix} 1 & 1 & 1 \\ 1 & -2 & -1 \\ 1 & -1 & -1 \end{bmatrix}$	$\begin{bmatrix} 5 & 5 & -3 \\ 5 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -1 \end{bmatrix}$	$\begin{bmatrix} 2 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -2 \end{bmatrix}$
WEST	$\begin{bmatrix} 1 & 1 & -1 \\ 1 & -2 & -1 \\ 1 & 1 & -1 \end{bmatrix}$	$\begin{bmatrix} 5 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & -3 & -3 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$
SOUTHWEST	$\begin{bmatrix} 1 & -1 & -1 \\ 1 & -2 & -1 \\ 1 & 1 & 1 \end{bmatrix}$	$\begin{bmatrix} -3 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & 5 & -3 \end{bmatrix}$	$\begin{bmatrix} 0 & -1 & -1 \\ 1 & 0 & -1 \\ 1 & 1 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & -1 & -2 \\ 1 & 0 & -1 \\ 2 & 1 & 0 \end{bmatrix}$
SOUTH	$\begin{bmatrix} -1 & -1 & -1 \\ 1 & -2 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	$\begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & -3 \\ 5 & 5 & 5 \end{bmatrix}$	$\begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$	$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$
SOUTHEAST	$\begin{bmatrix} -1 & -1 & 1 \\ -1 & -2 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	$\begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & 5 \\ -3 & 5 & 5 \end{bmatrix}$	$\begin{bmatrix} -1 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$	$\begin{bmatrix} -2 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 2 \end{bmatrix}$
EAST	$\begin{bmatrix} -1 & 1 & 1 \\ -1 & 2 & 1 \\ -1 & 1 & 1 \end{bmatrix}$	$\begin{bmatrix} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & 5 \end{bmatrix}$	$\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$
NORTHEAST	$\begin{bmatrix} 1 & 1 & 1 \\ -1 & -2 & 1 \\ -1 & -1 & 1 \end{bmatrix}$	$\begin{bmatrix} -3 & 5 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & -3 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{bmatrix}$

Figure 6. Typical directional masks.

the convolution to prevent overflow or underflow in the encoding of the result.

It should be pointed out that the templates can be constructed to test for the presence of lines and points as shown in Figure 7.

A relatively new approach to the three-by-three system of edge detection involves the development of an orthogonal basis vector set which spans the implied nine dimensional space. A set of nine three-by-three masks which satisfy the spanning criteria in a manner meaningful to edge detection has been developed by Frei-Chen [7], and is shown in Figure 8.

To use this basis vector set it is necessary to compute the sum of the projections of the image on the first four bases of Figure 8. This value is retained as the indication of edge presence.

Many other basis vector sets exist which span nine dimensional space, since all that is required of the set is that it consist of nine vectors and that any three-by-three pixel pattern can be represented by a linear combination of the members of the set. Consideration should be given to the structure of the vectors derived for they must be similar to the patterns of interest in order to be useful.

One technique for generating an orthonormal or orthogonal basis vector set which spans r -dimensional space is known as Gram-Schmidt orthonormalization [9]. Gram-Schmidt orthonormalization is a mathematical process for deriving a spanning set of orthonormal vectors from a given set of linearly independent data waveforms.

-1	-1	-1	-1	-1	2	-1	2	-1	2	1	-1
2	2	2	-1	-2	-1	-1	2	-1	-1	2	-1
-1	-1	-1	2	-1	-1	-1	2	-1	-1	-1	2

(a)

-4	5	5	-1	1	-1	4	4	-5	-3	-3	6
5	-4	-4	1	-1	1	-5	-5	4	-3	6	-3
5	-4	-4	-1	1	-1	4	4	-5	-3	6	-3

(b)

-1	-1	-1
-1	8	-1
-1	-1	-1

(c)

Figure 7. Examples of specially designed templates. (a) straight lines, (b) various compound edges and curves, and (c) points.

1	$\sqrt{2}$	1
0	0	0
-1	$-\sqrt{2}$	-1

1	0	-1
$\sqrt{2}$	0	$\sqrt{2}$
1	0	-1

0	-1	$\sqrt{2}$
1	0	-1
$-\sqrt{2}$	1	0

$\sqrt{2}$	-1	0
-1	0	1
0	1	$-\sqrt{2}$

Basis of edge subspace

0	1	0
-1	0	-1
0	1	0

-1	0	1
0	0	0
1	0	-1

1	-2	1
-2	4	-2
1	-2	1

-2	1	-2
1	4	1
-2	1	-2

Basis of line subspace

1	1	1
1	1	1
1	1	1

Average subspace

Figure 8. Frei-Chen orthogonal templates.

The development of a meaningful basis vector set is nontrivial, and the complexity increases drastically with increased dimensionality of the space.

This form of edge detector, like that of directional masks is more computationally expensive because of the number of mathematical operations necessary.

The fourth major class of three-by-three operators combines elements of the other three into a computational form which processes their raw numerical output according to some predetermined mathematical system of equations. The result of this analysis is taken as the indicator of edge presence within the neighborhood.

The Sobel operator [10], is perhaps the most widely employed operator of this type. The edge magnitude is given by:

$$E(f(x,y)) = \sqrt{X^2 + Y^2}$$

Where X and Y are the results of the convolutions of the two three-by-three masks of Figure 9 with the neighborhood of (x,y).

1	2	1
0	0	0
-1	-2	-1

(a)

1	0	-1
2	0	-2
1	0	-1

(b)

Figure 9. The two gradient masks used in the Sobel operator.
(a) Mask for X, (b) Mask for Y.

Obtaining Directional Information

Edge direction is as important as edge magnitude, but few simple operators deal with, or produce edge orientation information. Directional masks offer the capability of determining the approximate angle an edge takes as it passes through the image subarea, but is encumbered by the great computational expense of convolving eight masks and the accumulating of intermediate results, while not yielding very good resolution in terms of specific edge angle. The ability to ascertain the angle from a single operator or a few operations is a definite requirement of any sophisticated edge detection system.

An approximation of edge angle is often made from the two directional components of a gradient-type operator such as the Sobel operator. The same values X and Y which give the magnitude of the edge (see Figure 9) for the Sobel operator can be used to infer orientation from:

$$\theta = \tan^{-1} \left(\frac{Y}{X} \right)$$

In general, for any gradient system, whose directional derivative masks are W_1 and W_2 , in the x and y direction respectively, and an image neighborhood B , the angle of the contained edge is

$$\theta = \tan^{-1} \left[\frac{B \cdot W_2}{B \cdot W_1} \right]$$

where $(\cdot)$ indicates the dot product, or convolution operation. This result should be evaluated in the light of the edge magnitude obtained for the same subarea B . If no edge is indicated, this angular measurement is meaningless.

A more sophisticated method of determining edge angle has been developed by Davies [6]. This approach analyzes the arrangement of gray levels in the 2 x 2 neighborhood of (x,y) and estimates the angle from cross differences.

The first step is to rotate the neighborhood until the largest pixel value is in the upper left-hand position. The amount of rotation necessary to affect this is a coarse estimation of edge angle. To it is added a fine angle determined from

$$\alpha = \frac{\pi}{2} \left| \frac{G_1 G_4 - G_2 G_3}{(G_1 - G_4)(G_2 + G_3)} \right|$$

where the pixels G_1 , G_2 , G_3 and G_4 are as shown in Figure 10.

G_1	G_2
G_3	G_4

Figure 10. Arrangement of pixels for Davies operator. G_1 must be the maximum gray level.

Additionally, statements about the edge which passes through the neighborhood can be made from this approach. If $G_1 = G_2 = G_3 = G_4$, no edge exists. If G_4 is not the minimum gray level, then the pattern could not have been produced by a simple linear edge. Special cases occur when two or three of the four pixels have the same gray level. For a more detailed discussion see [6] or the Appendix.

Several excellent discussions of those operators mentioned here, as well as others of all kinds can be found in a paper by Davis [2], and texts by Pratt [10], Duda and Hart [5], Gonzalez and Wintz [3], and Hall [3].

Grading

With the wealth of edge detection methods available, the user is presented with the dilemma of evaluating and selecting a "best" operator for a certain application.

Many make this choice arbitrarily, without thorough investigation or forethought, simply because of the sheer size of the problem.

Instead of endeavoring to simplify this situation, many investigators choose to seek newer and more-nearly ideal operators, or develop more refined, multi-step algorithms for principle edge extraction. Consequently, the literature is a spectrum of edge operators that are optimized for a particular application and is relatively devoid of discussion on operator selection procedures [2,3,4,5,6,7].

One attempt [12] to quantitatively evaluate edge operators, as well as compare their results against human psychovisual interpretation, addresses only their performance in the presence of noise. The basic procedure was to develop a set of test images for which both the noise and edge signal could be characterized numerically. By adjusting the noise variance and edge contrast within the set, the range of acceptable operation for each detector was determined.

Detailed statistical interpretation of the results of this test was necessary to finally rank operators. The end product, while strictly

formulated and well defined, is tedious to carry out, and really does not sufficiently enable one to decide which operator is best suited for a given environment, unless the prime consideration is noise immunity.

Pratt [8] developed a scheme for numerical evaluation of operators by ascertaining their ability to accurately locate an edge. His figure of merit is defined by:

$$R = \frac{1}{I_n} \sum_{i=1}^{I_a} \frac{1}{1+\alpha d^2}$$

where I_a is the number of actual edge map points I_n is the maximum of I_a and I_i , the number of ideal edge points, α is a scaling factor and d is the separation distance between an actual edge point and a line of ideal edge points. If the edge detected aligns with the ideal edge, R should equal unity. The scaling factor α is used to create an adjustable penalty for offset edges, allowing smeared edges to be more highly rated than a thin edge which is dislocated.

Pratt's measure is most meaningful for noisy scenes and offers no means of computing detection reliability or consistency.

In developing any performance evaluator for edge detection techniques, it is necessary to decide which information is needed from the operator and which is ancillary. As has been shown [8], the prime data in terms of processing usefulness is edge magnitude and spatial orientation. To be considered "best" an operator must be able to derive this information while introducing as few errors as possible. It must also be able to

detect all edges of interest, if possible. Errors in edge magnitude are of three major types:

1. Misclassifying noise or texture points as edges.
2. Mislocating an edge.
3. Failure to detect valid edge points.

The first and third types actually constitute the error portions of the decision confusion matrix for edge detection as shown in Figure 11.

	True	False
Positive	Correctly located edge point	Noise or texture point erroneously classified as edge (Error)
Negative	Correctly identified non-edge point	Edge point which is not recognized as such (Error)

Figure 11. Confusion matrix for edge detection process.

These two types of errors can be treated as noise and undetected signal.

Errors in measuring edge angle are the result of conceptual flaws in the design logic behind the measurement scheme. This error is difficult to detect and almost impossible to correct.

From the output of a single operator, it is not feasible to identify and tabulate the errors by any simple automated technique despite the ease with which a human observer can locate the errors from a quick comparison with the original scene. If the output of more than one operator is available, however, a quick method of identifying at least some of these errors would be to compare the results to each other.

It is not unreasonable to expect that those errors which are caused by quirks characteristic to individual operators could be removed in much the same manner that spatial averaging eliminates random noise from sets of received signals. That is, points which are found to be edges in only one result from the set of results could be considered "noise" or errors and removed. Those points which are found in a majority of the result edge maps could be classified as missed edge points in those output maps which do not contain them.

This approach could easily be used to evaluate edge detection schemes by allowing the erroneous decisions to be counted, and a grade computed from the percentage of correct classifications. Because this involves relative comparisons in the evaluating process, it can be termed relative grading.

Conversely, a different type of grading scheme much like that encountered in many scoring systems may be used. By comparing the operators against a "correct" edge map, a numerical grade can be computed. This can be extended to checking against a target edge of special importance. Since the comparisons here involve a predetermined key, this approach can be called absolute grading.

Both the methods offer straightforward, uncomplicated, quantitative evaluation of edge detectors. Neither has previously been systematically explored but both exhibit great potential for reducing the uncertainty in selecting an appropriate edge operator.

Chapter 3 deals with relative grading, and contains a development of a basic relative grading system, while Chapter 4 addresses absolute grading, and the associated area of iterative operator refinement.

CHAPTER 3

RELATIVE GRADING

Thresholding

The decision that a point is, or is not an edge point is a binary one. The edge operators in the previous chapter do not actually make that decision. Instead, they provide a continuous measurement of the presence of an edge. Clearly then, some external process is involved in converting this range of numerical results into a yes-or-no value. This process is known as thresholding.

Thresholding involves comparing a function's value to a numerical quantity, termed the threshold, and making a binary decision based upon the results of an inequality. For example,

$$g(x,y) = \begin{cases} J[f(x,y)] & \text{if } G[f(x,y)] \geq T \\ H[f(x,y)] & \text{otherwise} \end{cases} \quad (1)$$

where $f(x,y)$ is the original image, $G[f(x,y)]$, $J[f(x,y)]$ and $H[f(x,y)]$ are functions of $f(x,y)$, and T is the threshold level. In some cases $J[f(x,y)]$ and $H[f(x,y)]$ can be constant values instead of functions.

In terms of edge detection $G[f(x,y)]$ is the edge operator, $H[f(x,y)]$ is often zero, $J[f(x,y)]$ is the maximum system gray level, and T is the level selected as that which minimally indicates an edge. This reduces equation 1 to:

$$g(x,y) = \begin{cases} 2^n - 1 & \text{if } G[f(x,y)] \geq T \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

with n equal to the number of bits used to code the gray level.

The simplest means of thresholding is to decide at what level or magnitude the operator indicates an edge. This threshold can be determined from a mathematical analysis of the operator, from past experience, or by examining the histogram of the image. Once the threshold is selected, a simple comparison is made, point-by-point, and an edge is said to exist wherever the magnitude result of an operator exceeds the threshold. A common choice for this threshold is the average value of the gradient [8].

A second type of thresholding involves estimating the edge content of the image. An educated guess is made of the percentage of edge points in the scene. Then, that same percentage of points is taken from the highest magnitude points in the operator's output. For example, it is estimated that 10 percent of the N^2 points in a certain $N \times N$ picture are edge points. The points which are selected as edge points are the 10 percent which have the largest magnitude, or the top $N^2/10$ points. The actual percentage is usually arrived at from a trial-and-error test, changing the value until the result agrees closely with human interpretation.

Rosenfeld [13] has proposed a nonlinear thresholding technique which identifies edge points by comparing them to their neighbors. Known as dominant-neighbor suppression, this procedure scans the edge map with a small window, and zeroes a point unless its value is larger than all

others in the window. A variation permits suppression only if a neighbor dominates by a specified amount. Conventional thresholding follows application of dominant-neighbor suppression, which can be viewed as a thinning technique.

G. Robinson [8] describes a method involving a locally adaptive threshold (LAT), obtained by comparing the result of a directional mask operator to a blurred version of the original threshold.

$$LAT = \frac{\text{Maximum value produced by set of directional masks}}{\text{Output of low pass filter at } (x,y)} \quad (3)$$

This thresholding scheme relies on a concept which is not considered. Most edge operators produce a numerical result indicative of the change in gray level in a particular image neighborhood. This result is not presented in any terms other than measured change. A more reasonable representation would be relative or percentage change.

It is not difficult to see that this value more closely depicts the edge quality at a point. A general thresholding technique would consider a transition from $f(x,y) = 100$ to $f(x+1,y) = 120$ more likely to be an edge than the transition from $f(x,y) = 10$ to $f(x+1,y) = 25$. A percentage comparison would do just the opposite, indicating those transitions which are the most drastic.

This process can be developed into a pre-thresholding enhancement technique which converts the magnitude result of an operator $H(x,y)$, into a relative change value $R(x,y)$.

$$R(x,y) = \frac{H(x,y)}{\text{output of low pass mask at } (x,y)} \quad (4)$$

The image R which results would show the major transitions as large values and lesser transitions, such as those associated with texture, as reduced values.

Relative Grading

The end result of any thresholding is a binary image, with each point tagged as edge or background. At this point the edge detection can be considered complete and all edges detected. This is where the task of evaluating the operator, and determining the accuracy of the detection process begins. As discussed in Chapter 2, the errors which must be found and counted are difficult to separate from the correctly classified points. Employing a relative grading scheme to the binary edge maps produced by several operators should, however, allow a reasonable tabulation of the errors and permit the computation of a grade.

As briefly mentioned in Chapter 2, relative grading involves the comparison of several operators' interpretation of the same scene, and uses mutual agreement to decide the edge status of a point.

The general assumption is that each operator, due to its own intricate innerworkings, will produce misclassifications on some regular basis because of oversights on the part of its designer, or unavoidable consequences of its operation. These mistakes are, to a certain extent, different from operator to operator. Additionally, some operators are known to have superior detection capabilities for specific types of edges, often because they are designed to do so.

By combining the results of several operators, the idiosyncratic errors can be reduced, due to their uniqueness, while various advantageous aspects can be combined to recognize those points which should

be edges, even though no one operator detected them all. Once the errors have been located, they can be compensated for, producing a "corrected" or improved edge map for further study. This result, as well as a numerical ranking of the operators involved combined with its simplicity and speed makes relative grading a potentially important tool for image processing.

The composition of the set of operators has a profound effect upon the grades which result. Suppose the set consists of four rather "poor" operators, and one "good" one. The comparison process will be skewed toward the poor operators due to the fact that they greatly overwhelm the good one. As a result, the good operator will receive a deflated grade, while the poor ones do well at its expense. It is important then, to be sure that some form of compensation be made so that no advantage is given to any operators.

This can be done either of two ways. By being certain that the operators in the set are roughly of the same precision, the tester can be more confident that no operator, or group of operators gets an upper hand. The estimation of operator quality need not be exceedingly accurate, and can be made by a human observer.

A second scheme for equalizing the grading process is to weight the operators by including them more than once in the set. The situation mentioned earlier could be compensated for by expanding the test set to eight operators — the four "poor" ones, and the "good" one repeated three extra times. As with the other equalization scheme, this weighting can be adequately determined by a human observer.

