USC–IPI REPORT #860

Semi-Annual Technical Report

by

Harry C. Andrews and William K. Pratt

March 1979

Signal and Image Processing Institute
UNIVERSITY OF SOUTHERN CALIFORNIA
Department of Electrical Engineering-Systems
3740 McClintock Avenue, Room 404
Los Angeles, CA 90089-2564 U.S.A.

ABSTRACT

This technical report summarizes the image understanding, image processing, and smart sensor research activities performed by the USC Image Processing Institute during the period of 1 October 1978 through 31 March 1979 under contract number F-33615-76-C-1203 with the Advanced Research Projects Agency, Information Processing Techniques Office, and monitored by the Wright-Patterson Air Force Base, Dayton, Ohio.

The research program has, as its primary purpose, the development of techniques and systems for understanding images. Methodologies range from low level image processing principles, smart sensor CCD LSI circuit design, up to higher level symbolic representations and relational structure manipulations.

PROJECT PARTICIPANTS

Project Director	Affiliation
Harry C. Andrews	Computer Sciences & Electrical Engineering
William K. Pratt	Electrical Engineering

Research Staff	Affiliation
Ramakant Nevatia	Computer Science
Keith E. Price	Image Processing Institute
Alexander A. Sawchuk	Electrical Engineering
Timothy C. Strand	Image Processing Institute

Visiting Scientist	Affiliation
Jean-Francois Abramatic	I.R.I.A. - FRANCE

Support Staff

Ted Bleecker
Gary Edwards
John Horner
Eileen Jurak
Hilda Marti
Toyone Mayeda
Charles McManis
Mary Monson
Michael Muha
Ray Schmidt
James Tertocha
Thomas Tertocha
Amy Yiu

ABSTRACT

This technical report summarizes the image understanding, image processing, and smart sensor research activities performed by the USC Image Processing Institute during the period of 1 October 1978 through 31 March 1979 under contract number F-33615-76-C-1203 with the Advanced Research Projects Agency, Information Processing Techniques Office, and monitored by the Wright-Patterson Air Force Base, Dayton, Ohio.

The research program has, as its primary purpose, the development of techniques and systems for understanding images. Methodologies range from low level image processing principles, smart sensor CCD LSI circuit design, up to higher level symbolic representations and relational structure manipulations.

Students

Ahmad Armand
Behnam Ashjari
K. Ramesh Babu
Bir Bhanu
Chung-Ching Chen
Peter Chuan
Fabrice Clara
David Garber
Chung-Kai Hsueh
Kenneth I. Laws
Sang Uk Lee
Chun Moo Lo
Howard Shao
Felicia Vilnrotter

TABLE OF CONTENTS

TABLE OF CONTENTS

1. Research Overview

This document presents results of research over the past six months at the USC Image Processing Institute. Research has been devoted to 3 major areas: image understanding, image processing, and smart sensor design. These areas are abstracted below.

Image Understanding Projects

The image understanding projects investigated during the past research period fall into two categories: texture analysis and image analysis. Pratt and Faugeras report on a method of texture feature extraction in which an image is first decorrelated, and large area spatial moments are formed as texture features. Laws presents an extension of this work concerned with an investigation of generalized spatial operators performing pseudo-decorrelation. A method of structural texture description based upon an edge representation of an image field is presented by Nevatia, Price, and Vilnrotter. Ashjari and Pratt discuss texture feature extraction from another viewpoint in their report of the use of singular values of a texture field as texture vector components.

Babu and Nevatia describe the use of linear features derived from image edges as a means of detecting roads in aerial imagery. Price reports on another image analysis project involving model matching at the symbolic level.

Image Processing Projects

The image processing projects reported during the research period are concerned with image processing system architecture, image restoration, radar image formation, and computer holography.

Pratt, Abramatic, and Lee describe a novel architecture for performing two-dimensional convolution with a minimum amount of hardware using the concept of sequential convolution with small generating kernels. Lee and Pratt present an algorithm for computing the condition number of a Wiener image restoration operator as a means of predicting the numerical accuracy of the restoration process. Two reports by Lo and Sawchuk describe image restoration for blurred images subjected to Poisson sensor noise. Garber and Morton describe a method of a posteriori image restoration. Chuan presents two reports; the first is concerned with errors associated with data sampling in the polar domain, and the second report is an application of the theory involving synthetic aperture radar imaging.

Smart Sensor Projects

The Hughes Research Laboratories present a section describing research progress on the development of smart sensors for image processing. Hughes is presently completing construction of a new CCD chip that performs the following functions:

3x3 Laplacian
5x5 median filter
5x5 programable weight convolver
7x7 bipolar convolver
26x26 edge detection convolver.

1. Research Overview

This document presents results of research over the past six months at the USC Image Processing Institute. Research has been devoted to 3 major areas: image understanding, image processing, and smart sensor design. These areas are abstracted below.

Image Understanding Projects

The image understanding projects investigated during the past research period fall into two categories: texture analysis and image analysis. Pratt and Faugeras report on a method of texture feature extraction in which an image is first decorrelated, and large area spatial moments are formed as texture features. Laws presents an extension of this work concerned with an investigation of generalized spatial operators performing pseudo-decorrelation. A method of structural texture description based upon an edge representation of an image field is presented by Nevatia, Price, and Vilnrotter. Ashjari and Pratt discuss texture feature extraction from another viewpoint in their report of the use of singular values of a texture field as texture vector components.

Babu and Nevatia describe the use of linear features derived from image edges as a means of detecting roads in aerial imagery. Price reports on another image analysis project involving model matching at the symbolic level.

Image Processing Projects

The image processing projects reported during the research period are concerned with image processing system architecture, image restoration, radar image formation, and computer holography.

Pratt, Abramatic, and Lee describe a novel architecture for performing two-dimensional convolution with a minimum amount of

2. Image Understanding Projects

2.1 Decorrelation Methods of Texture Feature Extraction

William K. Pratt and Olivier D. Faugeras*

Introduction

Previous studies [1-7] have helped to establish bounds for developing stochastic-based methods of visual texture feature extraction. It has been demonstrated that second order statistical meaures on stochastic texture fields are sufficient in the sense that human observers cannot distinguish between texture field pairs differing only in third and higher order statistics. Furthermore, it has been shown that mean, variance, and autocorrelation measures, by themselves, are not sufficient. These results have led to a new method of texture feature extraction based on spatial moment measurements of a decorrelated versin of the texture field [8].

Texture Feature Extraction Method

Stochatic texture fields can be computer generated by the system of Figure 1. An array of independent random numbers W(j,k) with probability density P(W) is input to a spatial operator with transfer function$\mathcal{O}\{\cdot\}$ to produce the correlated texture field F(j,k).

Texture fields generated by the model of Figure 1 can be compactly specified by P(W) and $\mathcal{O}\{\cdot\}$. This observation has led to a texture field description method by which F(j,k) is decorrelated to estimate W(j,k) and a histogram of the decorrelated field is formed as an estimate of P(W). The spatial operator $\mathcal{O}\{\cdot\}$ can be described by measurement of the autocorrelation function

*Dr. O.D. Faugeras is with Institut de Recherche d'informatique et d'automatique, Domaine de Voluceau, Le Chesnay, France.

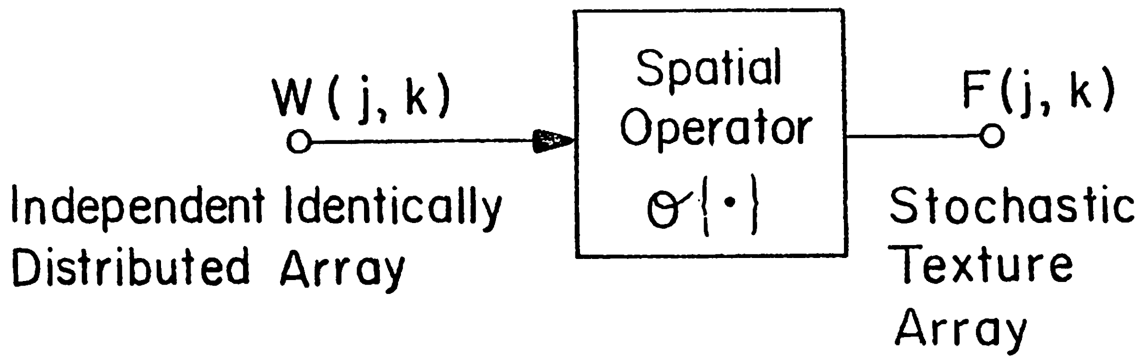


Figure 1. Stochastic texture field generation model

2. Image Understanding Projects

2.1 Decorrelation Methods of Texture Feature Extraction

William K. Pratt and Olivier D. Faugeras*

Introduction

Previous studies [1-7] have helped to establish bounds for developing stochastic-based methods of visual texture feature extraction. It has been demonstrated that second order statistical meaures on stochastic texture fields are sufficient in the sense that human observers cannot distinguish between texture field pairs differing only in third and higher order statistics. Furthermore, it has been shown that mean, variance, and autocorrelation measures, by themselves, are not sufficient. These results have led to a new method of texture feature extraction based on spatial moment measurements of a decorrelated versin of the texture field [8].

Texture Feature Extraction Method

Stochatic texture fields can be computer generated by the system of Figure 1. An array of independent random numbers $W(j,k)$ with probability density $P(W)$ is input to a spatial operator with transfer function$\mathcal{O}\{\cdot\}$ to produce the correlated texture field $F(j,k)$.

Texture fields generated by the model of Figure 1 can be compactly specified by $P(W)$ and $\mathcal{O}\{\cdot\}$. This observation has led to a texture field description method by which $F(j,k)$ is decorrelated to estimate $W(j,k)$ and a histogram of the decorrelated field is formed as an estimate of $P(W)$. The spatial operator $\mathcal{O}\{\cdot\}$ can be described by measurement of the autocorrelation function

*Dr. O.D. Faugeras is with Institut de Recherche d'informatique et d'automatique, Domaine de Voluceau, Le Chesnay, France.

$$A_F(m,n) = \sum_{u=j-W}^{j+W} \sum_{v=k-W}^{k+W} F(u,v)F(u-m,v-n) \qquad (1)$$

of F(j,k) computed over a (2W+1) by (2W+1) window.

Figure 2 contains a block diagram of the exture feature extraction method. In this system, the texture field sample is decorrelated by a whitening filter based on the mesured autocorrelation function $A_F(m,n)$ or by a fixed Laplacian or Sobel operator. A histogram P(b) for $0 \leq b \leq L-1$ amplitude levels is formed over a window of the decorrelated field and the first four moments of the histogram, defined below, are computed.

average

$$b_A = \sum_{b=0}^{L-1} bP(b) \qquad (2)$$

deviation

$$b_D = \left[\sum_{b=0}^{L-1} (b-b_A)^2 P(b)\right]^{\frac{1}{2}} \qquad (3)$$

skewness

$$b_S = \frac{1}{b_D^3} \sum_{b=0}^{L-1} (b-b_A)^3 P(b) \qquad (4)$$

kurtosis

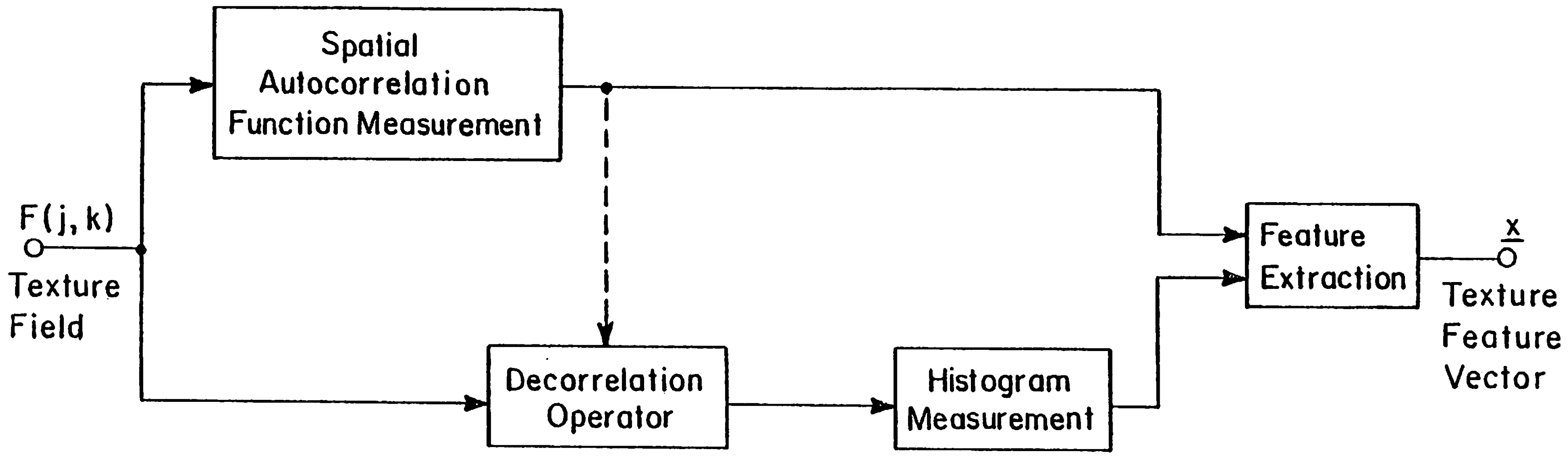


Figure 2. Decorrelation method of texture feature extraction

$$A_F(m,n) = \sum_{u=j-W}^{j+W} \sum_{v=k-W}^{k+W} F(u,v)F(u-m,v-n) \qquad (1)$$

of F(j,k) computed over a (2W+1) by (2W+1) window.

Figure 2 contains a block diagram of the exture feature extraction method. In this system, the texture field sample is decorrelated by a whitening filter based on the mesured autocorrelation function $A_F(m,n)$ or by a fixed Laplacian or Sobel operator. A histogram P(b) for $0 \leq b \leq L-1$ amplitude levels is formed over a window of the decorrelated field and the first four moments of the histogram, defined below, are computed.

average

$$b_A = \sum_{b=0}^{L-1} bP(b) \qquad (2)$$

deviation

$$b_D = \left[\sum_{b=0}^{L-1} (b-b_A)^2 P(b) \right]^{\frac{1}{2}} \qquad (3)$$

skewness

$$b_S = \frac{1}{b_D^3} \sum_{b=0}^{L-1} (b-b_A)^3 P(b) \qquad (4)$$

kurtosis

$$b_K = \frac{1}{b_D^4} \sum_{b=0}^{L-1} (b-b_A)^4 P(b) - 3 \tag{5}$$

The autocorrelation function is characterized by a set of two-dimensional spread measures defined by

$$S(u,v) = \sum_{m=0}^{T} \sum_{n=-T}^{T} (m-\eta_m)^u (n-\eta_n)^v A_F(m,n) \tag{6}$$