Additional considerations beyond test set composition can alter appreciably the grade produced. One is the manner in which the joint edge presence is made. For a particular point, some operators may indicate an edge, while others disagree. The number of no-edge decisions which determine a joint no-edge decision is a variable of the grading process, and has an important effect on the grades which result. If this level is too low or too high the operators which are actually correct will be unfairly penalized. Estimation of this level can be made from an examination of the number of isolated edge points in each of the members of the test set. If this number is great, the threshold level can be made low, while a low number of isolated points indicates a larger threshold is necessary.

The manner in which the comparison is made can also cause differences in the grade produced. From an implementation standpoint, the simplest technique is a straightforward comparison in which the grade for an operator is determined with its results included in the set of compared operators. For this case, a simple pass through the output of all operators, with the joint decision being made from the entire set results in the "corrected" image, and a grade is made by checking each operator against this scene. Such an approach can be considered a polling, because of its resemblance to the voting process of making a decision.

Voting mechanisms are well defined and understood within the realm of pattern recognition, where they are utilized to resolve decision-making dilemmas caused by nonlinearly separable training patterns [14]. Such dichotomization problems are handled by layered committee machines

which rely upon a majority or committee decision to obtain proper sample classification.

In general, a committee machine has P members, each being some form of threshold logic unit (TLU), where P is odd. These TLUs comprise the first layer of a two layer machine. The second layer is another TLU, known as the vote taking TLU, which polls the committee members and produces the majority response. This vote taking TLU can attach various weights to the members of the committee, and operate at vote taking thresholds other than a simple majority.

These concepts can readily be extended to edge operator evaluation. The first layer in the machine is made up of the operator/thresholding pairs, with the edge presence decision corresponding to a selection of either of two classes, edge or nonedge. Polling the operator's output, and the process of making a majority decision is, in essence, constructing and utilizing a vote taking TLU, complete with input weights and variable decision threshold.

A problem with this scheme is the biasing of each operator's grade by its inclusion in the decision making process. A more equitable approach would be to evaluate the output of an operator by comparing it to the joint decision of all other operators in the test set. This is much more tedious to implement because of the number of possible combinations of grading comparisons. A coding scheme is an excellent method of overcoming this drawback, and is described in the Appendix. This can be termed exclusion voting.

No matter what the grading approach, it is necessary to have enough operators in the test set to make the outcome meaningful. Including only

two operators reduces grading to an exclusive-or tabulation, with the corrected image being the logical-and product of the two. On the other hand, computational difficulties can be encountered if too many operators are tested at once, especially if exclusion voting is employed.

In most cases, the number of edge points in an image is far less than the number of nonedge, or background points. This can cause undesirable weighting and inflates the grade so that two operators might be less similar than the difference in their percentage grade indicates. A solution to this problem is to compute two partial grades, one which indicates what percentage of edge points were correctly classified, and the other indicating the percentage of background points which were identified accurately. If one operator were to correctly locate 100 more edge points than another, the total grade might not indicate this as any more than a fractional percentage difference, while the partial grading system would make the difference obvious. The total grade could be obtained, if necessary, by combining the two partial grades and recalculating the percent score.

The threshold method used to convert an operator's output into a binary image can have an effect on the grade as well. Simple thresholding does not introduce appreciable differences as long as it is consistently and intelligently applied. Percent thresholding, however, can cause noticeable errors due to the nature of gray level distribution within images.

An $N \times N$ image consists of N^2 points of M different gray levels. The percent thresholding technique operates upon the assumption that P percent of the N^2 points, or $PN^2/100.0$ points are edge points. The actual

threshold is selected by starting at gray level M , and moving downward through the histogram until the number of points above a gray level T is greater than or equal to $PN^2/100.0$. This means that, although it is assumed that P percent of the points are edge points, the binary image will most likely have more than $PN^2/100.0$ points in it, and each operator will produce a different number of points in its binary output. This is true of simple thresholding as well, but the number of points in the binary image is not directly related to the threshold level. In percentage thresholding, it is important that the number of output points is as close to the percentage P as possible to maintain the integrity of the thresholding process, otherwise additional points will be included and the grade substantially altered. This is analogous to each operator being thresholded at a different percent, an obviously undesirable situation.

A relative grading system, then, consists of several steps, and contains several options as to exact details of the process. First, a group of operators is chosen, and is assembled into an equalized set. These operators are applied to a particular image and their output thresholded by some legitimate thresholding algorithm. Next, a voting scheme, either polling, or exclusion voting is selected, and the binary operator outputs are compiled in a joint-decision process, and a numerical grade is produced, with one score for correctness of edge classification and another for background identification accuracy. These results can then be compared to decide which operator performed most satisfactorily on the original scene. From knowledge of the location of the errors in the edge detection process, a correlated image, with fewer misclassified points can be constructed.

CHAPTER 4

ABSOLUTE GRADING

Not surprisingly, the speed and simplicity of relative grading are gained at some expense, and, as is true with most simplified systems, these expenses are in the form of process drawbacks which have their roots in its underlying assumptions.

Often, a certain edge is of paramount importance in an image, and it is this edge which must be reliably detected. In fact, there are frequent situations in which no significance at all is placed on other edges, and whether they are detected or not is of no interest, as long as the prime edge is found.

A relative grading evaluation of operators in this situation may not be relevant, especially if none of the operators locate the principle edge.

In cases such as this it is necessary to determine which operator does the best job detecting a specific edge, regardless of its ability to function over the entire scene. This is where the second type of grading, absolute grading, is more useful.

As mentioned in Chapter 2, absolute grading compares a operators output to a "correct" image, or key. The key can be considered to be the output of an ideal operator, and the degree to which an actual operator can equal the ideal operator's performance is what is reflected in the grade. This approach is utilized widely and can be recognized as the form by which a teacher grades student's tests, a manufactured

part is checked-out or a device is repaired. It is important to note that the ideal operator need not exist, as long as its result is available, or can be constructed from some source.

To a great extent, absolute grading can be thought of as a measure of the degree of similarity between the output of a real operator and an ideal operator. A common method of computing this similarity is the basic correlator.

In this particular application the basic correlator method consists of forming a correlation measure, R , between the operator output and the key, with R defined as [11];

$$R(u,v) = \frac{\sum_{i=1}^J \sum_{j=1}^K K(i,j)O(u+i-1, v+j-1)}{\left[\sum_{j=1}^J \sum_{j=1}^K K^2(i,j) \right]^{1/2} \left[\sum_{i=1}^J \sum_{j=1}^K O^2(u+i-1, v+j-1) \right]^{1/2}} \quad (1)$$

where K is the key, a $J \times K$ array, and O is the operator output, a $M \times N$ array. The Cauchy-Schwartz inequality states that $-1 \leq R(u,v) \leq 1$, and for positive-gray level pictures, the range becomes $0 \leq R(u,v) \leq 1$.

Clearly, a great amount of computation must be performed, since $R(u,v)$ must be calculated for every (u,v) such that $1 \leq U \leq M-J+1$ and $1 \leq V \leq N-K+1$. Several simplifications can be made, however, to expedite the determination of R .

First, the value of the term

$$K_m = \left[\sum_{i=1}^J \sum_{j=1}^K K^2(i,j) \right]^{1/2} \quad (2)$$

is the same throughout the scene for any given key. This reduces the number of computations necessary.

Additionally, it should be remembered that the imagery involved is binary, and can be thought of as arrays of ones and zeros, eliminating the need for multiplication. In fact, the calculation of R can be reduced to a series of additions:

$$R(u,v) = \frac{N_C(u,v)}{K_m[N_O(u,v)]^{1/2}} \quad (3)$$

where $N_C(u,v)$ is the number of points common to both the key and the $J \times K$ region about (u,v) in the image O , $N_O(u,v)$ is the number of nonzero points within the same region and K_m is as given in equation 2. It is still necessary to compute $R(u,v)$ for all (u,v) within the bounds associated with equation 1.

The simplest case is that in which the key is the same size as the operator output. This requires $J=M$ and $K=N$, fixing (u,v) to $(1,1)$ and giving

$$R = \frac{N_C}{\sqrt{N_k N_O}} \quad 0 \leq R \leq 1 \quad (4)$$

where N_k is the number of nonzero points in the key and N_C , N_O as in equation 3.

From an absolute grading standpoint, the simplification involved in equation 4 implies that the nonzero portion of the key be located in the same place in the key image as the target edge is in the original scene. This introduces a registration aspect into the grade. Any operator which

functions in a manner that would produce a displacement in the detected edge position would run the risk of altering its own grade, most likely with detrimental effects.

This is not altogether undesirable, for in many cases the position of the edge is of greater importance than its magnitude. Such a penalty is, however, uncontrollable, and is not a function of the displacement produced.

By making the key image slightly smaller, a degree of variability can be introduced, allowing the actual displacement to be determined by calculating the distance from the center of the ideal edge to the point with the highest correlation value. This displacement can be used to adjust the grade to compensate for the error. For example, if the maximum correlation R_{max} occurs at a displacement of d , the weighted grade can be found from

$$R = \frac{R_{max}}{d^{\alpha}} \quad (5)$$

with α set to introduce the desired amount of grade reduction.

The decision to go through the extra steps to allow for displacement compensation should be based upon the importance of edge position, the application and the operators involved.

The correlation approach to absolute grading then can be broken down into several counting operations. The information necessary is N_o , the number of nonzero points in the thresholded version of the operator output, N_k , the number of nonzero points in the key image and N_c , the number of common points. The grade can then be computed from equation 4.

It is interesting to note that the expression

$$R = \sqrt{\frac{N_k}{N_o}} \left(\frac{N_c}{N_k} \right) \quad (6)$$

is equivalent to equation 4. The first term is the exponentially weighted ratio of the number of points in the key to the number of points in the operator output. This cover-ratio takes into account the degree to which the region outside the key area is populated. This is important, for these points are noise, and it is desirable that they be as few as possible. Obviously an image consisting of nothing but ones would be able to detect every point in the key, but it should be severely penalized for its inclusion of all the rest of the image points. This term performs the necessary penalization.

The second term is simply the fraction of key-points detected by the operator, and is a hit-rate, or measure of success within the key.

In general for an operator to score well it must be able to detect a large portion of the key-points while detecting the least possible number of points outside the boundary of the key. Such a prioritization is mathematically inherent in the correlation approach.

By no means is the correlation form the only method of computing the absolute grade. Any formalization which deals with the inverse relationship between grade and number of edge points found outside the key, as well as the direct relationship between grade and hit-rate, while producing consistent, meaningful results is equally as valid. By meaningful results is meant the ability to score operators faithfully,

with clearly superior results always getting the higher grade.

Just such a formulation is given in equation 7.

$$R = \left(1 - \frac{N_b}{N^2 - N_k} \right) \sqrt{\frac{N_c}{N_k}} \quad 0 \leq R \leq 1 \quad (7)$$

where N_b is the number of points detected outside the key area, N^2 is the number of possible points = $M \cdot N$, and N_c and N_k as before.

The first term of equation 7, like that of equation 6, adjusts the grade according to the number of nonkey noise points. Actually, this term is the fraction of correctly located nonkey points. That is, the fraction of the pixels in the area beyond the key boundaries which are zero.

The second term is the hit rate inside the key, with the square-root included to impart more importance to this value.

Examination of the elements of equation 7 reveals that the range of R , as with the correlation approach, is from zero to one inclusive. This facilitates the expression of R as a percent, with 100% being the best possible grade.

One additional consideration which is of interest in both these forms of absolute grading is the difference in detected point densities within and beyond the key area. A reasonable interpretation of the absolute grade would be a measure of the increase in the detection fraction in these two areas. Detection fraction is the fraction of the maximum number of detectable points which are actually selected. For the key area the detection fraction is

$$D_k = \frac{N_c}{N_k} \quad (8)$$

while for the area beyond

$$D_b = \frac{N_b}{N^2 - N_k}$$

The greater the difference between D_k and D_b , the higher the grade. If D_k is less than or equal to D_b the grade should be zero, for this suggests an operator which is not detecting the key edge at all.

This test should be incorporated into the computation of the absolute grade, no matter what formulation is used. In fact, a very simple grade would be

$$\begin{aligned} R &= D_k - D_b & D_k > D_b & & 0 \leq R \leq 1 \\ &= 0 & D_k \leq D_b & & \end{aligned} \quad (9)$$

It should be evident that the thresholding scheme employed will have a very great effect on the grade. If the threshold level is too high, the grade will be reduced due to the decrease in the number of common points. A threshold that is too low will reduce the grade by including a larger number of noise points.

A system for finding the most successful threshold would involve computing the absolute grade over a wide range of threshold parameters.

Those values which yield the largest grade could be considered optimal in terms of the particular key-edge sought.

Operator Improvement

Mention was made in Chapter 2 of the potential absolute grading has for facilitating an iterative improvement of operator output. While the grading does not itself make any changes, it offers a means of determining the amount of success or failure produced by modifications in the edge location process.

There are two basic ways in which the grade obtained for a particular operator can be improved; either the number of points detected outside the key-area can be reduced or the number inside increased. It should be remembered that any scheme for carrying out either or both of these improvements be applied throughout the entire scene. That is to say, the modification process should be unbiased by the location of the key area.

The first form of improvement is basically a noise elimination process, designed to locate and remove isolated sets of points whose population is insufficient to be considered an edge. Most frequently the approach is to delete solitary points, under the general assumption that they are artifacts of the thresholding step.

For the general case, where sets of points with population less than some value P are to be deleted, the neighborhood of points which must be scanned has a side of length $2p+1$. Within this neighborhood, a search is made, beginning at the center point for a point string. If one of length p (or less) is present, it will be found by tracing its boundary and

checking the area covered. In the simple case; where $p=1$, a three-by-three neighborhood is necessary, but the checking procedure reduces to determining if (x,y) is the only nonzero point in the neighborhood.

Removing isolated points should improve the absolute grade by eliminating spurious returns from the nonkey area of the image. If the grade is lowered after the removal process, it is because it was artificially high to start with or the key was not properly constructed.

Edge detector characteristics and resolution error in digitizing can transform the normally narrow edges in a scene into a broad transition covering many pixels. This is especially true of large mask operators, with the size of the mask being directly related to the degree of smearing. The uncertainty of location which results is often undesirable and can have a detrimental effect upon the operators overall performance.

Additionally, it is often noticed that lengthy edges in a scene suffer from apparent discontinuities, which appear as skipped points, or holes in the thresholded operator output. While humans can compensate for this quite readily, it is not simple to rectify from magnitude information alone.

It is in these areas that the directional information produced by some of the techniques mentioned in Chapter 2 plays an important role.

Edge orientation measures provide a second view of imagery that is quite different from magnitude operators, and offers valuable additional knowledge about the scene. The inclusion of this data in an edge detection system offers increased capacity to refine magnitude detection

results by post-processing and selective output enhancement by both error point removal and missed point identification.

The problem of edge smearing can be remedied by a technique known alternatively as association or thinning. An examination of the general vicinity of an edge might reveal something similar to Figure 12.

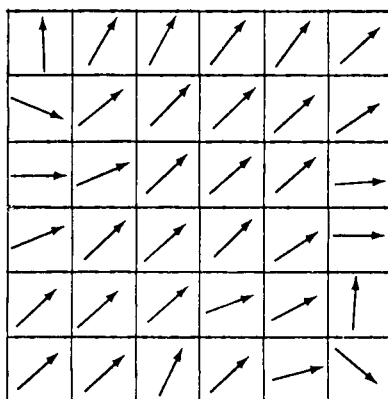


Figure 12. Typical directional pattern of an edge. Direction is indicated by the arrows.

Clearly there is some edge in this subarea, but it is smeared and has become overspecified. An easier to interpret and perhaps more accurate presentation of the edge would result from the elimination of the redundant information.

Association seeks to narrow these wide edges by merging neighboring points with similar magnitude and direction values. The magnitude and direction information for point (x_1, y_1) are obtained and the coordinates of the nearest neighbors (x_2, y_2) and (x_3, y_3) are obtained by moving in directions perpendicular to that of (x_1, y_1) . If there is sufficient similarity between the magnitude and direction of (x_1, y_1) and (x_2, y_2) or (x_1, y_1) and (x_3, y_3) then (x_1, y_1) is eliminated.

The similarity measure takes the form of either a difference or an inequality. A difference measure involves computing the percentage difference in magnitude and absolute difference in angle between (x_1, y_1) and its neighbors. If both these differences are less than a particular pair of preset thresholds, an association is made.

The inequality form is less variable but also less flexible. The association is made if the magnitude-angle ordered pair for each of two points satisfies some inequality. An example is the association criteria of Bowker [15]. He advocates association if

$$E_2 \cos (\theta_2 - \theta_1) > E_1 \quad (10)$$

where E_1 and θ_1 are the magnitude and angle values for (x_1, y_1) respectively, and E_2 and θ_2 are the values for (x_2, y_2) .

Association reduces the number of detected points. This acts to improve the grade by diminishing the fraction of points in the area outside the key, while doing little to alter the situation inside the key area. The improvement made depends greatly upon the characteristics of the key itself. Generally, the key is constructed from the original picture and is narrow like the edge it is patterned after. This type key lends itself to association improvement. On the other hand, if the key is broad, the use of association could very well bring about a reduction in the grade. Often a broad key is not the proper choice, but is used because it is much easier to build. This must be considered before employing association.

While association is quite successful at thinning out the population of extra-key points, and thus improving the absolute grade, it does not attempt to compensate for breaks in the edges which result from a detector/thresholding system. These discontinuities can often be eliminated by linking together contiguous pixels.

Like association, linking involves searching along directions implied by the edge orientation of a particular edge point. Unlike association, however, a link-search is along the line of the edge. If the next point bears sufficient resemblance to the one before it, they are connected together. This searching must be carried out in both a forward and reverse direction to establish the full extent of the edge. Provisions must be made to skip over breaks and checking out the next nonzero point. If it appears the edge continues, the break can be filled in and the search continued. After the length of an edge is determined, it is matched against a threshold to see if it is long enough.