where

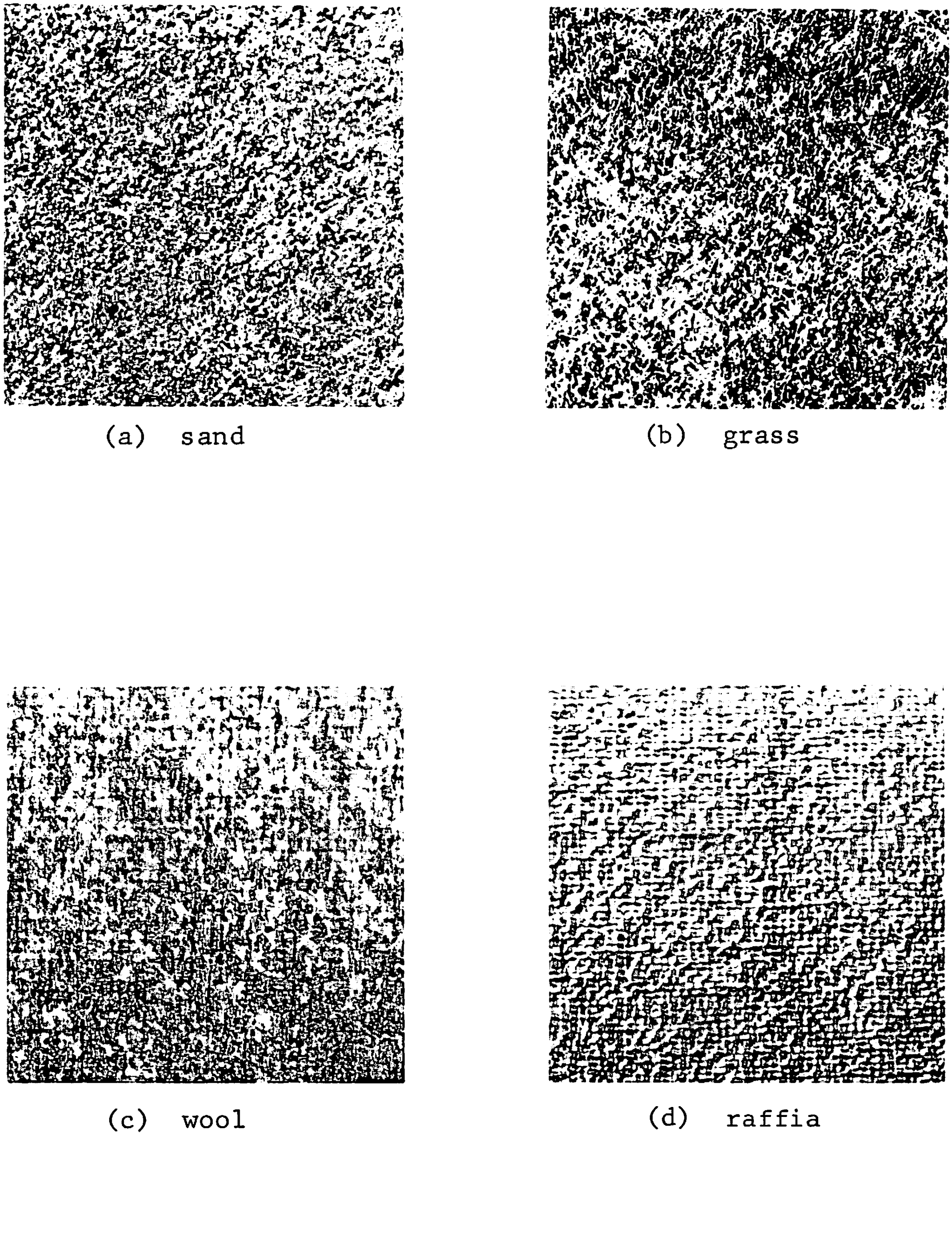
$$\eta_m = \sum_{m=0}^{T} \sum_{n=-T}^{T} m A_F(m,n) \tag{7}$$

$$\eta_n = \sum_{m=0}^{T} \sum_{n=-T}^{T} n A_F(m,n) \tag{8}$$

Evaluation

The decorrelation method of texture feature extraction previously described has been evaluated by measurement of the Bhattacharyya distance of texture features measured on pairs of the Brodatz [9] natural texture fields of Figure 3. The B-distance measure is

$$B(S_1,S_2) = \frac{1}{8}(\underline{u}_1-\underline{u}_2)\mathrm{T}\left[\frac{\underline{\Sigma}_1+\underline{\Sigma}_2}{2}\right]^{-1}(\underline{u}_1-\underline{u}_2)+\frac{1}{2}\ln\left\{\frac{\left|\frac{1}{2}(\underline{\Sigma}_1+\underline{\Sigma}_2)\right|}{|\underline{\Sigma}_1|^{\frac{1}{2}}|\underline{\Sigma}_2|^{\frac{1}{2}}}\right\} \tag{9}$$



(a) sand

(b) grass

(c) wool

(d) raffia

Figure 3. Examples of Brodatz Texture Fields.

where $\underline{u}_i$ and $\underline{\Sigma}_i$ represent the feature mean vector and the feature covariance matrix of the classes, respectively. For equally likely texture field pairs, a B-distance of 4 or greater corresponds to a classificaton error bound of about 1%.

In the experiments, the Brodatz texture fields have been subdivided into 64 non-overlapping prototype regions of 64x64 pixels. Texture features have been extracted from each region and formed into a texture feature vector. Next, the mean and covariance of the feature vector have been computed and substituted into eq.(9) to obtain the B-distance for pairs of prototype fields. In order to create a stringent test, the natural texture fields have been normalized to zero mean and unit standard deviation by independent point-by-point linear re-scaling. This operation insures that luminance bias and contrast differences between the texture pairs do not influence the discrimination.

Table 1 contains a listing of B-distances for three texture feature sets that measure the shape of the autocorrelation function of each prototype field for 20 spatial lags in each coordinate. With feature set 1, containing four features, the B-distances of the natural texture fields range from 8.70 to 1.49 corresponding to classification error bounds from about near zero to 11%. The B-distances are much smaller for feature sets 2 and 3 employing two features and one feature, respectively. The B-distance measurements of Table 1 indicate that autocorrelation shape features of texture fields, by themselves, are marginally adequate for the natural texture fields investigated.

Table 2 contains listings of B-distances for texture features consisting of histogram moments of decorrelated texture fields using whitening, Laplacian, and Sobel decorrelation operators. For the whitening operator, the average distance for the natural texture pairs

Table 1

Bhattacharyya Distance of Texture Feature Sets for Prototype Texture Fields Autocorrelation Features

FIELD	PAIRS	SET #1	SET #2	SET #3
GRASS	SAND	5.05	4.29	2.92
GRASS	RAFFIA	7.07	5.32	3.57
GRASS	WOOL	2.37	0.21	0.04
SAND	RAFFIA	1.49	0.58	0.35
SAND	WOOL	6.55	4.93	3.14
RAFFIA	WOOL	8.70	5.96	3.78
AVERAGE		5.21	3.55	2.30

SET #1: S(2,0), S(0,2), S(1,1), S(2,2)

SET #2: S(1,1), S(2,2)

SET #3: S(2,2)

where $\underline{u}_i$ and $\underline{\Sigma}_i$ represent the feature mean vector and the feature covariance matrix of the classes, respectively. For equally likely texture field pairs, a B-distance of 4 or greater corresponds to a classificaton error bound of about 1%.

In the experiments, the Brodatz texture fields have been subdivided into 64 non-overlapping prototype regions of 64x64 pixels. Texture features have been extracted from each region and formed into a texture feature vector. Next, the mean and covariance of the feature vector have been computed and substituted into eq.(9) to obtain the B-distance for pairs of prototype fields. In order to create a stringent test, the natural texture fields have been normalized to zero mean and unit standard deviation by independent point-by-point linear re-scaling. This operation insures that luminance bias and contrast differences between the texture pairs do not influence the discrimination.

Table 1 contains a listing of B-distances for three texture feature sets that measure the shape of the autocorrelation function of each prototype field for 20 spatial lags in each coordinate. With feature set 1, containing four features, the B-distances of the natural texture fields range from 8.70 to 1.49 corresponding to classification error bounds from about near zero to 11%. The B-distances are much smaller for feature sets 2 and 3 employing two features and one feature, respectively. The B-distance measurements of Table 1 indicate that autocorrelation shape features of texture fields, by themselves, are marginally adequate for the natural texture fields investigated.