Throughout this searching, thresholds which define the linking criteria are applied in order to make the connecting decisions.

There are four basic thresholds: edge length which defines the minimum number of connected points that can be considered an edge, skip threshold which establishes the maximum number of adjacent breaks that can be tolerated in an edge, magnitude difference, or the percentage difference allowed between the magnitudes of two contiguous points, and angle difference, which is the maximum angular deviation permitted among adjacent edge points.

Special-case versions of linking reduce the number of thresholds. A commonly implemented [8] form, known as connectivity testing, or angle linking deals only with angle information. The direction of a particular point is compared to that of the preceeding and succeeding points. If the difference in angle is less than the angle difference threshold, the points are linked together, regardless of the difference in magnitudes.

Quite a bit of flexibility is present in the linking process, but the large number of thresholds which must be established introduces an amount of uncertainty and leads to trial-and-error processing. The angle difference threshold is directly related to the minimum radius of curvature allowed for an edge. For the linking to be meaningful, this threshold cannot be very large, but it must not be so small as to disallow curved edges. This same sort of boundary condition problem exists with all the thresholds. The edge length threshold must be large enough to effectively eliminate isolated short edges without being unreasonably large. The skip threshold should not permit obviously independent edges to be connected, yet still must be capable of bridging breaks in edges.

Linking serves several basic functions in terms of absolute grade improvement. First, it is able to eliminate sets of points which are not of sufficient length to be considered edges. This will reduce the number of detected points in the nonkey portion of the image. The more nearly the thresholds characterize the key, the better the rejection should be.

Secondly, linking is able to fill in some of the breaks in the edges of interest, hopefully increasing the number of detected points within the key. The increase of noise points which results, if any,

should be small enough, provided the thresholds are reasonably selected to still result in an increased grade.

This basic linking system is by no means the only one available. Others have been developed [16] which involve linking hierarchies and multi-step algorithms. As with edge detectors, the possibilities for linking algorithms are fairly endless. The fundamentals, however, are the same as those employed in the basic approach, and their operation reflects the same concepts and principles. Their general thrust, and form of utilization in connection with absolute grading must of necessity remain the same - improve, if possible, the grade.

As mentioned previously, the construction of the key plays an important role in the development of a grading system, and has a direct bearing upon the range and accuracy of the grade. Such being the case, certain considerations are of interest if the key is to be of sound design.

First, and foremost, it must accurately reflect the characteristics of the principle edge. It must be of similar width, contour and position. This calls for a careful examination of the edge in the original scene. In actuality, the human is called upon to play the role of ideal edge detector, seeing to it that the representation is as exact as possible.

Additionally, the key must be of reasonable size. That is, it must be of near the same extent as the principle edge. The shorter the edge of interest, the more important it becomes that the key be of the same length. This is to insure that a sufficient number of samples are taken, in order to make the grade statistically meaningful. A longer key is a more unique key, and is able to more accurately determine the "best" operator.

Absolute grading, like relative grading can be used to evaluate thresholding schemes and, in an iterative manner, provide a most successful set of thresholding parameters. The addition of association and linking to this procedure imparts another means of altering the operator's output, with the end product being an edge detection system which does the best job of locating a particular class of edge.

The general approach is to use the grade as a form of feedback to the thresholding-association-linking system, allowing the parameters to be fine tuned so as to produce the highest grade. This directed development provides a closed-loop, mathematically defined means of building up a successful system for edge location.

Absolute grading, then, can be employed in two fundamental modes. It can evaluate an operator's ability to locate a specific principle edge, or it can be employed to measure quantitatively the performance of any edge detection process. In either mode it is compatible with association and/or linking schemes as operator improvement techniques.

CHAPTER 5

TYPICAL IMPLEMENTATION AND RESULTS

Relative and absolute grading, along with their accompanying enhancement techniques, are the building blocks of a large number of different, flexible edge operator evaluation and improvement systems. The principles involved in assembling one of these systems, as well as an indication of their operating potential can best be demonstrated by example. Several distinct approaches are available, each of varying emphasis and complexity.

Of great importance is the realization that, no matter what form of system is constructed, the details of the process of grading are data independent. Naturally the data input, and the edges of interest will have great effect upon the direction the grading will take. Nonetheless, the fundamentals of the grading system and the general tendencies of those improvement techniques employed will not be altered, no matter what the data under study is. This is due entirely to the way in which the formulation of both relative and absolute grading was developed.

In addition, the grading system is structurally independent of the operators which are to be tested. Despite differences in type, form or function, the operators can only alter the grades produced, not the concepts or algorithms used in obtaining the grade.

Such being the case, it is necessary only to exhibit the general principles involved in grading, while also outlining the important

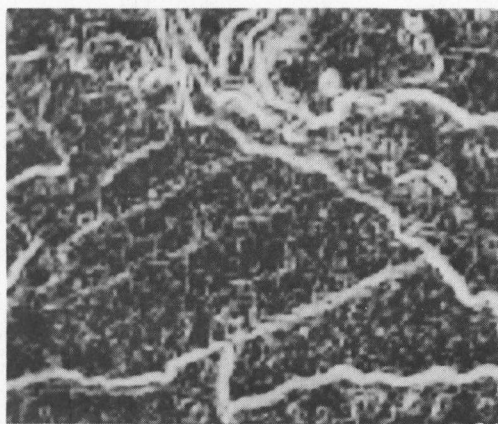
considerations which are encountered in the development of an operator evaluation system.

A typical image is shown in Figure 13. It is an aerial photograph of a rural area near Ava, New York, and contains relatively few but rather irregular edges.

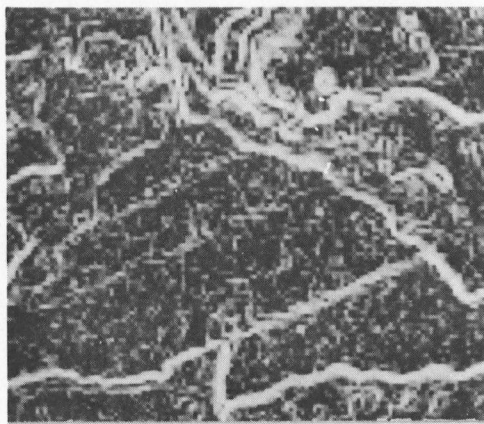
The operator set tested is that outlined in Chapter 2, namely the gradient, Robert's gradient, Robert's square-root gradient, range, Sobel, and Frei-Chen operators. For those situations requiring directional information, the Davies system, also discussed in Chapter 2, is used. These operators are well known, frequently used, and offer a wide range of simplicity and approach. Results of applying this operator set to the test scene of Figure 13 are shown in Figure 14.



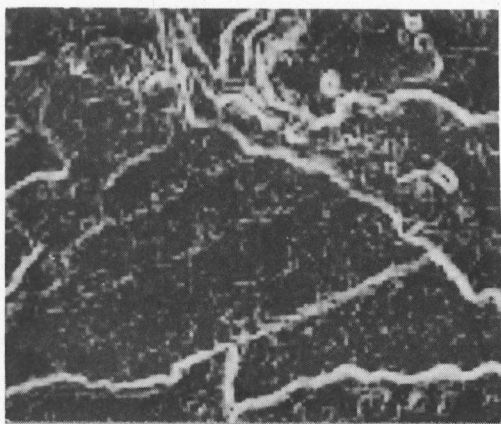
Figure 13. Test scene. Aerial photograph of area near Ava, New York.



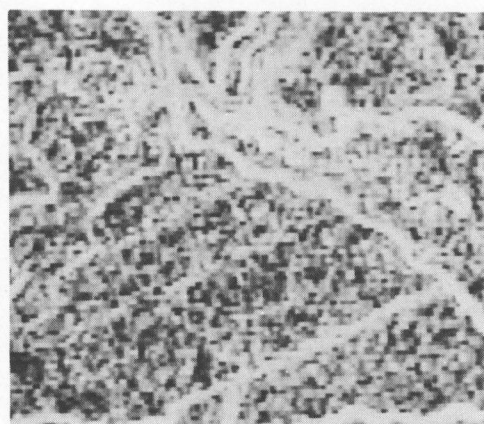
(a)



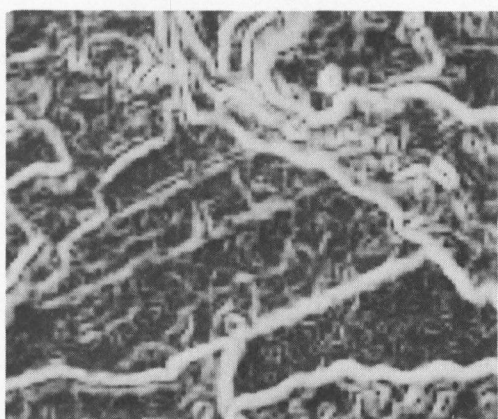
(b)



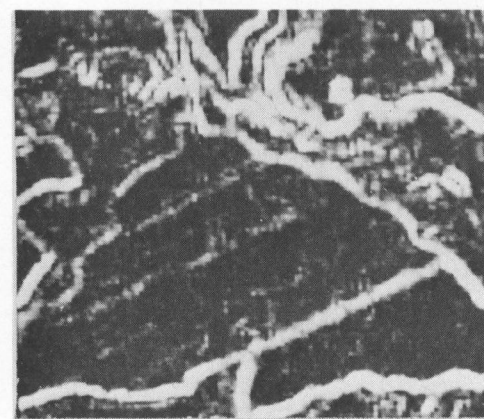
(c)



(d)



(e)



(f)

Figure 14. Operator test set output. (a) Gradient, (b) Robert's gradient, (c) Robert's square root gradient, (d) Range, (e) Sobel, (f) Frei-Chen operators.

Thresholding the images of Figure 14 at the 10% level as indicated by Davies [5] produced the binary edge maps of Figure 15. These maps illustrate those points which are to be considered edge points in white, with the nonedge background shown in black.

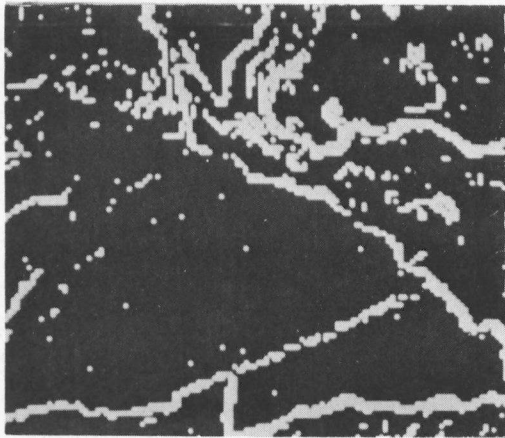
As such, these binary edge maps are the final product of six different edge operators. Brief examination reveals many differences and disagreements as to the edge state of points. At this point they are candidates for either form of grading.

Relative Grading Results

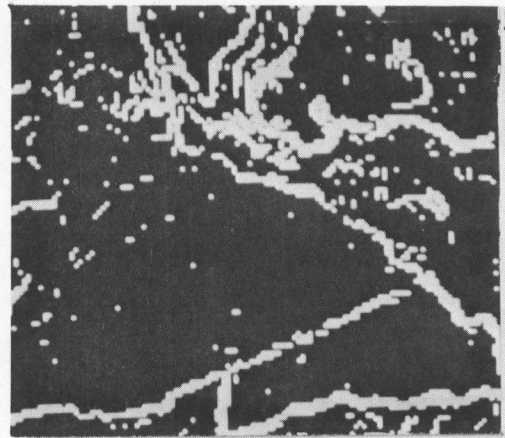
An equalized, exclusion-voting relative grading system with a "no" vote threshold of two produced the grades of Table 1. The "no" vote threshold is the number of nonedge votes required to produce a joint nonedge decision.

Columns one and two indicate the edge score for each operator. The first number is the number of times a particular operator agreed with a joint edge decision, with column two indicating this total as a percentage of the number of joint edge decisions made. Columns three and four give the same information for the nonedge, or background points. The last two columns are the totals for each operator, the score being computed as the percentage of the maximum 16384 decisions which were made correctly.

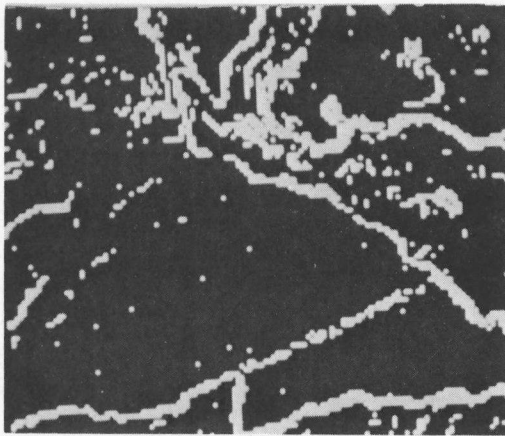
The outcome is divided into edge and background tallies to allow separate comparison of relative successes among operators on either their ability to detect edges or determine background, as well as evaluate their overall performance.



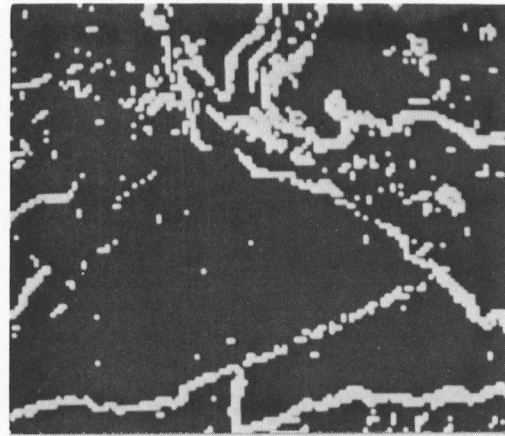
(a)



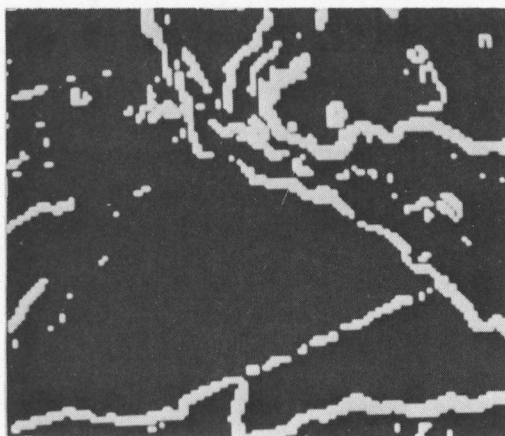
(b)



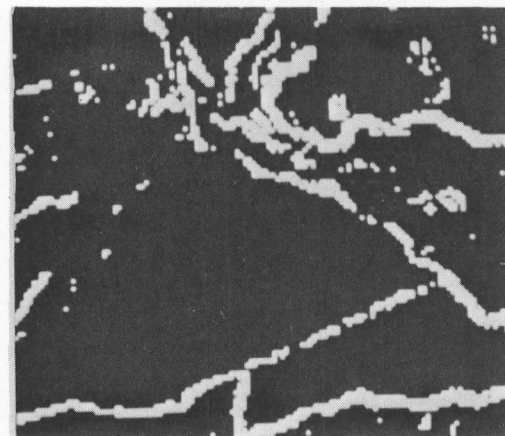
(c)



(d)



(e)



(f)

Figure 15. Threshold at 10%. (a) Gradient, (b) Robert's gradient, (c) Robert's square root gradient, (d) Range, (e) Sobel, and (f) Frei-Chen operators of Figure 14.

Table 1

Relative Grading Results "No" Vote Threshold = 2
Threshold Level = 10%

OPERATOR	EDGES	SCORE (%)	NON-EDGE	SCORE (%)	TOTAL	NET%
Sobel	1071	100.000	14656	95.716	15727	95.966
Robert's	1048	97.852	14276	93.234	15324	93.536
Roberts Sq.	1071	100.000	14400	94.044	15471	94.433
Gradient	1063	99.253	14510	94.762	15573	95.056
Range	1027	95.892	14492	94.645	15519	94.726
Frei-Chen	1071	100.000	14627	95.526	15698	95.819

Total of 1071 concensus edge points.

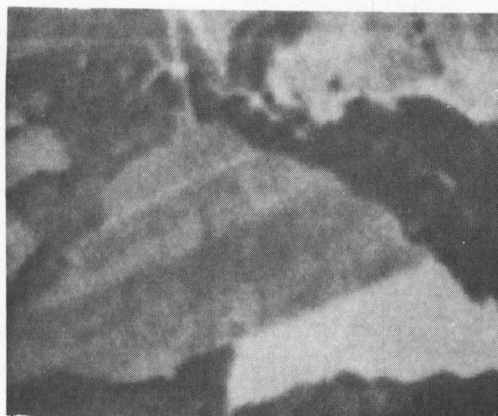
From the results of Table 1, it can be inferred that the Sobel operator was superior. Along with Robert's square-root gradient and the Frei-Chen operator, it correctly located all 1071 consensus edgepoints while also correctly classifying 14656 out of the 15313 consensus nonedge points. This indicates that the Sobel operator misclassified 657 background points as edges. Under these conditions, then, the Sobel operator was best able to anticipate the consensus edge map.

In addition to the grades produced, relative grading generates a consensus edge map, with those points mutually agreed upon as edge points indicated as such. For this particular grading, the consensus map is shown in Figure 16. Also shown are the original scene and the output of the Sobel operator.

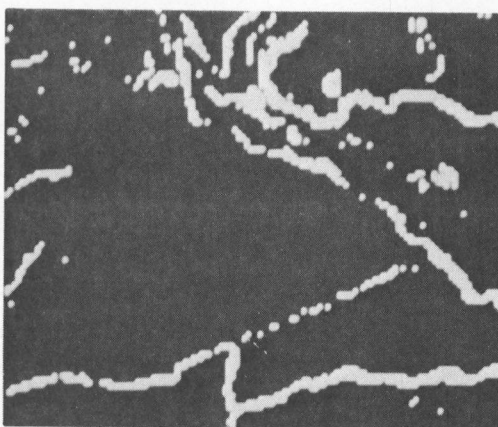
Changing the "no" vote threshold yields the grades of Table 2. The Sobel operator still achieved the highest score, but it lost its superiority in edge point detection to Robert's square-root gradient. This change was due to an increase in the number of consensus edge points, most of which had been previously considered background. Figure 17 shows the common edge point map and the Sobel operator output.

Another parameter which would change the grade, and perhaps the outcome of the grading process, is the original threshold level. While the 10% choice might appear visually to be superior, another value might be more appropriate in terms of relative grade.

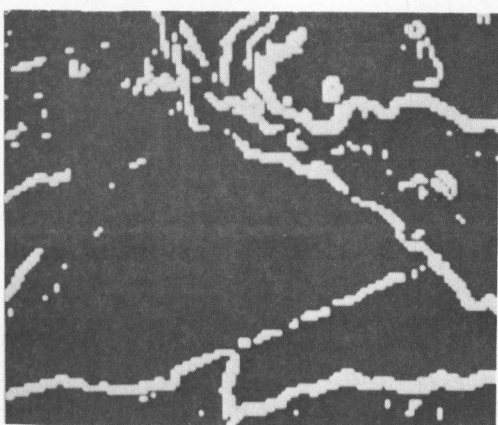
The results of choosing a 15% threshold are shown in Table 3. Figure 18 shows the consensus result, and the Sobel operator, which again did the best job of anticipating the common edge points.



(a)



(b)



(c)

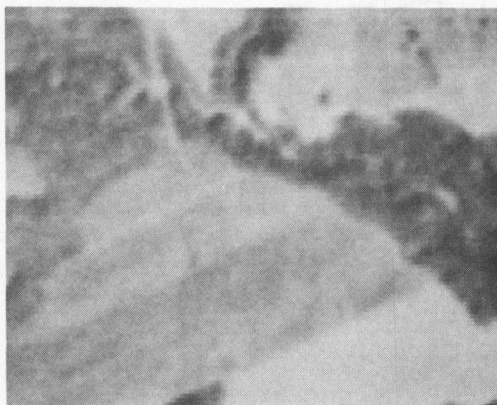
Figure 16. Relative grading results with "no" vote threshold of two and threshold level at 10%. (a) Original, (b) Consensus map, (c) Highest scoring operator (Sobel).

Table 2

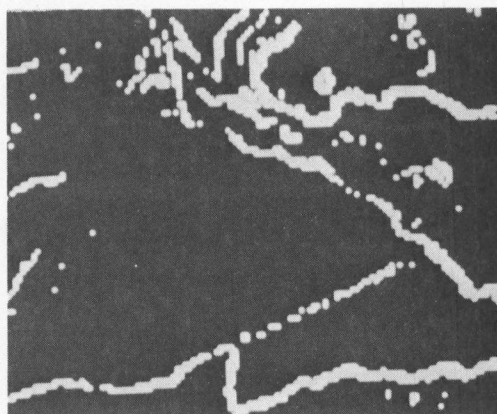
Relative Grading Results 'No' Vote Threshold = 3
Threshold Level = 10%

OPERATOR	EDGES	SCORE (%)	NON-EDGE	SCORE (%)	TOTAL	NET%
Sobel	1170	94.737	14643	96.666	15813	96.521
Robert's	1196	96.842	14263	94.158	15459	94.360
Robert's Sq.	1226	99.271	14356	94.772	15582	95.111
Gradient	1196	96.842	14461	95.465	15657	95.569
Range	1185	95.951	14435	95.293	15620	95.343
Frei-Chen	1167	94.494	14609	96.442	15776	96.295

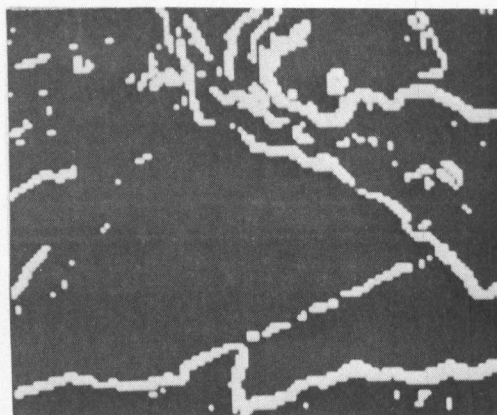
Total of 1235 consensus edge points.



(a)



(b)



(c)

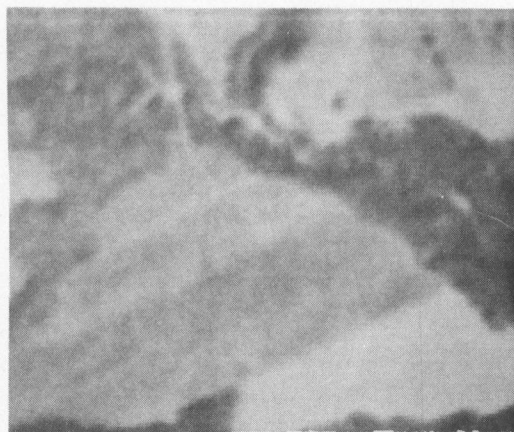
Figure 17. Relative grading results with "no" vote threshold of three and threshold level of 10%. (a) Original (b) Consensus map (c) Highest scoring operator (Sobel).

Table 3

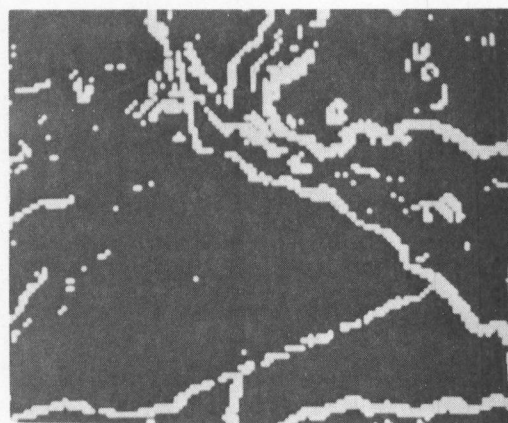
Relative Grading Results "No" Vote Threshold = 2
Threshold Level = 15%

OPERATOR	EDGES	SCORE (%)	NON-EDGE	SCORE (%)	TOTAL	NET%
Sobel	1621	100.000	13809	93.544	15430	94.183
Robert's	1540	95.003	13612	92.210	15152	92.486
Robert's Sq.	1619	99.877	13549	91.783	15168	92.584
Gradient	1621	100.000	13384	90.665	15005	91.589
Range	1545	95.312	13617	92.244	15162	92.547
Frei-Chen	1621	100.000	13772	93.294	15393	93.957

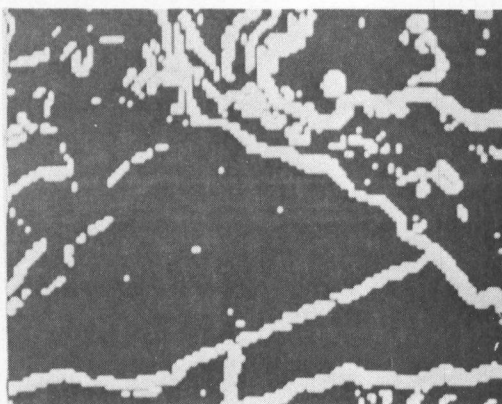
Total of 1621 concensus edge points.



(a)



(b)



(c)

Figure 18. Relative grading results with "no" vote threshold of two and threshold level at 15%. (a) Original, (b) Concensus map, (c) Highest scoring operator (Sobel).

No matter what parameterization is employed, the meaning of a relative grade is the same. Basically, it determines the points which are considered by a majority of the operators to be edge points, and measures how closely each operator came to this common result. In some applications, the consensus edge map might be the important result. Only in a few cases, however, will there be sufficient resources to permit such lengthy computation. It is more reasonable to expect that relative grading will be used to discern the most accurate operator for a particular situation.

Various relative grading results, such as those of Tables 1, 2 and 3, can be compared by superimposing their receiver operating characteristics (ROC) curves. An ROC curve is a plot of system probability of detection versus probability of false alarm. For an edge detection process, the probability of detection is the fraction of edge points that were correctly classified, while the probability of false alarm is the fraction of nonedge points misclassified as edges. In terms of the confusion matrix of Figure 11 (page 22), this amounts to a graph of the true positive percent—against the false negative percentage.

A ROC curve for all three relative grading systems is shown in Figure 19. Interpretation of this and other ROC curves is fairly straightforward. An ideal receiver would have a probability of detection of one and a probability of false alarm of zero. This corresponds to the upper-left corner point in the graph. The nearer a receiver's ROC curve comes to this point, the better its operation. From Figure 19 it can be inferred that, for all cases, the system with a 10% threshold performs

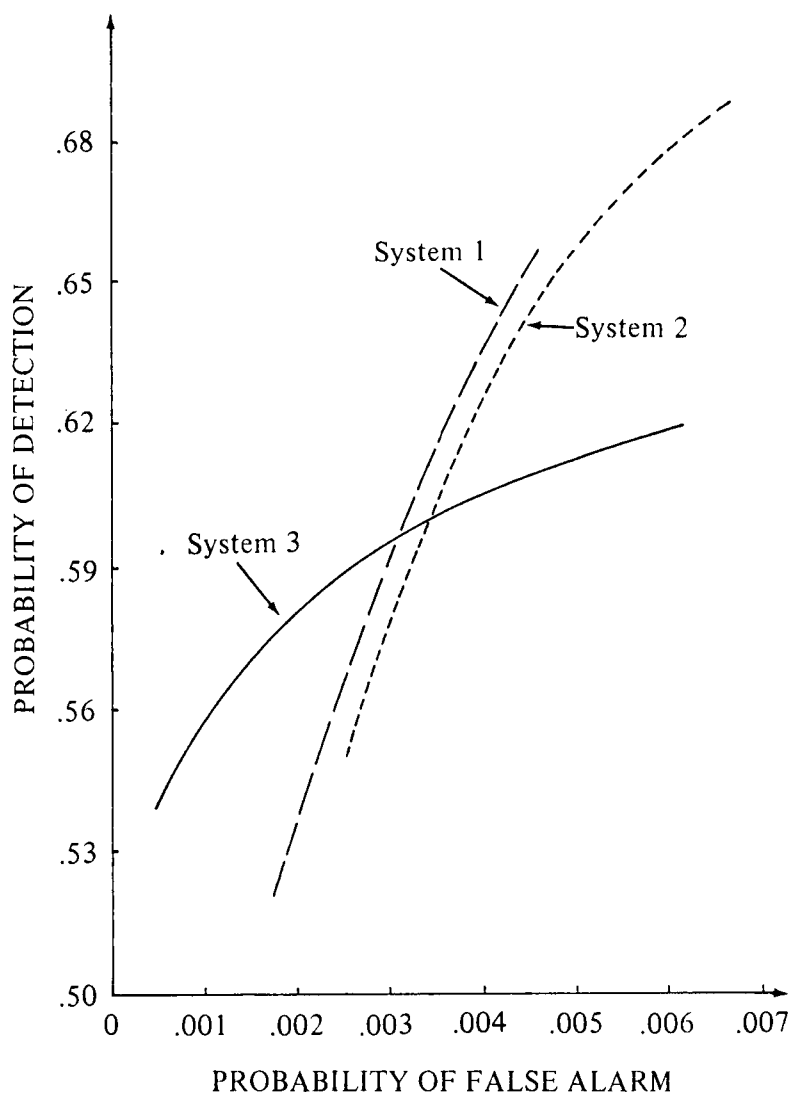


Figure 19. ROC curves for relative grading systems of Tables 1, 2, and 3.

better if the "no" vote threshold is two. Additionally, when the probability of false alarm is greater than 0.0032, the system with a 10% threshold and "no" vote threshold of two outperforms the other systems. If the probability of false alarm is less than 0.0032, a shift of the threshold level to 15% will improve the results.

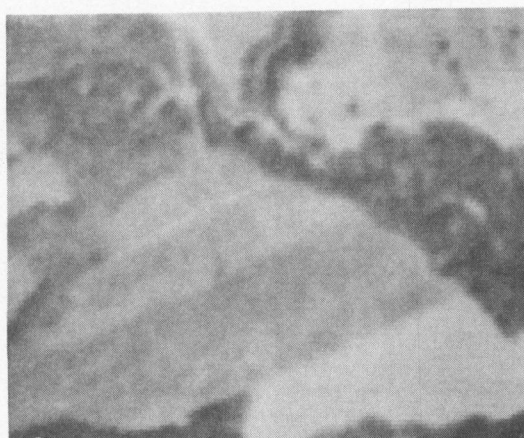
Absolute Grading Results

Many edge detection problems require the "best" operator to be the most successful in locating a certain important principle edge. Absolute grading offers a method of making that "best" choice.

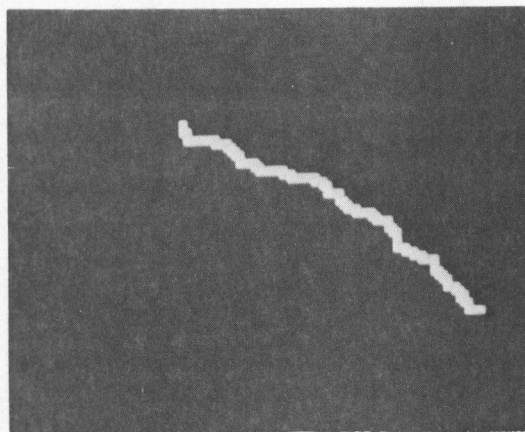
The single most critical element of any absolute grading system is the key. As discussed in Chapter 4, the key must be carefully constructed. Figure 20 shows both a good and bad key for the data scene indicated. The good key is roughly the same length and width as the selected principle edge, while the bad key is too short and broad. An improper key can lead to useless, invalid grades, and far less than optimum results.

Unlike relative grading, there are many formulations of absolute grading. This can lead to difficulties in interpreting an absolute grade in terms of performance. To provide a standard of comparison, a very poor operator can be graded under the same conditions and used as a worst-case measure.

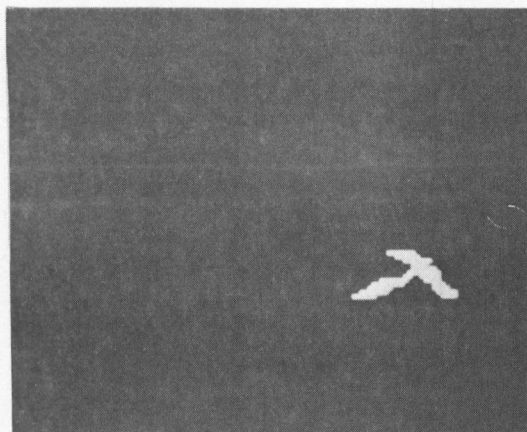
A good example of a poor operator is the random number generator. For a particular point, a uniformly distributed random number is generated. If it is greater than a preset value, that point is considered an edge detector, although it does not have anything to do with the actual presence or absence of an edge at a point.



(a)



(b)



(c)

Figure 20. Sample keys. (a) Original, (b) good key, (c) bad key.

Using the key of Figure 20 and the augmented operator set, which includes the poor operator and is thresholded at 10%, the absolute grades of Table 4 are obtained. The three different absolute grade formulations mentioned in Chapter 4 have been used. The correlation grade, R_c based upon its origin, is probably the most reliable. To be thorough, the fractional grade, R , and the difference grade R_d , are included. The general validity of these two, and any other formulation, is determined by grade consistency, and agreement with human interpretation as well as other established formulations.

From these results, it is evident that Robert's square-root gradient was most successful at detecting the key edge, while minimizing nonkey detections. Thus, this operator would be selected as most able to detect this type edge under similar conditions.

Absolute grading has the capability of evaluating operator improvement. The improvement achieved by such techniques as single point elimination, association and linking, as well as the effect of changes in threshold level can be directly ascertained from noting the change in grade.

The question of the validity of the 10% threshold can be answered by computing the grade obtained by other thresholds. This data is tabulated in Table 5 for the Sobel operator and the original scene of Figure 13 (page 49). The variation in R_c can most easily be seen from the graph of Figure 21. The threshold which scored the highest was 11%, although close agreement was obtained from 9% through 20%.

Table 4
Absolute Grade of Operator
Test Set

OPERATOR	R_c	R	R_d
Sobel	20.446	65.917	43.549
Robert's	19.868	66.772	45.555
Robert's Sq.	21.625	69.335	49.837
Gradient	20.652	67.137	45.771
Range	20.613	67.105	45.728
Frei-Chen	21.217	67.317	45.854
Random	5.810	36.810	1.278

Table 5

Absolute Grade of Sobel Operator Vs.
Threshold Level

THRESHOLD (%)	R_c	R	R_d
5	15.009	49.988	22.802
6	16.397	54.047	27.173
7	18.153	58.584	32.709
8	19.967	63.041	38.949
9	20.687	65.333	42.485
10	20.446	65.917	43.549
11	20.789	67.249	45.920
12	20.341	67.328	46.381
13	20.690	68.563	48.940
14	20.587	68.712	49.492
15	20.710	69.430	51.460
20	20.202	69.332	54.637
22	19.556	68.058	53.926
25	18.887	66.471	52.712
27.5	18.592	65.145	52.933
30	18.024	63.068	51.522
32.5	17.498	61.185	49.677
35	16.945	58.981	47.492

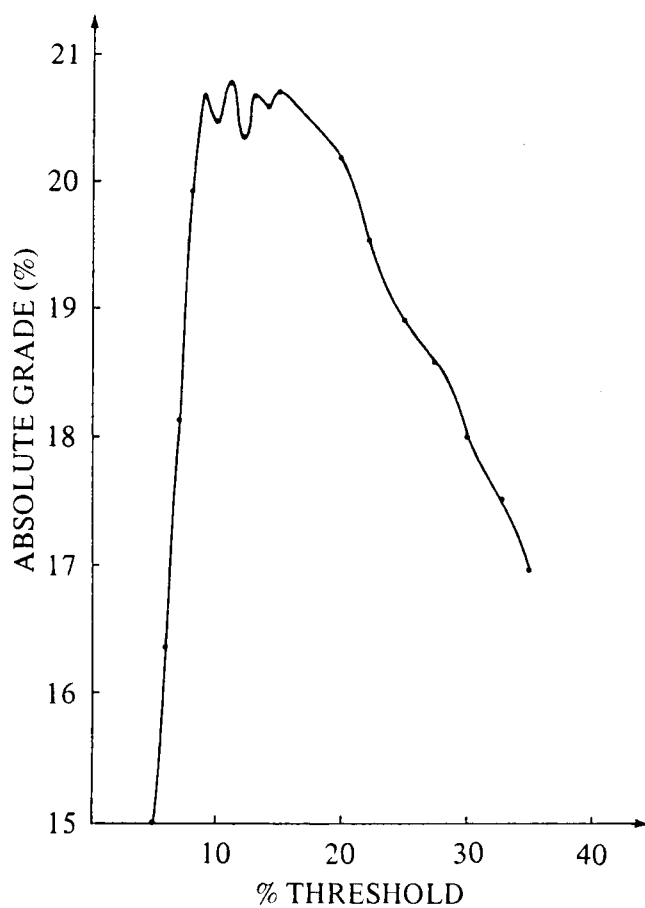


Figure 21. Absolute grade of Sobel operator vs. threshold level.