Table 2 contains listings of B-distances for texture features consisting of histogram moments of decorrelated texture fields using whitening, Laplacian, and Sobel decorrelation operators. For the whitening operator, the average distance for the natural texture pairs

Table 2

Bhattacharyya Distance of Texture Feature Sets For Prototype
Texture Fields-Histogram Features

FIELD PAIRS		TEXTURE FEATURES											
		WHITENING				LAPLACIAN				SOBEL			
		SET#1	SET#2	SET#3	SET#4	SET#1	SET#2	SET#3	SET#4	SET#1	SET#2	SET#3	SET#4
Grass	Sand	4.61	4.52	4.04	0.77	1.29	1.28	0.19	0.66	9.90	7.15	4.41	2.31
Grass	Raffia	1.15	1.04	0.51	0.52	3.48	3.38	0.55	1.87	2.20	1.00	0.27	0.02
Grass	Wool	1.68	1.59	1.07	0.14	2.23	2.19	1.76	0.13	2.98	1.67	1.01	1.46
Sand	Raffia	12.76	12.60	10.93	0.24	2.23	2.14	1.57	0.28	5.09	4.79	3.51	2.30
Sand	Wool	12.61	12.55	8.24	2.19	7.73	7.65	7.42	1.40	9.98	5.01	1.67	0.56
Raffia	Wool	4.02	3.87	0.39	1.47	4.59	4.43	1.53	3.13	7.73	2.31	0.41	1.41
AVERAGE		6.14	6.03	4.20	0.88	3.59	3.51	2.17	1.24	6.31	3.66	1.88	1.35

SET#1: b_A, b_D, b_S, b_K
SET#2: b_S, b_K
SET#3: b_S
SET#4: b_K

is quite large when all four moment features are utilized, but some texture pairs, e.g. grass-raffia, exhibit small distance. There is relatively little drop in B-distance when only the third and fourth order histogram moments, b_S and b_K, are used. This is to be expected since the whitened texture fields have been forced to zero mean and unit variance by the whitening operator. Use of only the kurtosis gives small distances.

With a Laplacian decorrelation operator, the B-distances of Table 2 are somewhat lower on average than for a whitening operator. However, there are some anomalies. Compare, for example, the grass-raffia distances for whitening and Laplacian decorrelation.

Decorrelation with the Sobel operator, as indicated in Table 2, gives quite large B-distances for natural textures using four histogram moments. Since the Sobel operator output is unipolar, the mean and standard deviation moments are meaningful, and in fact, contribute significantly to the B-distances. In the worst case of the grass-raffia pair, the B-distance of 2.20 corresponds to a classification error bound of about only 5% using feature set 1.

The conclusions obtained from Table 2 are that histogram moment features of decorrelated texture fields, by themselves, provide a reasonably good means of discriminating the natural texture fields investigated. The whitening operator is superior, on the average, to the Laplacian operator in terms of distance. But, the Sobel operator yields the largest average and largest minimum distances. This is particularly interesting since use of the Sobel operator obviously obviates the need to compute the autocorrelation function and generate the whitening filter.

Table 3 lists the B-distances obtainable using a hybrid feature set of autocorrelation shape and histogram moment features. In all cases, the B-distances are larger than obtained using only autocorrelation shape or histogram moment features.

is quite large when all four moment features are utilized, but some texture pairs, e.g. grass-raffia, exhibit small distance. There is relatively little drop in B-distance when only the third and fourth order histogram moments, b_S and b_K, are used. This is to be expected since the whitened texture fields have been forced to zero mean and unit variance by the whitening operator. Use of only the kurtosis gives small distances.

With a Laplacian decorrelation operator, the B-distances of Table 2 are somewhat lower on average than for a whitening operator. However, there are some anomalies. Compare, for example, the grass-raffia distances for whitening and Laplacian decorrelation.

Decorrelation with the Sobel operator, as indicated in Table 2, gives quite large B-distances for natural textures using four histogram moments. Since the Sobel operator output is unipolar, the mean and standard deviation moments are meaningful, and in fact, contribute significantly to the B-distances. In the worst case of the grass-raffia pair, the B-distance of 2.20 corresponds to a classification error bound of about only 5% using feature set 1.

The conclusions obtained from Table 2 are that histogram moment features of decorrelated texture fields, by themselves, provide a reasonably good means of discriminating the natural texture fields investigated. The whitening operator is superior, on the average, to the Laplacian operator in terms of distance. But, the Sobel operator yields the largest average and largest minimum distances. This is particularly interesting since use of the Sobel operator obviously obviates the need to compute the autocorrelation function and generate the whitening filter.

Table 3 lists the B-distances obtainable using a hybrid feature set of autocorrelation shape and histogram moment features. In all cases, the B-distances are larger than obtained using only autocorrelation shape or histogram moment features.

Table 3

Bhattacharyya Distance Of Texture Feature Sets For Prototype Texture Fields-Autocorrelation And Histogram Features

FIELD PAIRS		TEXTURE FEATURES											
		WHITENING				LAPLACIAN				SOBEL			
		SET#1	SET#2	SET#3	SET#4	SET#1	SET#2	SET#3	SET#4	SET#1	SET#2	SET#3	SET#4
Grass	Sand	9.80	9.72	8.94	7.48	6.39	6.37	5.61	4.21	15.34	12.34	11.48	10.12
Grass	Raffia	8.47	8.34	6.56	4.66	10.61	10.49	8.74	6.95	9.46	8.15	6.33	4.59
Grass	Wool	4.17	4.03	1.87	1.70	4.64	4.59	2.48	2.31	5.62	4.05	1.87	1.72
Sand	Raffia	15.26	15.08	13.22	12.98	3.85	3.76	2.74	2.49	6.75	6.40	5.39	5.13
Sand	Wool	19.14	19.08	17.43	15.72	14.43	14.38	12.72	10.86	18.75	12.36	10.52	8.29
Raffia	Wool	13.29	13.14	10.32	7.96	13.93	13.75	10.90	8.47	17.28	11.19	8.24	6.08
AVERAGE		11.69	11.57	9.72	8.42	8.98	8.89	7.20	5.88	12.20	9.08	7.31	5.99

SET#1: b_A, b_D, b_S, b_K, S(2,0), S(0 2), S(1,1), S(2,2)
SET#2: b_S, b_K, S(2,0), S(0,2), S(1,1), S(2,2)
SET#3: b_S, b_K, S(1,1), S(2,2)
SET#4: b_S, b_K, S(2,2)

The previous results seem to indicate that the histogram of a decorrelated texture field provides a substantial amount of information for texture discrimination. But, how important is the decorrelation process? What performance could be achieved if no decorrelation were to be performed, and the histogram were made directly on the texture field? Table 4 provides some answers to these questions.

Table 4 contains B-distances obtained with four histogram moment features using whitening, Laplacian, and Sobel decorrelation operators and with no decorrelation at all. This data, presented in the first four columns of Table 4, has been obtained by processing texture fields normalized to zero mean and unit standard deviation. The results show that, without decorrelation, fairly large B-distances can be obtained for most of the natural texture field pairs. Thus, the first order histogram of a texture field seemingly provides information important for texture discrimination. But, is this really so? Probably not, because the first order histogram of an image is dependent upon luminance point response of the imaging system in addition to the point reflectivity of the texture object. It is possible to nonlinearly scale the prototype texture fields such that their histograms are all identical. Yet the fields will retain visual texture differences. The B-distances obtained using such images as prototypes are presented in the last four columns of Table 4. It is observed that the B-distances for no decorrelation have become extremely small as expected, but the distances for the other decorrelation operators are not affected nearly so much. Moreover, the whitening operator yields the largest average and largest minimum distance. Thus, the justification for the decorrelation operation is strongly enforced.

Summary and Conclusions

A stochastic model of texture field generation has led to the development of a texture feature extraction technique. The method is

The previous results seem to indicate that the histogram of a decorrelated texture field provides a substantial amount of information for texture discrimination. But, how important is the decorrelation process? What performance could be achieved if no decorrelation were to be performed, and the histogram were made directly on the texture field? Table 4 provides some answers to these questions.

Table 4 contains B-distances obtained with four histogram moment features using whitening, Laplacian, and Sobel decorrelation operators and with no decorrelation at all. This data, presented in the first four columns of Table 4, has been obtained by processing texture fields normalized to zero mean and unit standard deviation. The results show that, without decorrelation, fairly large B-distances can be obtained for most of the natural texture field pairs. Thus, the first order histogram of a texture field seemingly provides information important for texture discrimination. But, is this really so? Probably not, because the first order histogram of an image is dependent upon luminance point response of the imaging system in addition to the point reflectivity of the texture object. It is possible to nonlinearly scale the prototype texture fields such that their histograms are all identical. Yet the fields will retain visual texture differences. The B-distances obtained using such images as prototypes are presented in the last four columns of Table 4. It is observed that the B-distances for no decorrelation have become extremely small as expected, but the distances for the other decorrelation operators are not affected nearly so much. Moreover, the whitening operator yields the largest average and largest minimum distance. Thus, the justification for the decorrelation operation is strongly enforced.

Summary and Conclusions

A stochastic model of texture field generation has led to the development of a texture feature extraction technique. The method is

Table 4

Bhattacharyya Distance Of Texture Feature Sets For Prototype Texture Fields-Histogram Features With And Without Histogram Modification

FIELD PAIRS		WITHOUT HISTOGRAM MODIFICATION				WITH HISTOGRAM MODIFICATION			
		WHITE	LAPLACIAN	SOBEL	NULL	WHITE	LAPLACIAN	SOBEL	NULL
Grass	Sand	4.61	1.29	9.90	4.56	3.89	0.74	1.53	0.06
Grass	Raffia	1.15	3.48	2.20	0.37	0.98	3.24	2.70	0.02
Grass	Wool	1.68	2.23	2.98	3.80	1.61	1.07	1.52	0.70
Sand	Raffia	12.76	2.23	5.09	7.68	7.63	3.61	0.35	0.11
Sand	Wool	12.61	7.73	9.98	7.42	6.93	0.84	1.73	0.98
Raffia	Wool	4.02	4.59	7.73	5.16	1.06	3.49	2.95	0.65
AVERAGE		6.14	3.59	6.31	4.83	3.68	2.16	1.80	0.42

SET: b_A, b_D, b_S, b_K

based on representation of the autocorrelation function of a texture field plus the gray scale histogram of a decorrelated version of the texture field. Feature repesentation is in terms of shape measures of the autocorrelation function and moments of the histogram. The feature vector so obtained has been evaluated by Bhattacharyya distance measurements. Testing with prototype texture fields indicates that large Bhattacharyya distances can be obtained between texture field pairs with the stochastic-based feature extraction method.

References

1. B. Julesz, "Visual Pattern Discrimination," IRE Transactions on Information Theory, Vol. IT-8, no. 1, pp. 84-92, February 1962.

2. B. Julesz, Foundations of Cyclopean Perception, University of Chicago Press, 1970.

3. B. Julesz, E.N. Gilbert, L.A. Shepp, and H.L. Frisch, "Inability of Humans to Discriminate Between Visual Textures that Agree in Second-Order Statistics - Revisited," Perception, Vol. 2, pp.391-405, 1973.

4. B. Julesz, "Experiments in the Visual Perception of Texture," Scientific American, Vol. 232, no.4, pp.34-43, April 1975.

5. I. Pollack, Perceptual Psychophysics, Vol. 13, pp.276-280, 1973.

6. S.R. Purks and W. Richards, "Visual Texture Discrimination Using Random Dot Patterns," Journal Optical Society of America, Vol. 67, no.6, pp.765-771, June 1977.

7. W.K. Pratt, O.D. Faugeras, and A. Gagalowicz, "Visual Discrimination of Stochastic Texture Fields," IEEE Transactions on Systems, Man, and Cybernetics, November 1978.

based on representation of the autocorrelation function of a texture field plus the gray scale histogram of a decorrelated version of the texture field. Feature repesentation is in terms of shape measures of the autocorrelation function and moments of the histogram. The feature vector so obtained has been evaluated by Bhattacharyya distance measurements. Testing with prototype texture fields indicates that large Bhattacharyya distances can be obtained between texture field pairs with the stochastic-based feature extraction method.

References

1. B. Julesz, "Visual Pattern Discrimination," IRE Transactions on Information Theory, Vol. IT-8, no. 1, pp. 84-92, February 1962.

2. B. Julesz, Foundations of Cyclopean Perception, University of Chicago Press, 1970.

3. B. Julesz, E.N. Gilbert, L.A. Shepp, and H.L. Frisch, "Inability of Humans to Discriminate Between Visual Textures that Agree in Second-Order Statistics - Revisited," Perception, Vol. 2, pp.391-405, 1973.

4. B. Julesz, "Experiments in the Visual Perception of Texture," Scientific American, Vol. 232, no.4, pp.34-43, April 1975.

5. I. Pollack, Perceptual Psychophysics, Vol. 13, pp.276-280, 1973.

6. S.R. Purks and W. Richards, "Visual Texture Discrimination Using Random Dot Patterns," Journal Optical Society of America, Vol. 67, no.6, pp.765-771, June 1977.

7. W.K. Pratt, O.D. Faugeras, and A. Gagalowicz, "Visual Discrimination of Stochastic Texture Fields," IEEE Transactions on Systems, Man, and Cybernetics, November 1978.

8. O.D. Faugeras and W.K. Pratt, "Decorrelation Methods of Texture Feature Extraction", (submitted for publication), January 1979.

9. P. Brodatz, Textures: A Photograph Album for Artists and Designers, Dover. New York, 1956.

2.2 Stochastic Texture Characterization

Kenneth I. Laws

Visual textures arise from many sources. Cellular textures are composed of repeated similar elements called primitives. Examples are leaves on a tree or bricks in a wall. Other texture types include flow patterns, fiber masses, wood grains, and stress cracking. A complete analysis of any texture would require modeling of the underlying physical structure.

The human visual system, however, is capable of discriminating and classifying all of these textures. It is obvious that spontaneous recognition does not require built-in models of physical texture generators, although such models may be used by trained observers.

The eye must use the same feature extraction methods on each texture field, regardless of its source. We do not know what these methods are, although there is indirect evidence that edge detection is involved. We do know that any retinal transform must retain enough information to distinguish different textures (as identified by human observers). Information which would distinguish equivalent textures must be suppressed or ignored.

The chief characteristic of texture is shift-invariance. Perception of a texture does not change as its position on the fovea changes. This seems to be the very definition of a texture field: an image which is not significantly changed by shifting. A region or object, on the other hand, is position dependent.

Textures are often composed of identifiable sub-regions. Texture perception is not invariant to all rearrangements of these regions. Whether an image is seen as a uniform texture field or as an arrangement of regions seems to depend on two factors: scale and discontinuity. Large regions with closed boundaries are seen as separate objects. Small regions with indistinct edges are seen as a texture field.

We shall define texture to be that which remains constant as a window (or fovea) is moved across an image. This presupposes that the image is a single texture field. The definition does not explicitly include the closed boundary effect, but does include the resolution ambiguity: texture may change as a function of window size.

There is another ambiguity in the common meaning of texture. Let two texture fields be identical except for a difference in luminance. Most observers will say that the textures are identical, although the two fields are easily distinguished. Similar results will be obtained with texture fields differing in contrast, color, size, rotation, or geometric warp. Texture is thus perceived to be invariant to changes in illumination or camera position.

We shall consider all of these differences to be differences in texture, although ones which are easily measured or compensated. Experimental work for this study uses monochrome images quantized to have nearly uniform gray-level histograms. This compensates for any differences in illumination, sensor type, or film developing parameters. We will also attempt to measure and adjust for camera orientation parameters, although it is not clear whether these differ

8. O.D. Faugeras and W.K. Pratt, "Decorrelation Methods of Texture Feature Extraction", (submitted for publication), January 1979.

9. P. Brodatz, Textures: A Photograph Album for Artists and Designers, Dover. New York, 1956.

2.2 Stochastic Texture Characterization

Kenneth I. Laws

Visual textures arise from many sources. Cellular textures are composed of repeated similar elements called primitives. Examples are leaves on a tree or bricks in a wall. Other texture types include flow patterns, fiber masses, wood grains, and stress cracking. A complete analysis of any texture would require modeling of the underlying physical structure.

The human visual system, however, is capable of discriminating and classifying all of these textures. It is obvious that spontaneous recognition does not require built-in models of physical texture generators, although such models may be used by trained observers.