Several pre- and post-thresholding enhancement techniques could improve the grade as well. One approach would be to convert the absolute change implied by the operator's output to a relative change as discussed in Chapter 3. The thresholding that follows is based upon the magnitude of the relative change. Results for this improvement are given in Table 6.

The improved grade of 20.957 at the 55% change level is an improvement of 0.168 which, although small, is significant. This transformation produces a visually noticeable change in the operator's result, as can be seen from Figure 22.

A post-thresholding technique, single point elimination, involves the zeroing of any point which has no edge points as neighbors. The general assumption is that these points represent noise, or incorrectly classified edge points.

Single point elimination can be used alone, or in combination with other grade improvement schemes. Applying it to the optimally thresholded Sobel operator, and to the result of the relative change transformation approach produced the results of Table 7.

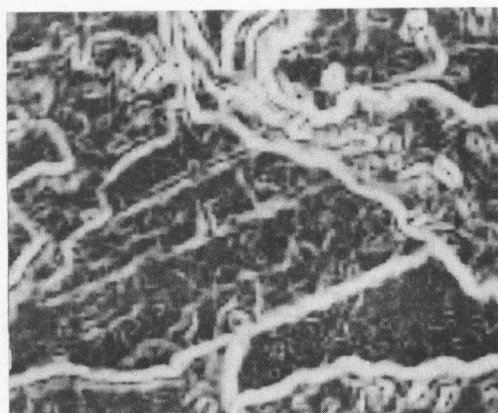
These techniques operate solely on magnitude information. Two additional operator improvement schemes, association and linking, further refine the results by incorporating directional information into the enhancement process.

Association, or edge thinning, acts to improve the grade by thinning and isolating the major edges, overcoming the smearing which can lead to a lower grade. It can be carried out on any operator, provided the

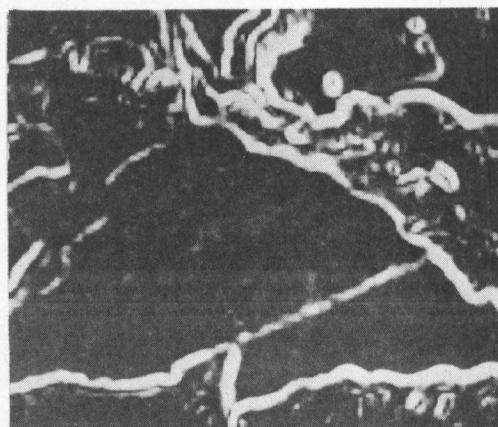
Table 6

Absolute Grade of Sobel Operator vs. Threshold Level
For Relative Change Transformation

$\Delta\%$	THRESHOLD LEVEL	R_c	R	R_d
30	77	17.916	63.619	50.328
35	89	19.151	67.206	52.947
40	102	19.713	68.417	51.791
45	115	19.596	67.543	48.017
50	128	19.970	66.871	45.685
55	140	20.957	66.988	45.338
60	153	20.650	64.487	41.124
65	166	19.419	60.430	35.280
70	179	18.170	56.005	29.730



(a)



(b)

Figure 22. Relative change transformation of Sobel operator.
Transformation of (a) yields (b) New Sobel output.

Table 7

Results of Single Point
Elimination

OPERATOR	R_d	R	R_d
Sobel at Optimum Threshold	21.046	67.453	46.193
Relative Change Sobel	21.087	67.085	45.468

appropriate directional information is available from the Davies or similar system. The output of an association process is a modified magnitude image. This image is then thresholded to produce a binary version of the narrow centerline of the major edges. By slightly enlarging the edges to make them more connected, an improved edge map can be produced.

The results of association on the Sobel operator are shown in Table 8 and Figure 23. By using association improvement, the grade of the Sobel operator was improved significantly from 20.789 to 22.474. If the threshold value is too small or too large, the benefits of association are eliminated by the excess thinning or thickening of the central edges.

Linking is potentially the most powerful technique for operator improvement, for it is able to reject edges shorter than a certain length. This allows almost all small edges and isolated points to be removed, while retaining the long major edges.

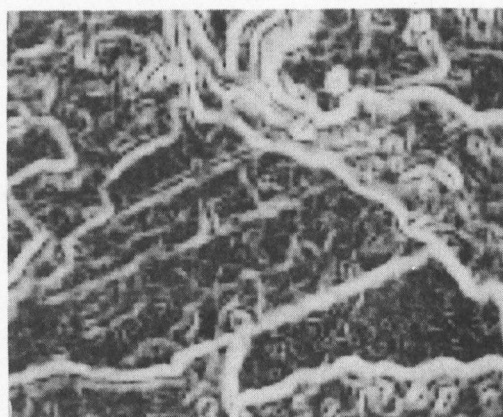
The simplest form of linking, known as connectivity, connects contiguous points as long as their angle components are within a certain tolerance, regardless of their magnitudes. Performing connectivity testing on the Sobel operator's interpretation of Figure 13 (page 49), produces the results of Table 9 and Figures 24 and 25.

The improvement achieved is striking, with the grade being nearly doubled. Visual interpretation of the results indeed confirms that linking produced an edge map which is predominantly made up of the key edge. An ideal operator would detect only the key, or target edge. Use of linking results in an "operator" which approaches this goal.

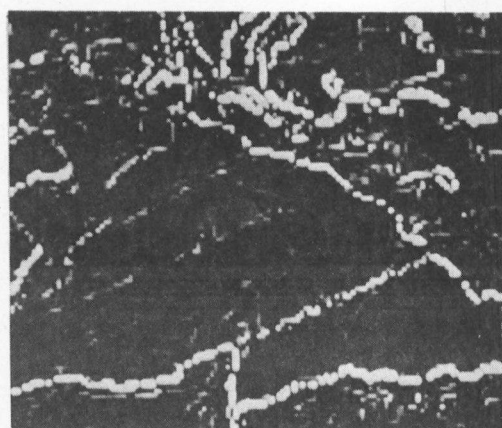
Table 8

Association Improvement
of Sobel Operator

THRESHOLD (%)	R_d	R	R_d
2	17.136	59.483	33.741
3	22.057	70.926	53.301
4	22.474	72.943	59.722
5	21.381	71.359	60.067
6	20.000	68.308	57.754
7	18.939	65.382	54.994
8	17.864	61.881	51.402



(a)



(b)

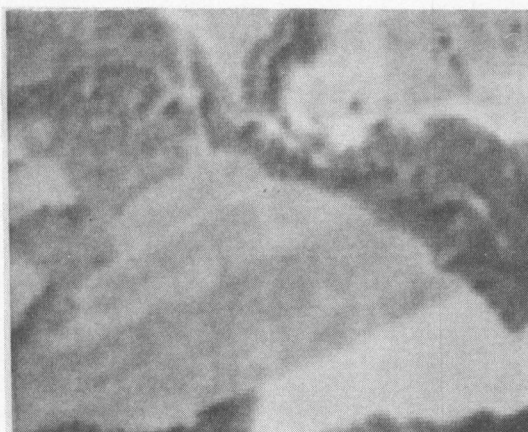
Figure 23. Results of association. (a) Sobel operator produces
(b) Thinned Sobel output.

Table 9

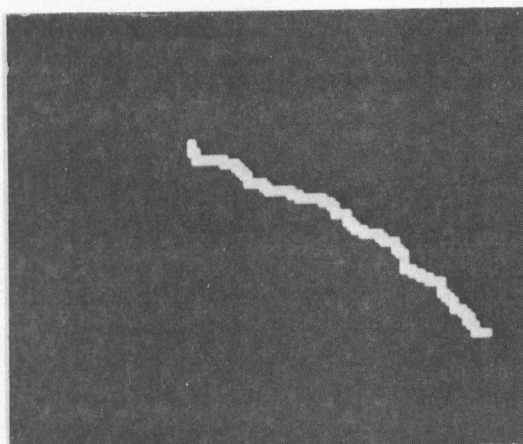
Linking Improvement of Sobel Operator

LENGTH THRESHOLD	R_d	R	R_d
2	22.252	71.687	55.236
3	23.150	71.636	53.784
4	23.418	70.632	51.365
5	24.491	70.094	49.936
6	24.796	68.318	48.846
7	26.012	67.359	45.270
8	27.253	66.373	43.801
9	25.515	62.327	38.218
10	23.832	57.973	32.759
11	26.250	58.331	33.333
12	27.387	58.468	33.583
13	27.900	57.294	32.254
14	31.941	57.608	32.793
15	33.147	57.680	32.917
16	33.478	52.882	27.677
17	33.915	51.788	26.562
18	37.390	51.879	26.735
19	37.264	51.497	26.343
20	36.905	48.332	23.222
21	37.090	48.335	23.228
22	37.090	48.335	23.228
23	32.248	43.516	18.787
24	27.554	39.128	15.143
25	27.005	38.610	14.738

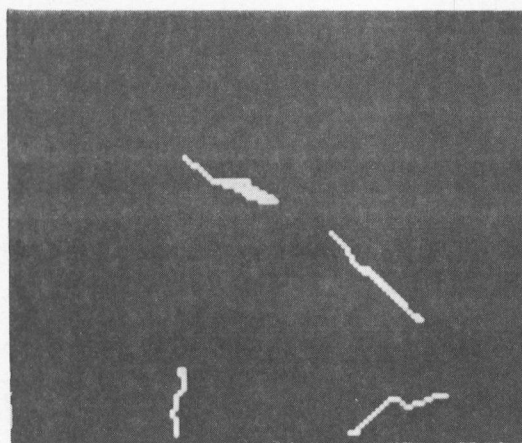
With 45° maximum connectivity angle.



(a)



(b)



(c)

Figure 24. Linking results. (a) Original, (b) key, (c) best linking output.

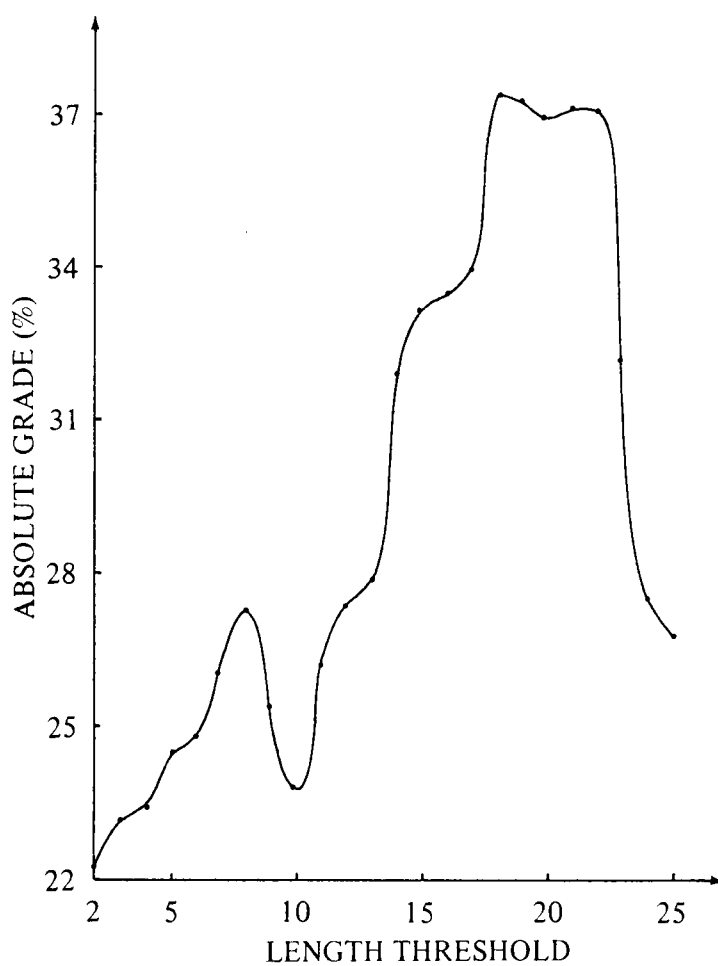


Figure 25. Plot of absolute grade vs. length threshold of linking process for data in Table 9.

Both linking and association are best used as post-thresholding improvement techniques. Their operation is greatly expedited by removing as many points as possible from the images which they operate upon. For this reason, some improvement is often helpful before using either linking or association to achieve a superior operator output.

Many possible combinations of grading and improvement techniques exist. They are all equally viable as long as the general principles are followed.

It should be noted that, although three different absolute grades were calculated for each case, the correlation grade was treated as the most accurate, reliable grade. This is due to its more conceptually solid foundation and consistent agreement with human interpretation.

General, overall operator performance can be evaluated by computing the absolute grade over a wide range of edge types and environments. The average grade which results is indicative of typical operator success, as long as the grades are arrived at in a consistent, thorough manner.

Consider the two additional test scenes shown in Figure 26. Along with the scene of Figure 13 (page 49), these images typify the most common forms and configurations of edges, ranging from a close up view containing curved edges, to a medium distance building with sharp, orthogonal edges and considerable structure, to a high altitude aerial scene with much texture and natural edges.

This three member test set can be used to obtain a measure of the general performance of the operators used previously in this chapter. The additional keys required are shown in Figure 26 as well. These keys

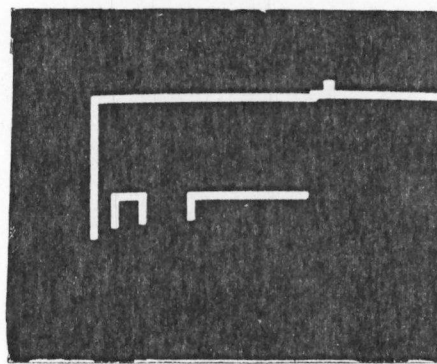
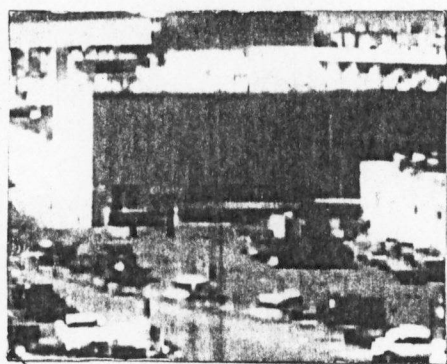
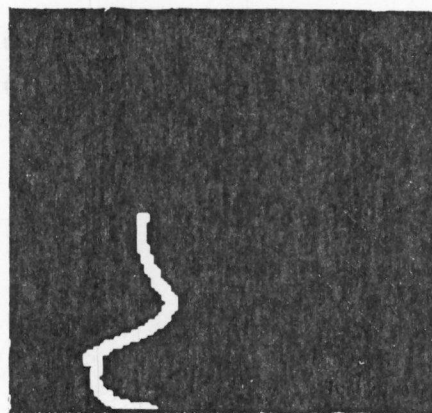


Figure 26. Additional test scenes.

depict the more common types of close-in and medium distance edges encountered in image processing.

The grades obtained for the Kodak girl, which is a frequently employed television monitor and color display registration standard, are in Table 10. A threshold level of 10% was used to create scores corresponding to those of Table 4 (page 65). From these results, it is apparent that the range operator did the best job, at the 10% threshold level, of locating the key edge.

The third test scene is of a typical commercial building, with the key edge being made up of the building's outline, and the boundary of the door and windows. Table 11 contains the grades for this scene, again with the threshold level set at 10%. For this scene, the highest grade was obtained by the Robert's gradient operator.

Overall average grades are shown in Table 12. Based on composite scores, Robert's square root gradient, at a basic threshold level of 10%, exhibited superior performance in detecting the key edges, while including a minimum of extraneous points. Consequently, it would be a justifiable first choice as a edge detector under conditions corresponding to those assumed here.

Table 10

Absolute Grade of Operator Test Set
"Kodak Girl"

OPERATOR	R_c	R	R_d
Sobel	16.979	59.637	33.924
Robert's	21.191	66.742	44.855
Robert's Sq.	21.715	69.444	47.909
Gradient	21.734	69.208	47.222
Range	22.106	68.213	47.264
Frei-Chen	19.216	63.599	39.888

Table 11

Absolute Grade of Operator Test Set
Building Scene

OPERATOR	R_c	R	R_d
Sobel	12.824	45.512	15.607
Robert's	15.037	49.367	20.144
Robert's Sq.	14.424	48.139	18.768
Gradient	14.499	48.522	19.082
Range	13.903	47.400	17.802
Frei-Chen	9.265	38.503	8.353

Table 12

Average Absolute Grade of
Operator Test Set

OPERATOR	R_c	R	R_d
Sobel	16.750	57.022	31.027
Robert's	18.699	60.960	36.851
Robert's Sq.	19.255	61.973	38.838
Gradient	18.962	61.289	37.358
Range	18.874	60.906	36.931
Frei-Chen	16.566	56.473	31.365

CHAPTER 6

CONCLUSIONS

A general introduction to edge detection and basic edge detection systems has been presented. Additionally, the fundamentals of thresholding have been described, along with several means of improving and enhancing thresholded imagery. To evaluate these varied techniques, two different grading schemes have been developed, along with associated processes, in order to enable the evaluation of edge operators. Examples of the implementation and function of both these grading systems have been shown and discussed so that their operation could be demonstrated and documented.