The eye must use the same feature extraction methods on each texture field, regardless of its source. We do not know what these methods are, although there is indirect evidence that edge detection is involved. We do know that any retinal transform must retain enough information to distinguish different textures (as identified by human observers). Information which would distinguish equivalent textures must be suppressed or ignored.

from texture parameters.

Structural and statistical approaches to texture description have been proposed. Structural methods first locate primitive regions, then analyze the spatial relationships. This requires that the texture have identifiable primitives, and that the vision system be able to determine which primitives are present. This is probably the correct way to analyze regions and objects, but is too knowledge-dependent for a preliminary texture segmentation system. The most promising general approaches measure the relationships between edge elements or small regions. This is similar to the statistical approach described below.

Texture is both spatial and statistical. It is spatial since texture is the relationship of groups of picture elements. Nothing can be learned about texture from an isolated pixel, or from a histogram of pixel values. Monotonic transformations leave texture largely unchanged. This is why we are able to use histogram equalization.

Texture perception is largely unchanged by random variation in the shape, orientation, structure, or relative position of texture elements. While it is true that a highly regular texture can be disrupted by the introduction of a few irregularities, irregular textures are nearly immune to noise or variation. It seems that variability is an important texture dimension, and that changes in other texture measures must be interpreted relative to variability.

Statistical features are non-spatial. The most powerful and appropriate statistics for a particular type of texture are those which estimate parameters of the generating process. A general vision system, however, must use features which are common to many types of texture. One way to find such features is to model the human visual system. We have yet to develop a system which works as well. If we find such a system, however, we can undoubtedly improve upon it for particular applications.

Natural texture dimensions can also be discovered by studying homogeneous texture fields. Each field contains variation inherent to that texture type. Different fields have different types of variation. Discriminant analysis is an appropriate tool for identifying the significant variations. It is only necessary that we propose a set of texture measures; the analysis determines which combinations are useful.

Traditional Texture Measures

There is good evidence that the human visual system does not respond to spatial dependencies of higher than second order. The relationship between any two pixels may be significant, but their joint relationship with any third pixel in an image field is not. This suggests the autocorrelation function as a matrix of texture descriptors. Unfortunately the autocorrelation function is very similar for most images. It is neccessary to consider large lag values before significant differences occur. These differences tend to be regularly spaced regions of high correlation energy corresponding to repetition frequencies within the texture fields. Such pockets of energy are not easy to identify and analyze. About the best which can be done cheaply is to describe the correlation function by its first few spatial moments. Clearly this method will have little power unless correlations are measured over very large windows. This would be inappropriate in image analysis, since relatively small regions of texture must be identified.

One way to gather the significant energy in the correlation function is to compute its Fourier transform. Equivalently, one may transform the original image window, discarding the phase information. Although this takes a large amount of computation, moving window transforms may be updated without too much trouble. The chief difficulty with transform methods is that they must be computed over from texture parameters.

Structural and statistical approaches to texture description have been proposed. Structural methods first locate primitive regions, then analyze the spatial relationships. This requires that the texture have identifiable primitives, and that the vision system be able to determine which primitives are present. This is probably the correct way to analyze regions and objects, but is too knowledge-dependent for a preliminary texture segmentation system. The most promising general approaches measure the relationships between edge elements or small regions. This is similar to the statistical approach described below.

Texture is both spatial and statistical. It is spatial since texture is the relationship of groups of picture elements. Nothing can be learned about texture from an isolated pixel, or from a histogram of pixel values. Monotonic transformations leave texture largely unchanged. This is why we are able to use histogram equalization.

Texture perception is largely unchanged by random variation in the shape, orientation, structure, or relative position of texture elements. While it is true that a highly regular texture can be disrupted by the introduction of a few irregularities, irregular textures are nearly immune to noise or variation. It seems that variability is an important texture dimension, and that changes in other texture measures must be interpreted relative to variability.

Statistical features are non-spatial. The most powerful and appropriate statistics for a particular type of texture are those which estimate parameters of the generating process. A general vision system, however, must use features which are common to many types of texture. One way to find such features is to model the human visual system. We have yet to develop a system which works as well. If we find such a system, however, we can undoubtedly improve upon it for large windows. Small window transforms reveal only high-frequency information, negating the theoretical justification of the transform. Further, single frequencies are seldom important or reliable. The spectrum may be reduced to a smaller number of features by computing the cepstrum, or Fourier transform of the spectrum. Another way to extract significant energy is to compute moments of the spectrum.

The co-occurrence matrix measures more general second-order properties. It is an estimate of the joint probability density function for pixels separated by a particular row and column shift. A diffierent matrix must be computed for each of several row and column shifts, although there is some reduction if rotational isotropy can be assumed. For texture segmentation by image classification, each of these matrices must be computed around each image pixel. It is not feasible to compute full 256x256 co-occurrence matrices for an 8-bit image. Images are typically requantized to 16 levels before joint probabilities are estimated. This leads to poor performance on low-contrast textures. The co-occurrence matrices must also be reduced to a reasonable number of features. This is best done by computing moments around the matrix diagonals. Many weighted moments have been suggested, but none has yet proven effective.

It has been seen that spatial moments are a good way to measure the distribution of energy in a correlation, spectral, or co-occurrence matrix. Spatial moments can also be used to measure texture directly, as described below.

Spatial Moments

Since texture is a locally spatial phenomenon, we must use local spatial operators to generate our feature planes. Computation of spatial moments is equivalent to multiplying an image window by a mask and then summing. This is exactly what is done in convolution. It seems reasonable to convolve small spatial moment masks with an image to produce a set of feature planes. Then statistical measures can be computed over large moving windows in each plane. These measures form the texture features for the point at the center of each large window.

The spatial moments of a local window are

$$M_{ij} = (1/N) \sum_{r} \sum_{c} r^{i}c^{j}I(r,c) \qquad (1)$$

where N is the number of pixels in the window, r and c are row and column indices, and I(r,c) is the image function. It is assumed that row and column indices are relative to the window center, and that the computed moments are assigned to this center point as a feature vector.

When spatial moments are computed over a probability density, such as a co-occurrence matrix, it is desirable to relate higher moments to the center of the probability mass, $(M_{10}/M_{00}, M_{01}/M_{00})$. For instance,

$$\begin{aligned} M'_{20} &= (1/N) \sum_{r} \sum_{c} (r-M_{10}/M_{00})^{2}I(r,c) \\ &= M_{20}-M_{10}^{2}/M_{00} \end{aligned} \qquad (2)$$

For small image windows, however, this standardization makes little difference. It is not worth the extra computation, and may not even be appropriate.

Other types of standardization may be more useful. We can make our texture measures invariant to the affine transformations $r' = sr + u$, $c' = tc + v$ by dividing the higher moments by the row and column standard deviations, $\sqrt{(M'_{20}/M_{00})}$ and $\sqrt{(M'_{02}/M_{00})}$. This removes the effect of camera zoom or texture coarseness. Other geometric large windows. Small window transforms reveal only high-frequency information, negating the theoretical justification of the transform. Further, single frequencies are seldom important or reliable. The spectrum may be reduced to a smaller number of features by computing the cepstrum, or Fourier transform of the spectrum. Another way to extract significant energy is to compute moments of the spectrum.

The co-occurrence matrix measures more general second-order properties. It is an estimate of the joint probability density function for pixels separated by a particular row and column shift. A diffierent matrix must be computed for each of several row and column shifts, although there is some reduction if rotational isotropy can be assumed. For texture segmentation by image classification, each of these matrices must be computed around each image pixel. It is not feasible to compute full 256x256 co-occurrence matrices for an 8-bit image. Images are typically requantized to 16 levels before joint probabilities are estimated. This leads to poor performance on low-contrast textures. The co-occurrence matrices must also be reduced to a reasonable number of features. This is best done by computing moments around the matrix diagonals. Many weighted moments have been suggested, but none has yet proven effective.

It has been seen that spatial moments are a good way to measure the distribution of energy in a correlation, spectral, or co-occurrence matrix. Spatial moments can also be used to measure texture directly, as described below.

Spatial Moments

Since texture is a locally spatial phenomenon, we must use local spatial operators to generate our feature planes. Computation of spatial moments is equivalent to multiplying an image window by a mask and then summing. This is exactly what is done in convolution. It seems reasonable to convolve small spatial moment masks with an image to produce a set of feature planes. Then statistical measures can be warps can be removed by standardizing with respect to the row and column correlation.

Rotational invariance can also be achieved. Suppose that the image texture has a dominant direction, such as a gradient or major Fourier component direction. Let the camera or texture field be rotated through an angle A, and let a = cos(A), b = sin(A). The new moments can be computed from the original window as

$$M_{ij}(A) = (1/N) \sum_{r} \sum_{c} (ar+bc)^{i}(-br+ac)^{j} I(r,c) \qquad (3)$$

Haralick computes several features of this form to measure energy along co-occurrence matrix diagonals. Using the binomial expansion it can be seen that these moments are linear combinations of the M_{ij}. For instance,

$$M_{11}(A) = -abM_{20}+(a^{2}-b^{2})M_{11}+abM_{02} \qquad (4)$$

Haralick and other investigators have also suggested that spatial moments be computed over non-linear functions of the co-occurrence probabilities. These can be duplicated as closely as desired by combinations of the spatial-statistical moments to be introduced later.

Statistical Moments

A texture field is an extended entity composed of repetitions of similar local primitives. We require, therefore, global measures of local properties. These global measures must be statistical since they must be shift-invariant and insensitive to random texture variations. They should also be easy to compute since large windows are involved.

The set of statistical moments are particularly good global measures. Consider a window placed on an image, or on any feature plane computed as a transform of the image. One likely texture measure is the average value of the feature within the window. Another is the standard deviation of the feature. Skewness and kurtosis are also good candidates, although somewhat harder to explain. It is known that the histogram of an 8-bit feature plane can be completely characterized by a set of 256 such statistics. Statistical moments above the fourth, however, are likely to be unreliable and to have little energy or importance. Initial results suggest that even the skewness and kurtosis are of little use.

The basic statistical moments of a window are

$$M_k = (1/N) \sum_r \sum_c I^k(r,c) \qquad (5)$$

It is convenient, however, to standardize the higher moments to remove the effect of mean and standard deviation. The statistics used in our study are of the form warps can be removed by standardizing with respect to the row and column correlation.

Rotational invariance can also be achieved. Suppose that the image texture has a dominant direction, such as a gradient or major Fourier component direction. Let the camera or texture field be rotated through an angle A, and let a = cos(A), b = sin(A). The new moments can be computed from the original window as

$$M_{ij}(A) = (1/N) \sum_{r} \sum_{c} (ar+bc)^{i}(-br+ac)^{j} I(r,c) \tag{3}$$

Haralick computes several features of this form to measure energy along co-occurrence matrix diagonals. Using the binomial expansion it can be seen that these moments are linear combinations of the M_{ij}. For instance,

$$M_{11}(A) = -abM_{20}+(a^2-b^2)M_{11}+abM_{02} \tag{4}$$

Haralick and other investigators have also suggested that spatial moments be computed over non-linear functions of the co-occurrence probabilities. These can be duplicated as closely as desired by combinations of the spatial-statistical moments to be introduced later.

Statistical Moments

A texture field is an extended entity composed of repetitions of similar local primitives. We require, therefore, global measures of local properties. These global measures must be statistical since they must be shift-invariant and insensitive to random texture variations. They should also be easy to compute since large windows are involved.

$$AVE = M_1 \qquad (6a)$$

$$SDV = \sqrt{(M_2-M_1{}^2)} \qquad (6b)$$

$$SKW = (M_3-3(M_1*M_2)+2(M_1{}^3))/SDV^3 \qquad (6c)$$

$$KRT = [(M_4-4(M_1*M_3)+6(M_1{}^2*M_2-3(M_1{}^4))/SDV^4]-3.0 \qquad (6d)$$

The kurtosis has been reduced by 3.0 so that a Gaussian distribution will have zero skewness and kurtosis.

Computation of the four moments at every picture point can be done in a single pass. On our PDP-10KL this takes two minutes for a 512x512 image, regardless of the moving window size. The number of image rows which must be kept in core is equal to the number of rows in the window. Each pixel is examined only twice, once as it enters the moving window and once as it leaves.

Experimental Results

Two sets of textures have been used to test the discriminating power of these features. The first set consists of the Brodatz pictures of grass, raffia, sand, and wool. These pictures were chosen for their strong, uniform structures and for their similarity. They have been made more similar by histogram equalization. The second set includes the first and the Brodatz pictures of bark, straw, herringbone cloth, pressed calf leather, water, wood, and plastic bubbles, each histogram equalized.

Feature vectors are computed for 240 non-overlapping 32x32 windows in each picture. These vectors are then passed to the SPSS analysis system, where discriminant analysis is performed. The analysis involves stepwise inclusion and deletion of features to identify significant eigen-dimensions in the feature space. Each feature is adjusted to have zero mean and unit standard deviation across the total population of sample vectors. This corresponds to a Fisher linear discriminant analysis. Classification functions are then computed from the principal texture axes. The accuracy of classification on the training set will be reported here as the primary quality measure for a set of texture features.

Co-occurrence features have been computed for the first texture set. These can be used as a reference for evaluation of later results. Let $P_{mn}M_{ij}$ be the moment M_{ij} computed on the joint probability matrix for row and column shifts m and n. These features have been computed for all subscript values 0, 1, and 2. All of the M_{00} features are equivalent, leaving 73 independent features. Computing time was approximately 4.5 minutes for 32x32 image blocks requantized to 32 levels.

Of the 73 features, only the M11 moments showed strong individual discriminating power. The strongest were $P_{22}M_{11}$ and $P_{02}M_{11}$. Discriminant analysis with all of the features identified two dominant texture dimensions. The first may be approximated by

$$-1.5P_{11}M_{21}+1.3P_{01}M_{12}-1.2P_{00}M_{22}-1.2P_{20}M_{12}+1.2P_{00}M_{22} \qquad (7a)$$

the second by

$$-2.4P_{10}M_{11}+1.9P_{11}M_{11}-1.7P_{01}M_{11}-1.5P_{10}M_{22}+1.5P_{20}M_{11} \qquad (7b)$$

Together these provide 98% classification accuracy, which seems to be about what a trained human observer could achieve on 32x32 blocks. The classification errors are mainly between grass and sand.

$$AVE = M_1 \tag{6a}$$

$$SDV = \sqrt{(M_2-M_1^{\,2})} \tag{6b}$$

$$SKW = (M_3-3(M_1*M_2)+2(M_1^{\,3}))/SDV^3 \tag{6c}$$

$$KRT = [(M_4-4(M_1*M_3)+6(M_1^{\,2}*M_2-3(M_1^{\,4}))/SDV^4]-3.0 \tag{6d}$$

The kurtosis has been reduced by 3.0 so that a Gaussian distribution will have zero skewness and kurtosis.

Computation of the four moments at every picture point can be done in a single pass. On our PDP-10KL this takes two minutes for a 512x512 image, regardless of the moving window size. The number of image rows which must be kept in core is equal to the number of rows in the window. Each pixel is examined only twice, once as it enters the moving window and once as it leaves.

Experimental Results

Two sets of textures have been used to test the discriminating power of these features. The first set consists of the Brodatz pictures of grass, raffia, sand, and wool. These pictures were chosen for their strong, uniform structures and for their similarity. They have been made more similar by histogram equalization. The second set includes the first and the Brodatz pictures of bark, straw, herringbone cloth, pressed calf leather, water, wood, and plastic bubbles, each histogram equalized.

Feature vectors are computed for 240 non-overlapping 32x32 windows in each picture. These vectors are then passed to the SPSS

The same spatial moment subroutine has been used to compute 3x3 spatial moments for the first texture set. It took ten minutes to extract 32x32 window statistics from the original image and the nine spatial moment feature planes. We believe that a moving-window update algorithm will reduce this to less than five minutes per 512x512 image. Using special techniques for low-order moments might cut this time in half.