This study has exposed an area of tremendous utility and value to image processing. By no means has the ultimate absolute grading system been formulated. Nonetheless, two tools have been developed which allow evaluation of any operator on any data set, as well as providing numerical measurements as feedback on edge detection schemes. These grading approaches should prove to be quite valuable to those selecting an operator for a particular task, designing and building new operators, or identifying the critical edges in any scene.

LIST OF REFERENCES

LIST OF REFERENCES

1. G. S. Robinson, "Detection and Coding of Edges Using Directional Masks", Optical Engineering, Vol. 16, No. 6, November-December, 1977.
2. L. S. Davis, "A Survey of Edge Detection Techniques", Computer Graphics and Image Processing, Vol. 4, No. 3, p. 248-270, (1975).
3. R. C. Gonzalez and P. Wintz, Digital Image Processing, Reading, Mass.: Addison-Wesley, 1977.
4. L. G. Roberts, "Machine Perception of three-dimensional solids", J. T. Tippett et. al., eds. Cambridge, Mass.: MIT Press, 1965.
5. R. O. Duda and P. E. Hart, Pattern Classification and Scene Analysis, New York: Wiley, 1971.
6. D. L. Davies, "Edge Detection in Digital Images Using Small Mask and Vector Operators and the Method of Polar Histograms", Unpublished Master's thesis, The University of Tennessee, Knoxville, 1978.
7. W. Frei and C-C Chen, "Fast Boundary Detection: A Generalization and a New Algorithm", IEEE Transactions on Computers, Vol. C-26, No. 10, October 1977.
8. G. S. Robinson, "Edge Detection by Compass Gradient Masks", Computer Graphics and Image Processing, Vol. 6, No. 5, October 1977.
9. J. M. Wozencraft and I. M. Jacobs, Principles of Communication Engineering, New York: Wiley and Sons, 1965.
10. W. K. Pratt, Digital Image Processing, New York: Wiley and Sons, 1978.
11. E. L. Hall, Computer Image Processing and Recognition, New York: Academic Press (in preparation).
12. J. R. Fram and E. S. Deutsch, "On the Quantitative Evaluation of Edge Detection Schemes and Their Comparison with Human Performance", IEEE Transactions on Computers, Vol. C-24, No. 6, June, 1975.
13. A. Rosenfeld and M. Thurston, "Edge and Curve Detection for Visual Scene Analysis", IEEE Transactions on Computers, Vol. C-20, No. 5, May 1971.
14. N. Nilsson, Learning Machines, New York: McGraw-Hill, 1965.

15. J. K. Bowker, "Edge Vector Image Analysis", IEEE Proceedings of the Second International Joint Conference on Pattern Recognition, August 1974, Copenhagen, Denmark.
16. R. Nevatia, "Locating Object Boundaries in Textured Environments", IEEE Transactions on Computers, Vol. 25, No. 11, November 1976.

APPENDIX

IMPLEMENTING THE DAVIES EDGE VECTOR SYSTEM

The Davies edge vector system [6] is an algorithm for determining magnitude and direction of an edge. The magnitude measurement is equivalent to that made by the range or maximum difference operator (see Chapter 2). The angle determination is a sophisticated measurement technique based upon the physics of sensor operation and quantization.

The theoretical basis is quite straightforward. Consider a two-by-two neighborhood of (x,y) . If a simple, linear edge passes through this neighborhood, the maximum and minimum pixels will be diagonally opposite each other. The edge need not pass through the groups center for this to be true.

A rough approximation for the edge's orientation is the amount of rotation needed to position the maximum-minimum diagonal pair in an upper left-lower right arrangement, respectively, as shown in Figure 1.

G_1 (max)	G_2
G_4	G_3 (min)

Figure A-1. Alignment of pixels and labelling scheme of Davies method, after necessary rotation.

Added to the coarse estimation of edge angle obtained from the rotation of the neighborhood, is a fine measurement, α :

$$\alpha = \frac{\pi}{2} \left[\frac{G_1 G_2 - G_3 G_4}{(G_1 - G_3)(G_2 + G_4)} \right]$$

The magnitude of the edge is given by $E = G_1 - G_3$.

Special cases exist, depending upon the number of maximum values in the neighborhood. If there is one maximum, but the minimum value is not diagonally opposite, no edge exists and the magnitude is set to zero.

If there are two maxima, the one which is opposite the minimum value is chosen to be G_1 . Should there be two minima as well, the angle measurement becomes trivial.

Example 1: Assume $G_1 = G_2 = G_a$ and $G_3 = G_4 = G_b$

$$\begin{aligned} \alpha &= \frac{\pi}{2} \left[\frac{G_a^2 - G_b^2}{(G_a - G_b)(G_a + G_b)} \right] \\ &= \frac{\pi}{2} \left[\frac{G_a^2 - G_b^2}{G_a^2 - G_b^2} \right] \\ &= \frac{\pi}{2} \end{aligned}$$

Example 2: Assume $G_1 = G_4 = G_a$, $G_2 = G_3 = G_b$

$$\begin{aligned} \alpha &= \frac{\pi}{2} \left[\frac{G_a G_b - G_b G_a}{(G_a - G_b)(G_b + G_a)} \right] \\ &= 0 \end{aligned}$$

In either case, no additional adjustment need be made over the two maxima one minimum case.

Should there be three maxima, the one diagonally opposite the fourth value is chosen as G_1 . In this case, the edge is obviously off center, and has $\alpha = 45^\circ$.

Example 3: $G_1 = G_2 = G_4 = G_a, G_3 = G_b$

$$\alpha = \frac{\pi}{2} \left[\frac{G_a^2 - G_a G_b}{(G_a - G_b)(G_a + G_a)} \right]$$

$$\alpha = \frac{\pi}{2} \left[\frac{G_a(G_a - G_b)}{2G_a(G_a - G_b)} \right]$$

$$= \frac{\pi}{4} \text{ or } 45^\circ$$

Four maxima indicate a uniform area with no edge.

Some pixel arrangements cannot be caused by a simple linear edge. Such a configuration is one in which there are two maxima, diagonally opposite each other. Only two closely spaced parallel edges, or an edge with an extremely small radius of curvature could produce this type pattern. In either case, the edge magnitude result is set to zero.

The program DAVIES which implements the Davies system must check for these special cases before computing the magnitude and angle for a neighborhood. This prevents divide-by-zero errors and other computational inconsistencies.

There are several important details in the DAVIES program. The sort is a bubble sort, both of G , the array containing the pixel values, and

of IORD, an array which contains the original locations in the neighborhood. Upon completion of the sort and tally of the maxima, the first value of IORD can be used to compute the amount of 90° rotations necessary to bring the maximum gray level into position G_1 . Should additional rotations be necessary, such as could be required by the two and three maxima cases, the fourth value of IORD is used as a pointer to the minimum value G_4 . For these additional rotations, array IROT is used to determine the point diagonally opposite IORD(4). This is done by reading IROT(IORD(4)+2). IROT is essentially a rough modulo-4 adder with no carry.

The process of evaluating the accuracy of each operator using a grading approach requires the comparison of the operator's output to the rest of the operator set, thus requiring knowledge of every operator's decision on each point in the image.

From a programming standpoint, the most expedient comparison scheme would be to have the detector data always available and, when considering a certain image point, examine that point in each and every output. This puts an unreasonable demand on the computer communication channels and involves a large amount of slow I/O operations.

Encoding the binary "yes-no" decision of each operator as a 1 or 0 would permit many operator output images to be compressed into a small amount of space. The series of ones and zeroes that represent the decisions of the operators on a particular point can be assembled into a single word stored at the exact same location in a special picture.

Relative grading can be carried out on just this single coded image, with comparisons and counting being done on a bit-versus-bit basis.

The two programs necessary to perform this version of relative grading are CODVOT, which handles the data encoding, and GRADE 2 which calculates the grade and various point-counts.

```

C      DAVIES FOR LOGICAL, LOGICAL, LOGICAL
C
C      THIS PROGRAM USES THE EDGE MAGNITUDE/ANGLE DETECTOR
C      DEVELOPED BY DAVE DAVIES. IT PRODUCES TWO OUTPUT
C      IMAGES: ONE OF THE EDGE MAGNITUDE, THE OTHER EDGE
C      ANGLE.
C
C      (THE ANGLE IS RESCALED FROM 0=>2*PI TO 0=>255)
C
C      AUTHOR: DAVID J. BRYANT
C      SEPT 1978
C
0001      INTEGER*2 OLPI1
0002      DIMENSION GRID(4), G(4), IORD(4), IROT(7)
0003      DATA IROT/1, 2, 3, 4, 1, 2, 3/
0004      PI2=3.14159/2.0
0005      NDIM=OLPI1()
0006      IF(NDIM.NE.NWPI2()) STOP 'MISMATCHED OUTPUT'
0008      IF(NDIM.NE.NWPI3()) STOP 'MISMATCHED OUTPUT #2'
0010      DO 1000 I=1, NDIM
0011      DO 1000 J=1, NDIM
C
C      CHECK FOR AND ZERO THE RIGHT EDGE AND BOTTOM POINTS
C
0012      IF(I.NE.NDIM.AND.J.NE.NDIM) GO TO 4
0014      CALL IWPI2(I, J, 0)
0015      CALL IWPI3(I, J, 0)
0016      GO TO 1000
C
C      ESTABLISH THE 2X2 NEIGHBORHOOD IN ARRAY 'GRID'
C
0017      4      GRID(1)=FLOAT(IRPI1(I, J))
0018      GRID(2)=FLOAT(IRPI1(I, J+1))
0019      GRID(3)=FLOAT(IRPI1(I+1, J+1))
0020      GRID(4)=FLOAT(IRPI1(I+1, J))
0021      DO 15 II=1, 4
0022      IORD(II)=II
0023      G(II)=GRID(II)
0024      15      CONTINUE
C
C      SORT THE NEIGHBORHOOD
C
0025      18      IFLAG=0
0026      DO 20 II=1, 3
0027      IF(G(II).GE.G(II+1)) GO TO 20
0029      IT=IORD(II)
0030      IORD(II)=IORD(II+1)
0031      IORD(II+1)=IT
0032      T=G(II)
0033      G(II)=G(II+1)
0034      G(II+1)=T
0035      IFLAG=1
0036      20      CONTINUE
0037      IF(IFLAG.EQ.1) GO TO 18

```

```

C
C CHECK THE NUMBER OF MAXIMUMS AND BRANCH ACCORDINGLY
C
0039 IF (GRID(IORD(1)).NE. GRID(IORD(2))) GO TO 100
0041 IF (GRID(IORD(2)).NE. GRID(IORD(3))) GO TO 200
0043 IF (GRID(IORD(3)).NE. GRID(IORD(4))) GO TO 300
C
C FOUR MAXIMUMS---NO EDGE PRESENT
C
0045 50 IMAG=0
0046 ANG=0.0
0047 GO TO 450
C
C ONE MAXIMUM
C
0048 100 IF (MOD((IORD(1)+IORD(2)),2).EQ.0) GO TO 50
0050 IIN=IORD(1)-1
C
C GENERALIZED COMPUTATION OF (MAG,ANG) BASED ON THE
C CORRECT ARRANGEMENT OF THE NEIGHBORHOOD
C
0051 125 DO 150 II=1,4
0052 G(II)=GRID(IROT(IIN+II))
0053 150 CONTINUE
0054 IF (G(2).EQ. G(3).AND. G(3).EQ. G(4)) GO TO 50
0056 IF ((G(2)+G(4)).EQ. 0) GO TO 50
0058 ALPHA=((G(3)*G(4)-G(1)*G(2))/((G(3)-G(1))*(G(2)+G(4))))
0059 ANG=(ALPHA+FLUAT(IIN))*PI2
0060 IMAG=IFIX(.5+G(1)-G(3))
0061 GO TO 450
C
C TWO MAXIMUM AND THREE MAXIMUM CASES ARE QUITE
C SIMILAR AND CAN BE HANDLED IN THE SAME SECTION
C
0062 200 IF (MOD((IORD(1)+IORD(2)),2).EQ.0) GO TO 50
0064 300 IIN=IROT(IORD(4)+2)-1
0065 GO TO 125
C
C OUTPUT SECTION---WRITE (MAG,ANG) FOR EACH POINT
C
0066 450 IANG=IFIX(.5+255.*ANG/6.28318)
0067 CALL IWPI2(I,J,IMAG)
0068 CALL IWPI3(I,J,IANG)
0069 1000 CONTINUE
0070 CALL CLPI1
0071 CALL CLPI2
0072 CALL CLPI3
0073 CALL ITTOUR("007")
0074 STOP 'END OF PROCESSING'
0075 END

```

```

C      COBVOT. FOR LOGICAL<LOGICAL, LOGICAL
C
C      THIS PROGRAM CONVERTS THE RESULTS OF AN EDGE OPERATOR
C      TO A BINARY CODED VOTE AND ADDS IT TO THE STRING OF
C      VOTES ALREADY PROCESSED.
C
C      AUTHOR: DAVID J. BRYANT
C      SEPT 1978
C
0001      INTEGER*2 OLPI1, OLPI2
0002      TYPE 5
0003      5      FORMAT(/ LP11 IS THE CURRENT CODED VOTE,
* LP12 IS THE NEW VOTER'//)
C
C      OPEN NEW AND OLD FILES AND SET THINGS UP
C
0004      NDIM1=OLPI1()
0005      NDIM2=OLPI2()
0006      IF(NDIM1 NE. NDIM2) STOP 'MISMATCHED INPUT'
0008      IF(NDIM1 NE. NUPIS()) STOP 'MISMATCHED OUTPUT'
0010      NVOT=IRPI1(1, 1)
0011      IADD=IFIX(2. **NVOT)
0012      TYPE 10
0013      10      FORMAT(/ INPUT THE VOTE-FOR THRESHOLD =>', $)
0014      ACCEPT 15, ITHR
0015      15      FORMAT(I3)
0016      TYPE 20, IADD, NVOT, ITHR
0017      20      FORMAT(1X, I3, /='2**', I3, / ITHR=', I3)
C
C      LOOP THROUGH INPUT PICTURES
C
0018      DO 100 I=1, NDIM1
0019      DO 100 J=1, NDIM1
0020          CALL IWPIS(I, J, 0)
0021          IF(I.EQ. 1. AND. J.EQ. 1) GO TO 100
0023          IVAL2=IRPI2(I, J)
C
C      IS NEW INPUT A VOTE FOR??
C
0024          IF(IVAL2.LT. ITHR) GO TO 90
C
C      YEP, SO LETS CODE IT IN...
C
0026          IVAL1=IRPI1(I, J)
0027          IVAL3=IVAL1+IADD
0028          CALL IWPIS(I, J, IVAL3)
0029          GO TO 100
C
C      THIS PROCESSES A NO VOTE BY REWRITING THE OLD CODED DATA
C
0030      90          IVAL1=IRPI1(I, J)
0031          CALL IWPIS(I, J, IVAL1)
0032      100      CONTINUE
0033      NVOT=NVOT+1

```

```
0034      CALL IWF13(1,1,NVOT)
0035      CALL CLP11
0036      CALL CLP12
0037      CALL CLP13
0038      STOP
0039      END
```

```

C      GRADE2. FOR  LOGICAL
C
C      THIS PROGRAM COMPUTES THE GRADE FOR EACH OPERATOR BY
C      COMPARING ITS ANALYSIS TO THE CONSENSUS OPINIONS OF THE
C      OTHER OPERATORS.  THE INPUT IS THE OUTPUT FILE OF CODVOT,
C      THE VOTE CODING PROGRAM.
C
C      AUTHOR: DAVID J. BRYANT
C      SEPT 1978
C
0001      DIMENSION IBIN(8),CNT(2,8)
0002      INTEGER*2 OLPI1
0003      DATA CNT/16*0.0/
0004      NDIM=OLPI1()
0005      IF(NDIM.NE.NWPI2()) STOP'MISMATCHED OUTPUT'
0006      TYPE 2
0007      FORMAT(' INPUT THE "NO" VOTE CEILING=>',%)
0008      2   ACCEPT 3, IZRO
0009      3   FORMAT(I1)
0010      NVOT=IRPI1(1,1)
0011      IECNT=0
0012      DO 1000 I=1,NDIM
0013      DO 1000 J=1,NDIM
0014      CALL IWPI2(I,J,0)
0015      IF(I.EQ.1.AND.J.EQ.1) GO TO 1000
0016      IVAL=IRPI1(I,J)
0017      DO 5 II=1,8
0018      IBIN(II)=0
0019      5   CONTINUE
0020      15  DECODE IVAL INTO BINARY
0021      DO 10 II=NVOT,1,-1
0022      NUM=IFIX(2.**((II-1)))
0023      IF(IVAL.GE.NUM) IBIN(II)=1
0024      IVAL=IVAL-NUM*IBIN(II)
0025      10  CONTINUE
0026      17  CHECK FOR ZERO'S IN THE BINARY FORM
0027      IC=0
0028      DO 15 II=1,NVOT
0029      IF(IBIN(II).NE.0) GO TO 15
0030      IC=IC+1
0031      15  CONTINUE
0032      IF(IC.EQ.IZRO) GO TO 1000
0033      IF(IC.GT.IZRO) GO TO 17
0034      IPOINT=1
0035      IECNT=IECNT+1
0036      CALL IWPI2(I,J,255)
0037      GO TO 19
0038      17  IPOINT=0
0039      17  INCREMENT PROPER COUNTERS

```

```

C
0043 19 DO 20 II=1,NVOT
0044 IF (IBIN(II).EQ. IPOINT) CNT(IPOINT+1,II)=
* CNT(IPOINT+1,II)+1.
0046 20 CONTINUE
0047 1000 CONTINUE
C
C COMPUTE AND PRINT GRADES
C
0048 TYPE 25
0049 25 FORMAT(' DPR #',3X,'EDGES',4X,'SCORE',3X,'BKGRD',3X,'SCORE',
*4X,'TOTAL',5X,'NET %',/,
* '*****')
0050 PER=100./ (FLOAT (NDIM)**2-1.)
0051 DO 30 I=1,NVOT
0052 SCOREB=CNT(1,I)*100./ (FLOAT (NDIM)**2-1.-FLOAT (IECNT))
0053 SCOREE=CNT(2,I)*100./FLOAT (IECNT)
0054 ICNTB=IFIX(.5+CNT(1,I))
0055 ICNTE=IFIX(.5+CNT(2,I))
0056 ICOUNT=ICNTE+ICNTB
0057 SCORT=FLOAT (ICOUNT)*PER
0058 TYPE 35, I, ICNTE, SCOREE, ICNTB, SCOREB, ICOUNT, SCORT
0059 35 FORMAT(3X, I1, 5X, I5, 3X, F7.3, 3X, I5, 3X, F7.3, 3X, I5, 3X, F7.3)
0060 30 CONTINUE
0061 TYPE 40
0062 40 FORMAT(' *****',
* '*****', //)
0063 CALL CLPI1
0064 CALL CLPI2
0065 CALL ITTOUR('007')
0066 STOP 'GRADING COMPLETE'
0067 END

```

```

C      KEYBLD.FOR  LOGICALC
C
C      THIS PROGRAM ALLOWS THE USER TO CONSTRUCT A KEY TO BE USED
C      IN ABSOLUTE GRADING.  PROVIDED WITH THE STARTING ROW
C      OF THE KEY, IT PROMPTS THE USER FOR THE STARTING POINT IN
C      THE ROW AND THE NUMBER OF POINTS TO BE IN THE KEY IN THAT
C      PARTICULAR ROW.  AFTER OBTAINING THIS INFORMATION
C      IT MOVES ON TO THE NEXT ROW.
C
C      ONLY SIMPLE, CONTINUOUS, FUNCTION KEYS MAY BE BUILT
C      WITH THIS PROGRAM.  THAT IS, THERE MUST BE AT LEAST ONE
C      POINT IN EACH ROW BETWEEN THE STARTING AND ENDING
C      POINT.  THE ELEMENTS IN A ROW MUST BE ADJACENT. (NO BLANK
C      SPACES ALLOWED)
C
C      TO EDIT THE KEY PRODUCED, OR TO ADD FANCY STUFF, USE
C      THE KEY EDITOR EDKEY.FOR.
C
C      AUTHOR: DAVID J. BRYANT
C      OCT 1978
C
0001      NDIM=NMPI1()
0002      TYPE 2
0003      2      FORMAT(/ YOU WILL BE PROMPTED BY THE ROW #.  ENTER THE//
*      STARTING COLUMN AND THE # OF POINTS TO BE ENTERED (I3,I3)//
*      ENTER 0,0 TO EXIT//)
C
C      ZERO THE OUTPUT PICTURE
C
0004      DO 3 I=1,NDIM
0005      DO 3 J=1,NDIM
0006          CALL INPI1(I,J,0)
0007      3      CONTINUE
C
C      GET STARTING ROW FROM USER
C
0008      TYPE 5
0009      5      FORMAT(/ ENTER STARTING ROW=>/,*)
0010      ACCEPT 10,IROW
0011      10     FORMAT(I3)
C
C      HERE BEGINS THE LOOP FOR BUILDING EACH ROW.  EACH
C      INPUT MUST BE ECHOED TO LET USER KNOW WHAT HE IS DOING.
C      THEY MUST ALSO BE TESTED TO SEE IF THEY ARE IN BOUNDS
C      OR IF AN EXIT (ICOL=0) IS DESIRED.
C
0012      15     TYPE 20,IROW
0013      20     FORMAT(/ ROW #/,I3,/=>/,*)
0014      ACCEPT 22,ICOL,INUM
0015      22     FORMAT(2I3)
0016      TYPE 23,ICOL,INUM
0017      23     FORMAT(2I5)
0018      IF (ICOL.EQ.0.OR.ICOL.GT.NDIM) GO TO 1000
0020      DO 30 I=ICOL,(ICOL+INUM-1)

```

```
0021          IF(I.GT.NDIM) GO TO 30
0023          CALL IWPI1(IROW,I,255)
0024  30      CONTINUE
0025          IROW=IROW+1
0026          IF(IROW.GE.NDIM) GO TO 1000
0028          GO TO 15
0029  1000    CALL CLPI1
0030          STOP
0031          END
```

```

C      EDKEY.FOR    LOGICAL<LOGICAL
C
C      THIS PROGRAM IS CAPABLE OF MAKING CHANGES IN THE KEY
C      PRODUCED BY KEYELD.FOR.  IT SCANS THROUGH THE KEY, TELLING
C      THE USER WHAT POINTS ARE SET AND WHERE THEY ARE.  IF
C      ANY ARE IN ERROR, OR IF SOME MODIFICATIONS ARE NEEDED, THE
C      USER INFORMS THE PROGRAM, AND AN EDIT MODE IS ENTERED.
C
C      IN THE EDIT MODE, THE ROW CAN BE REPLACED BY ANY
C      NEW SET OF POINTS, OR ADDITIONAL POINTS CAN BE INSERTED.
C
C      AUTHOR: DAVID J. BRYANT
C      OCT 1978
C
0001      INTEGER*2 OLPI1
0002      NDIM=OLPI1()
C
C      DUE TO THE UNUSUAL NATURE OF THE LINE MANIPULATIONS,
C      A DO-LOOP IS NOT PRACTICAL.  HENCE A COUNTER SYSTEM IS
C      USED, WITH I AND J BEING THE COUNTERS
C
0003      I=1
0004      J=1
C
C      HERE BEGINS THE LOOP.  IF A ZERO POINT IS READ I AND
C      J ARE INCREMENTED TO THE NEXT POINT.  IF (I,J) IS NON-ZERO
C      A SEARCH IS CONDUCTED TO DETERMINE HOW MANY POINTS ARE CONNECTED
C      TO (I,J)  THE LENGTH OF THE STRING AND ITS STARTING POINT
C      IS OUTPUT FOR USER INSPECTION.
C
0005 30      IVAL=IRPI1(I,J)
0006      IF(IVAL.EQ.0) GO TO 1000
0008      IC=1
0009 99      IVAL=IRPI1(I,(J+IC))
0010      IF(IVAL.EQ.0) GO TO 100
0012      IC=IC+1
0013      GO TO 99
0014 100     TYPE 101,I,J,IC
0015 101     FORMAT(' AT POINT (',I3,',',I3,',') LIES A CHAIN OF LENGTH',I3)
C
C      AT THIS POINT, THE USER DECIDES IF ALL IS O.K. OF IF CHANGES
C      ARE CALLED FOR.
C
0016      TYPE 102
0017 102     FORMAT(' O.K. ?? (1=NO)=>',%)
0018      ACCEPT 103,IANS
0019 103     FORMAT(I1)
0020      IF(IANS.NE.1) GO TO 2000
0022      TYPE 104
0023 104     FORMAT(' INPUT THE NEW STARTING POINT AND LENGTH=>',%)
0024      ACCEPT 105,ICOL,INUM
0025 105     FORMAT(2I3)
0026      TYPE 106
0027 106     FORMAT(' ERASE THE OLD STRING? (1=NO)=>',%)

```

```
0028      ACCEPT 103, IANS
0029      IF(IANS.EQ.1) GO TO 111
0031      DO 110 II=J, J+IC-1
0032          CALL IWPI1(I, II, 0)
0033 110    CONTINUE
0034 111    DO 115 II=ICOL, ICOL+INUM-1
0035          CALL IWPI1(I, II, 255)
0036 115    CONTINUE
      C
      C      THIS IS THE INDEX HANDLING SECTION.  IT KEEPS UP
      C      WITH I AND J AND DOES WHAT NEEDS TO BE DONE
      C      AS CALLED FOR BY THE PROGRAM AND THE USER.
      C
0037 1000  J=J+1
0038      IF(J.GT.NDIM) GO TO 2000
0040      GO TO 30
0041 2000  I=I+1
0042      J=0
0043      IF(I.GT.NDIM) GO TO 2500
0045      GO TO 30
0046 2500  CALL CLPI1
0047      STOP
0048      END
```

```

C      ABSGRD. FOR <LOGICAL, LOGICAL
C
C      THIS PROGRAM COMPARES THE BINARY OUTPUT OF A
C      DETECTOR/THRESHOLDING DUET AGAINST A BINARY TARGET
C      OR KEY.  THE GRADE IS CALCULATED FROM:
C
C       $R = 100\% * (1 - NB / (N - NK)) * \text{SORT}(NC / NK)$ 
C
C      WHERE:
C      N= # OF POINTS IN IMAGE
C      NK=# OF POINTS IN KEY
C      NC=# CORRECT (INSIDE KEY AREA) EDGE POINTS
C      NB=# INCORRECT (OUTSIDE KEY AREA) EDGE POINTS
C
C      AUTHOR: DAVID J. BRYANT
C      OCT 1978
C
0001      INTEGER*2 OLPI1, OLPI2
0002      TYPE 5
0003 5      FORMAT(' LP11 IS THE TARGET/KEY, LP12 IS OPERATOR OUTPUT'//)
0004      NDIM=OLPI1()
0005      NDIM1=OLPI2()
0006      IF(NDIM.NE.NDIM1) STOP'MISMATCHED INPUT'
0008      NB=0
0009      NC=0
0010      NK=0
0011      DO 1000 I=1,NDIM
0012      DO 1000 J=1,NDIM
0013          IVAL1=IRP11(I,J)
0014          IVAL2=IRP12(I,J)
C
C      COUNTERS NB,NC AND NK ARE INCREMENTED IF A NON-ZERO POINT
C      LIES IN THEIR AREA OF JURISDICTION.
C
0015          IF(IVAL1.NE.0) GO TO 10
0017          IF(IVAL2.NE.0) NB=NB+1
0019          GO TO 1000
0020 10      NK=NK+1
0021          IF(IVAL2.NE.0) NC=NC+1
0023 1000    CONTINUE
C
C      THE TERMS (T1 AND T2) OF THE GRADING EQUATION ARE COMPUTED.
C      IF T1 IS GREATER THAN OR EQUAL TO T2, THE GRADE IS FORCED
C      TO ZERO TO INDICATE FAILURE TO LOCATE THE KEY.
C
0024      T1=FLOAT(NB)/(FLOAT(NDIM)**2-FLOAT(NK))
0025      T2=FLOAT(NC)/FLOAT(NK)
0026      IF(T1.GE.T2) GO TO 999
0028      R=100. *(1.-T1)*SORT(T2)
0029      RC=100. *SORT(FLOAT(NK)/FLOAT(NC+NB))*T2
0030      RD=100. *(T2-T1)
0031      GO TO 99
0032 999      R=0.
0033 99      TYPE 100, NC, NK, NB, RC, R, RD

```

```
0034 100  FORMAT(' NC=',I5,' OF NK=',I5,' WITH NB=',I5,' /  
          *' CORRELATOR GRADE =',F6.3,' %', /  
          *' FRACTIONAL GRADE =',F6.3,' %', /  
          *' DIFFERENCE GRADE =',F6.3,' %')  
0035      CALL CLPI1  
0036      CALL CLPI2  
0037      STOP 'END ABSOLUTE GRADING'  
0038      END
```

```

C      PICK FOR LOGICAL LOGICAL
C
C      THIS PROGRAM SCANS THROUGH A BINARY PICTURE AND
C      ELIMINATES THOSE POINTS WHICH ARE ISOLATED. THAT IS,
C      A 3-BY-3 NEIGHBORHOOD OF (X,Y) IS EXAMINED AND, IF
C      (X,Y) IS THE ONLY NON-ZERO POINT PRESENT, IT IS ZEROED.
C
C      THIS IS A SIMPLE FORM OF ASSOCIATION, AND CAN BE USED
C      IN EFFORTS TO IMPROVE THE PERFORMANCE OF AN OPERATOR-
C      THRESHOLDING SYSTEM.
C
C      AUTHOR: DAVID J. BRYANT
C      OCT 1978
C
0001      INTEGER*2 OLPI1
0002      NDIM=OLPI1()
0003      IF(NDIM.NE.NMPI2()) STOP 'MISMATCHED OUTPUT'
C
C      ZERO THE OUTPUT IMAGE TO START WITH
C
0005      DO 5 I=1,NDIM
0006      DO 5 J=1,NDIM
0007          CALL IWPI2(I,J,0)
0008      5  CONTINUE
0009      DO 1000 I=2,NDIM-1
0010      DO 1000 J=2,NDIM-1
0011          IVAL=IRPI1(I,J)
0012          IF(IVAL.EQ.0) GO TO 1000
C
C      THIS SECTION READS THE 3-BY-3 NEIGHBORHOOD AND
C      CHECKS ALL POINTS EXCEPT (I,J) TO SEE IF THEY ARE ZERO.
C
0014      DO 10 II=1,3
0015      DO 10 JJ=1,3
0016          IF(II.EQ.2.AND.JJ.EQ.2) GO TO 10
0017          IT=II+I-2
0018          JT=JJ+J-2
0019          IVAL=IRPI1(IT,JT)
0020          IF(IVAL.NE.0) GO TO 20
0021      10  CONTINUE
0022      GO TO 1000
C
C      WRITE THE POINT TO THE OUTPUT IF IT IS NOT ISOLATED
C
0025      20  CALL IWPI2(I,J,IVAL)
0026      1000 CONTINUE
0027      CALL CLPI1
0028      CALL CLPI2
0029      CALL ITTOUR('007')
0030      STOP 'NOISE REDUCTION COMPLETE'
0031      END

```

```

C      THICK FOR LOGICAL LOGICAL
C
C      THIS PROGRAM CREATES A THICKENED BUFFER ZONE
C      AROUND THE EDGES IN A BINARY IMAGE. IT DOES
C      SO BY EXAMINING A 3-BY-3 NEIGHBORHOOD OF (X,Y).
C      IF A NON-ZERO POINT EXISTS WITHIN THE NEIGH-
C      BORHOOD, (X,Y) IS SET TO 255.
C
C      THIS PROGRAM IS USED TO PREPARE AN INPUT MASK
C      FOR THE LINKING OPERATION. THE OUTPUT IS TO
C      BE LOGICALLY "AND-ED" WITH THE OPERATOR'S
C      OUTPUT TO CREATE THE INPUT TO LINKER FOR.
C
C      AUTHOR: DAVID J. BRYANT
C      NOV 1978
C
0001      INTEGER*2 OLPI1
0002      NDIM=OLPI1()
0003      IF(NDIM.NE.NWPI2()) STOP 'MISMATCHED OUTPUT'
C
C      ZERO THE OUTPUT IMAGE TO START WITH
C
0005      DO 5 I=1,NDIM
0006      DO 5 J=1,NDIM
0007          CALL IWPI2(I,J,0)
0008      5      CONTINUE
0009      DO 1000 I=2,NDIM-1
0010      DO 1000 J=2,NDIM-1
0011          IVALA=IRPI1(I,J)
C
C      CHECK THE VALUE OF THE POINT. IF IT IS
C      NON-ZERO, THERE IS NO NEED TO PROCESS
C      IT.
C
0012          IF(IVALA.EQ.0) GO TO 7
0014          CALL IWPI2(I,J,255)
0015          GO TO 1000
C
C      THIS SECTION READS THE 3-BY-3 NEIGHBORHOOD AND
C      CHECKS ALL POINTS EXCEPT (I,J) TO SEE IF THEY ARE ZERO.
C
0016      7      DO 10 II=1,3
0017      DO 10 JJ=1,3
0018          IF(II.EQ.2.AND.JJ.EQ.2) GO TO 10
0020          IT=II+I-2
0021          JT=JJ+J-2
0022          IVAL=IRPI1(IT,JT)
0023          IF(IVAL.NE.0) GO TO 20
0025      10      CONTINUE
0026          GO TO 1000
C
C      WRITE OUT THE POINT
C
0027      20      CALL IWPI2(I,J,255)

```

```
0028 1000 CONTINUE
0029      CALL CLPI1
0030      CALL CLPI2
0031      CALL ITTOUR("007)
0032      STOP 'THICKENING COMPLETE'
0033      END
```

```

C      ASSOC. FOR LOGICAL<LOGICAL, LOGICAL
C
C      THIS PROGRAM TAKES THE MAGNITUDE AND ANGLE FILES
C      GENERATED BY THE DAVIES OPERATOR (DAVIES. FOR) AND
C      ASSOCIATES EDGES WHICH ARE OF SUFFICIENT MAGNITUDE
C      AND ANGLE SIMILARITY. IT USES THE ASSOCIATION CRITERION
C      OUTLINED BY J. KENT BOWKER (SEE HIS PAPER), NAMELY
C
C      IS  $E2 * (\cos(A2 - A1))$  GREATER THAN  $E1$  ?
C
C      FOR A 3 BY 3 NEIGHBORHOOD OF (E1, A1).
C
C      AUTHOR: DAVID J. BRYANT
C      OCT 1978
C
0001  DIMENSION INABOR(4,6), IMAGN(9), IANGN(9)
0002  INTEGER*2 OLPI1, OLPI2
0003  DATA INABOR/1,2,3,7,8,9,1,2,4,6,8,9,1,4,7,3,6,9,2,3,6,4,7,8/
0004  RTD=180./3.14159
0005  NDIM1=OLPI1()
0006  NDIM2=OLPI2()
0007  IF(NDIM1.NE.NDIM2) STOP 'MISMATCHED INPUT'
0009  IF(NDIM1.NE.NWPI3()) STOP 'MISMATCHED OUTPUT'
0011  DO 2 I=1,NDIM1
0012      CALL IWPI3(1,I,0)
0013      CALL IWPI3(NDIM1,I,0)
0014      CALL IWPI3(1,1,0)
0015      CALL IWPI3(I,NDIM1,0)
0016  2  CONTINUE
0017  DO 1000 I=2,NDIM1-1
0018  DO 1000 J=2,NDIM1-1
0019      IVAL=IRPI1(I,J)
0020      E1=FLOAT(IVAL)
0021      IF(E1.EQ.0) GO TO 999
0023      A1=FLOAT(IRPI2(I,J))
0024      ANG1=RTD*6.28318*A1/255.
0025      IF(ANG1.GE.157.5) ANG1=ANG1-180.
0027      IF(ANG1.GE.157.5) ANG1=ANG1-180.
0029      IF(ANG1.GE.-22.5) INER=1
0031      IF(ANG1.GE.22.5) INER=2
0033      IF(ANG1.GE.67.5) INER=3
0035      IF(ANG1.GE.112.5) INER=4
0037      DO 10 II=I-1,I+1
0038      DO 10 JJ=J-1,J+1
0039          IMAGN((II-1+1)*3+(JJ-J+2))=IRPI1(II,JJ)
0040          IANGN((II-1+1)*3+(JJ-J+2))=IRPI2(II,JJ)
0041  10  CONTINUE
0042      DO 20 II=1,6
0043          E2=FLOAT(IMAGN(INABOR(INER,II)))
0044          A2=FLOAT(IANGN(INABOR(INER,II)))
0045          E2P=E2*COS(6.28318*(A2-A1)/255.)
0046          IF(E2P.GE.E1) GO TO 999
0048  20  CONTINUE
0049      CALL IWPI3(I,J,IVAL)

```

```
0050          GO TO 1000
0051 999      CALL IWPI3(I, J, 0)
0052 1000    CONTINUE
0053        CALL CLPI1
0054        CALL CLPI2
0055        CALL CLPI3
0056        CALL ITTOUR("007")
0057        STOP 'ASSOCIATION COMPLETE'
0058        END
```

```

C LINKER FOR LOGICAL-LOGICAL, LOGICAL
C
C THIS PROGRAM TAKES MAGNITUDE AND ANGLE INPUT IMAGES AND
C LINKS TOGETHER THOSE EDGES WHICH SATISFY A PARTICULAR
C SET OF LINKING CRITERIA. THE PARAMETERS IN THE
C CRITERIA ARE:
C
C     EDGE LENGTH---THE NUMBER OF APPROXIMATELY CO-LINEAR
C                     POINTS WHICH DEFINE AN EDGE.
C     SKIP THRESHOLD---ALLOWABLE FOR THE BREAKAGE OF AN EDGE
C                     BY A MAXIMUM NUMBER OF ZERO POINTS.
C     LENGTH DIFFERENCE---THE PERCENTAGE DIFFERENCE
C                     ALLOWED BETWEEN TWO ADJACENT POINTS.
C     ANGLE DIFFERENCE---THE ANGULAR DIFFERENCE
C                     ALLOWED BETWEEN TWO ADJACENT POINTS.
C
C THESE THRESHOLDS ARE SET BY THE USER AT RUN TIME.
C
C TWO BASIC TECHNIQUES OF LINKING ARE ALLOWED. THE FIRST
C EXAMINES EACH POINT IN THE INPUT, REGARDLESS OF WHETHER IT
C HAS BEEN PREVIOUSLY IDENTIFIED AS A MEMBER OF AN EDGE.
C THE SECOND, MORE RAPID METHOD ZEROS POINTS IN THE INPUT IMAGE
C AS THEY ARE IDENTIFIED. THE SECOND METHOD DESTROYS ITS
C MAGNITUDE INPUT FILE.
C
C THE MAGNITUDE FILE CAN BE THE OUTPUT OF ANY EDGE DETECTOR,
C BUT THE ANGLE FILE MUST BE FROM THE DAVIES METHOD (DAVIES.FOR)
C
C AUTHOR: DAVID J. BRYANT
C     OCT 1978
C
0001 DIMENSION ISTAK(1024,2)
0002 INTEGER*2 OLPI1,OLPI2
0003 COMMON ISTAK
C
C USER INTERROGATION SECTION. ESTABLISHES THE LINKING METHOD
C AND PROMPTS FOR INPUT OF THE THRESHOLDS.
C
0004 TYPE 2
0005 2 FORMAT(' INPUT ZEROING OPTION FOR HIGHER SPEED? (1=YES)=>',%)
0006 ACCEPT 3,IOPT
0007 3 FORMAT(11)
0008 IF(IOPT.NE.1) GO TO 8
0009 TYPE 5
0010 5 FORMAT(' ***WARNING--THE MAGNITUDE FILE WILL BE DESTROYED***')
0011 8 TYPE 9
0012 9 FORMAT(//)
0013 NDIM=OLPI1()
0014 NDIM1=OLPI2()
0015 IF(NDIM.NE.NDIM1) STOP 'MISMATCHED INPUT'
0016 IF(NDIM.NE.NDIM1) STOP 'MISMATCHED OUTPUT'
0017 TYPE 10
0018 10 FORMAT(' INPUT THE EDGE-LENGTH THRESHOLD=>',%)
0019 ACCEPT 11,LTHR

```

```

0023 11  FORMAT(I2)
0024      TYPE 12
0025 12  FORMAT(' SKIP THRESHOLD=>',%)
0026      ACCEPT 11, IZTHR
0027      TYPE 14
0028 14  FORMAT(' LENGTH DIFFERENCE THRESHOLD (PERCENT)=>',%)
0029      ACCEPT 17, PTHR
0030      TYPE 16
0031 16  FORMAT(' ANGLE DIFFERENCE THRESHOLD (DEGREES)=>',%)
0032      ACCEPT 17, TTHR
0033 17  FORMAT(F4.3)
0034      TTHR=TTHR*3.14159/180.
      C
      C  ZERO THE OUTPUT IMAGE
      C
0035      DO 900 I=1,NDIM
0036      DO 900 J=1,NDIM
0037          CALL IWPI3(I,J,0)
0038 900  CONTINUE
      C
      C  BEGIN PROCESSING THE INPUT IMAGE
      C
0039      DO 1000 I=1,NDIM
0040      DO 1000 J=1,NDIM
0041          L=1
0042          IP1=IRPI1(I,J)
0043          IF(IP1.EQ.0) GO TO 1000
0045          IA1=IRPI2(I,J)
      C
      C  CALL SCAN TO CHECK FOR EDGE CONNECTIVITY IN THE
      C  FORWARD DIRECTION (IDIR=1)
      C
0046          IT=I
0047          JT=J
0048          CALL SCAN(IP1, IA1, IT, JT, IZTHR, L, PTHR, TTHR, 1, NDIM)
      C
      C  CALL SCAN AGAIN, THIS TIME FOR THE BACKWARD DIRECTION
      C  SEARCH (IDIR=-1)
      C
0049          IT=I
0050          JT=J
0051          CALL SCAN(IP1, IA1, IT, JT, IZTHR, L, PTHR, TTHR, -1, NDIM)
      C
      C  IF THE RETURNED LENGTH OF THE OVERALL EDGE IS GREATER THAN
      C  OR EQUAL TO THE EDGE LENGTH THRESHOLD, THE EDGE IS
      C  DRAWN IN THE OUTPUT IMAGE BY POPPING THE POINT COORDINATES OFF
      C  THE STACK (ISTAK). IF THE INPUT-ZEROING OPTION HAS BEEN
      C  SELECTED, THE INPUT IS ZEROED AS WELL.
      C
0052          IF(L.LT.LTHR) GO TO 30
0054          DO 20 II=1,L
0055              CALL IWPI3(ISTAK(II,1), ISTAK(II,2), 255)
0056 20  CONTINUE
0057 30  IF(IOPT.NE.1) GO TO 1000

```

0059	DO 35 II=1,L	
0060	CALL IWP11(ISTAK(11,1),ISTAK(11,2),0)	
0061	CONTINUE	35
0062	CONTINUE	1000
0063	CALL CLP11	
0064	CALL CLP12	
0065	CALL CLP13	
0066	CALL ITT0UR("007")	
0067	STOP	
0068	END	

```

0001 SUBROUTINE SCAN(P1, IAI, I, J, IZTHR, L, PTHR, TTHR, NDIR, NDIRM)
0002 THIS SUBROUTINE SCANS ALONG THE DIRECTION SPECIFIED
0003 BY THE ANGLE INFORMATION OF A POINT (X,Y) IN SEARCH OF
0004 NEIGHBORING POINTS WHICH SATISFY THE LINKING CRITERIA.
0005 IF THE NEXT POINT CHECKED IS WITHIN THESE LIMITS, IT
0006 IS CONSIDERED TO LIE ON THE SAME EDGE AS (X,Y) AND ITS
0007 COORDINATES ARE PUSHED ON THE STACK (ISTAK) FOR RETURN TO
0008 THE MAIN PROGRAM.
0009 DIMENSION IA(8), JA(8)
0010 DATA (A,JA/-1,0,3*1,0,2*-1,3*1,0,3*-1,0,
0011 COMMON ISTAK(1024,2)
0012 IZ=1
0013 ANG1=6.28318*FLOAT(IA1)/255.
0014 PUSH I AND J ONTO THE STACK AS EDGE POINTS
0015 ISTAK(L,1)=I
0016 ISTAK(L,2)=J
0017 COMPUTE THE COORDINATES OF THE NEXT POINT IN THE
0018 EDGE AS DIRECTED BY THE ANGLE OF (X,Y)
0019 IANG=FIX(.5+ANG1*4./3.14159)
0020 IF(IANG.EQ.0) IANG=8
0021 IZ=I+(IDIR*IA(IANG))
0022 JZ=J+(IDIR*JA(IANG))
0023 IF(IZ.LT.1.OR.IZ.GT.NDIRM) GO TO 30
0024 IF(JZ.LT.1.OR.JZ.GT.NDIRM) GO TO 30
0025 READ THE VALUES OF THE NEXT POINT.
0026 IPZ=IRP11(IZ,JZ)
0027 IAZ=IRP12(IZ,JZ)
0028 ANG2=6.28318*FLOAT(IAZ)/255.
0029 IF(IPZ.EQ.0) GO TO 20
0030 IZ=1
0031 IF THIS IS A BACKWARD SEARCH, THE ANGLE THRESHOLD
0032 MUST BE CENTERED AROUND PI INSTEAD OF ZERO.
0033 IF(101K.EQ.1) GO TO 4
0034 ANG2=ANG2-3.14159
0035 IF(ANG2.LT.0) ANG2=6.28318+ANG2
0036 IPTHR=FIX(.5+PTHR*FLOAT(IP1))
0037 CHECK NEW POINT AGAINST LINKING CRITERIA
0038 IF(IABS(IPZ-IP1).GE.IPTHR) GO TO 30
0039 IF(ABS(ANG2-ANG1).GE.ITHR) GO TO 30
0040 LINKING TEST IS PASSED. NEW CENTER POINT MUST BE ESTABLISHED

```

0034	IF1=IF2	
0035	IA1=IA2	
0036	ANG1=ANG2	
0037	L=L+1	5
0038	I=I2	
0039	J=J2	
0040	GO TO 10	
C		
C	THIS SECTION CHECKS TO SEE IF THE ZERO-SKIPPING THRESHOLD	
C	HAS BEEN EXCEEDED. IF NOT, CONTINUE AFTER INCREMENTING	
C	I2, THE ZERO COUNTER. IF SO, RETURN TO MAIN PROGRAM.	
0041	IF(I2.EQ.12THR) GO TO 30	20
0043	I2=I2+1	
0044	GO TO 5	
0045	RETURN	30
0046	END	

```

0001      INTEGER*2 OLPI1
0002      INTEGER HIST(256)
0003      DATA HIST/256*0/
0004      NDIM=OLPI1()
0005      IF(NDIM.NE.NMPI2()) STOP/MISMATCHED OUTPUT/
0006      TYPE 5
0007      FORMAT(/ THRESHOLD BY GREY LEVEL OR PERCENT? (1=%)>?,$)
0008      ACCEPT 6,IFLAG
0009      FORMAT(I1)
0010      IF(IFLAG.EQ.0) GO TO 50
0011
0012      THIS IS THE PERCENT THRESHOLDING SECTION. IT COMPUTES THE
0013      THRESHOLD LEVEL, ITHR, BY BACKING DOWN THE HISTOGRAM
0014      OF THE INPUT SCENE UNTIL THE SPECIFIED PERCENTILE HAS
0015      BEEN REACHED. ITHR IS THEN PASSED TO THE
0016      THRESHOLD SECTION.
0017      HIST IS THE HISTOGRAM OF THE INPUT SCENE.
0018      DO 10 I=1,NDIM
0019      DO 10 J=1,NDIM
0020      IVAL=IRPI1(I,J)+1
0021      HIST(IVAL)=HIST(IVAL)+1
0022      CONTINUE
0023      GET THE PERCENT CUT-OFF FROM USER, CALCULATE HOW MANY POINTS
0024      THIS TRANSLATES INTO, AND ECHO THIS BACK TO USER.
0025      TYPE 12
0026      FORMAT(/ ENTER THE PERCENT CUT-OFF (0. TO 100.)=>?,$)
0027      ACCEPT 14,PER
0028      FORMAT(F6.2)
0029      ICUT=IFIX(.5+PER*FLOAT(NDIM*2)/100.)

```

THR FOR LOGICAL
 THIS PROGRAM IS A THRESHOLDING UTILITY. IT OFFERS THREE
 DIFFERENT SCHEMES FOR CARRYING OUT THE THRESHOLDING.
 1) PERCENT THRESHOLDING--SELECTS THE THRESHOLD
 AT A USER SPECIFIED PERCENTILE.
 2) AVERAGE THRESHOLD--THE AVERAGE GREY LEVEL IS
 COMPUTED AND USED AS THE THRESHOLD.
 3) LEVEL THRESHOLD--THE USER SELECTS A GREY LEVEL
 (FROM 0 TO 255 INCLUSIVE) TO SERVE AS THE
 THRESHOLD.
 IN ALL CASES THE USER IS ASKED FOR THE NEW GREY LEVELS
 WHICH ARE TO BE USED IN THE BINARY RESULT. THESE MAY
 BE ANY LEVEL FROM 0 TO 255 (INCLUSIVE) AS WELL.

AUTHOR: DAVID J. BRYANT
 OCT 1978

```

0023      TYPE 15,PER,ICUT
0024      FORMAT(1X,'THE ',F6.2,' % CUT OFF IS AT ',I6)
0025      ICNT=0
0026      START AT THE TOP (GREY LEVEL 255) OF THE HISTOGRAM AND
0027      BACK DOWN UNTIL THE NUMBER OF POINTS ABOVE GREY LEVEL 1
0028      IS GREATER THAN THAT IMPLIED BY THE PERCENT CUT-OFF.
0029      DO 20 I=256,1,-1
0030      ICNT=ICNT+HIST(I)
0031      IF(ICNT.GE.ICUT) GO TO 30
0032      CONTINUE
0033      PASS ITHR ON TO THE THRESHOLDING SECTION
0034      ITHR=I-1
0035      GO TO 100
0036      THIS IS THE AVERAGE GREY LEVEL SECTION. THE
0037      AVERAGE IS COMPUTED VIA A PARTIAL SUM SCHEME SINCE
0038      NUMBERS WOULD BE GENERATED BY A TYPICAL AVERAGING
0039      APPROACH WHICH WOULD BE TOO BIG FOR THE COMPUTER.
0040      TYPE 52
0041      FORMAT(' AVERAGE GREY LEVEL AS THRESHOLD? (1=YES)=>',$)
0042      ACCEPT 6,IANS
0043      IF(IANS.NE.1) GO TO 54
0044      CNT=0.
0045      SUM=0.
0046      THE AVERAGE IS COMPUTED BY COUNTING HOW MANY PARTIAL
0047      SUMS OF 16384 ARE IN THE TOTAL SUM OF ALL THE GREY LEVELS.
0048      THIS COUNT IS THEN CONVERTED TO THE AVERAGE FOR THE
0049      SCENE.
0050      DO 53 I=1,NDIM
0051      DO 53 J=1,NDIM
0052      IVAL=IRP1(I,J)
0053      SUM=SUM+FLOAT(IVAL)
0054      IF(SUM.LT.16384.) GO TO 53
0055      CNT=CNT+1.
0056      SUM=SUM-16384.
0057      CONTINUE
0058      TAVG=CNT+SUM/16384.
0059      AVG=TAVG/((FLOAT(NDIM)/128.))**2)
0060      THE THRESHOLD LEVEL ITHR IS EQUAL TO THE AVERAGE GREY LEVEL.
0061      ITHR=IFIX(.5+AVG)
0062      TYPE 2000,ITHR
0063      FORMAT(' THE AVERAGE GREY LEVEL IS ',I3)
0064      GO TO 100
0065      TYPE 55
0066      FORMAT(' ENTER THE GREY LEVEL CUT-OFF (0 TO 255)=>',$)

```

0057	ACCEPT 60, ITHR	
0058	FORMAT(13)	60
0059	TYPE 102	100
0060	FORMAT(/ VALUE FOR POINTS ABOVE THRESHOLD (0 TO 255)=> /, \$)	102
0061	ACCEPT 60, IHI	
0062	TYPE 104	
0063	FORMAT(/ VALUE FOR POINTS BELOW THRESHOLD (0 TO 255)=> /, \$)	104
0064	ACCEPT 60, ILO	
0065	DO 150 I=1, NDIH	
0066	DO 150 J=1, NDIH	
0067	IVAL=IRPI1(I, J)	
0068	IF(IVAL GE. ITHR) IVAL=IHI	
0070	IF(IVAL LT. ITHR) IVAL=ILO	
0072	CALL IMP12(I, J, IVAL)	150
0073	CONTINUE	
0074	CALL CLP11	
0075	CALL CLP12	
0076	STOP / THRESHOLDING COMPLETE /	
0077	END	

00 60 J=ISIA/ IENDY	0051
IBUFF(J-ISIA+1)=IRP11(I,J)	0052
CONTINUE	60
WRITE(LUN,55) (IBUFF(JJ),JJ=1,IYDL)	0054
FORMAT(1X,3214)	55
CONTINUE	50
CALL CLP11	0057
STOP	0058
END	0059