The most significant spatial statistics are M_{11}SDV, M_{01}SKW, M_{11}KRT, and M_{21}SKW. Other strong measures are M_{10}SDV, M_{01}SDV, M_{01}KRT, M_{12}SDV, M_{21}SKW, M_{10}SKW, and M_{11}SKW. Conspicuously absent from this list are any of the AVE statistics. For 3x3 convolutions, at least, the convolution sum is not as important as its variability.

Two dominant texture dimensions were again found. The principle axes appear to be the same as found with co-occurrence statistics, although the second axis is reversed in sign. These components may be approximated by

$$.91M_{01}\mathrm{SDV}-.84M_{21}\mathrm{SDV}-.77M_{11}\mathrm{SDV} \tag{8a}$$

and

$$4.6M_{01}\mathrm{SDV}-4.1M_{21}\mathrm{SDV}+2.4M_{10}\mathrm{SDV}-2.4M_{12}\mathrm{SDV} \tag{8b}$$

Notice that only SDV features are important.

The classification accuracy using these principle components is 98%, the same as for co-occurrence statistics. The error pattern is slightly different, however. Grass and sand are still confused, but so are wool and sand. The errors are distributed more evenly.

Another approach is to substitute 3x3 statistical moments for the 3x3 spatial moments. We have computed these texture measures for both the first and second texture sets. The image was again sampled at 240 non-overlapping 32x32 blocks. Computing time is 2.5 minutes for 240 non-overlapping 32x32 samples.

The strong features over the first set of four textures are SDVSDV, SKWSDV, SDVAVE, SDVSKW, and SDVKRT. (The local operation is mentioned first, then the global one.) Again the standard deviation features are dominant. Classification accuracy with these features is 87%. The principle components are

1.5AVESDV-1.5IMGSDV-.8SDVSDV

1.7IMGSDV-1.0SKWSDV

where IMG statistics alone have no classifying power because of the histogram equalization. The statistics do differ from window to window, however, and may be useful in combination with other features.

Over the full set of 11 textures the strong features are SDVAVE, SKWAVE, SDVSKW, and SDVKRT. SDVSDV, which is primarily an edge measure, is much less important than for the first set of textures. Classification accuracy drops to 84%, with the three dominant components

-1.5SDVAVE+.8IMGSDV-.7AVESDV

1.5IMGSDV-1.3AVESDV-1.3SDVAVE+1.1IMGAVE

-1.2IMGAVE-1.1SKWAVE

Notice the relative unimportance of the skewness and kurtosis features.

Variations of the local statistical features have also been tried. Adding the 3x3 Sobel edge detector increases classification accuracy on the first set of textures to 99%. SBLSDV and SBLAVE are

The same spatial moment subroutine has been used to compute 3x3 spatial moments for the first texture set. It took ten minutes to extract 32x32 window statistics from the original image and the nine spatial moment feature planes. We believe that a moving-window update algorithm will reduce this to less than five minutes per 512x512 image. Using special techniques for low-order moments might cut this time in half.

The most significant spatial statistics are M_{11}SDV, M_{01}SKW, M_{11}KRT, and M_{21}SKW. Other strong measures are M_{10}SDV, M_{01}SDV, M_{01}KRT, M_{12}SDV, M_{21}SKW, M_{10}SKW, and M_{11}SKW. Conspicuously absent from this list are any of the AVE statistics. For 3x3 convolutions, at least, the convolution sum is not as important as its variability.

Two dominant texture dimensions were again found. The principle axes appear to be the same as found with co-occurrence statistics, although the second axis is reversed in sign. These components may be approximated by

$$.91M_{01}\mathrm{SDV}-.84M_{21}\mathrm{SDV}-.77M_{11}\mathrm{SDV} \tag{8a}$$

and

$$4.6M_{01}\mathrm{SDV}-4.1M_{21}\mathrm{SDV}+2.4M_{10}\mathrm{SDV}-2.4M_{12}\mathrm{SDV} \tag{8b}$$

Notice that only SDV features are important.

The classification accuracy using these principle components is 98%, the same as for co-occurrence statistics. The error pattern is slightly different, however. Grass and sand are still confused, but so are wool and sand. The errors are distributed more evenly.

Another approach is to substitute 3x3 statistical moments for the 3x3 spatial moments. We have computed these texture measures for both the first and second texture sets. The image was again sampled at 240 both strong features. Normalizing each image block (to zero mean and unit standard deviation) prior to computing local features has little effect: classification accuracy drops to 98%. Use of 5x5 instead of 3x3 local statistics reduces accuracy to 82%, even with the 3x3 Sobel.

Spatial-Statistical Moments

The effectiveness of both spatial and statistical moments as local texture measures suggests the use of combined spatial-statistical moments. Let

$$M_{ijk} = (1/N) \sum_{r} \sum_{c} r^{i} c^{j} I^{k}(r,c) \tag{9}$$

This reduces to the spatial moments when k = 1 and to the statistical moments when i = j = 0. It may be, however, that the joint moments are more powerful local descriptors than the spatial and statistical features together. We are now setting up discriminant analyses to test this hypothesis.

2.3 Describing Natural Textures

Ramakant Nevatia, Keith E. Price and Felicia Vilnrotter*

Introduction

Many times, areas of an image are best characterized by their texture rather than purely intensity information. Texture is most easily described as the pattern of the spatial arrangement of different intensities (or colors). The different textures in an image

*Felicia Vilnrotter is supported by a Hughes Aircraft Company Doctoral fellowship.

are usually very apparent to a human observer, but automatic description of these patterns has proved to be very complex. We are concerned with a description of the texture which corresponds, in some sense, to a description produced by a person looking at the image.

Many statistical textural measures have been proposed in the past [1-4], therefore one can use some of their results indicating what measures may be useful. Among the statistical measures which have been discussed, and used, are analysis of the discrete Fourier transform to find indications of the structure [4], analysis of generalized gray-level co-occurence matrices [1], and analysis of the edges (or micro-edges) in a subwindow [3]. We are not interested in finding one texture measure which will distinguish between all regions (this is the ultimate, but extremely difficult problem) but in finding a texture measure to use in conjunction with many other features of the region [9].

The work in what can be called structural texture description has been more limited [5-7]. Maleson [5] used simple regions as the basic elements and used relations between regions and shape properties of the region in his analysis. Tamura et al. [6] tried to develop a set of operators which would rate textures on several scales, comparable to their ratings by human subjects. The proposals of Marr [7] for texture analysis based on the primal sketch are similar to some of the analysis which we perform.

Analysis of Texture

One of the most striking patterns seen in aerial images of a certain scale is the regular street or housing pattern of many cities (see Fig. 1). The appearance of this regularity is its most distinguishing characteristic, and because the pattern is so clear in the image it should be easy to extract. An obvious method to extract this regular pattern is the use of a 2-dimensional discrete Fourier transform. We computed this for various subwindows from the image in both strong features. Normalizing each image block (to zero mean and unit standard deviation) prior to computing local features has little effect: classification accuracy drops to 98%. Use of 5x5 instead of 3x3 local statistics reduces accuracy to 82%, even with the 3x3 Sobel.

Spatial-Statistical Moments

The effectiveness of both spatial and statistical moments as local texture measures suggests the use of combined spatial-statistical moments. Let

$$M_{ijk} = (1/N) \sum_{r} \sum_{c} r^{i} c^{j} I^{k}(r,c) \qquad (9)$$

This reduces to the spatial moments when k = 1 and to the statistical moments when i = j = 0. It may be, however, that the joint moments are more powerful local descriptors than the spatial and statistical features together. We are now setting up discriminant analyses to test this hypothesis.

2.3 Describing Natural Textures

Ramakant Nevatia, Keith E. Price and Felicia Vilnrotter*

Introduction

Many times, areas of an image are best characterized by their texture rather than purely intensity information. Texture is most easily described as the pattern of the spatial arrangement of different intensities (or colors). The different textures in an image

*Felicia Vilnrotter is supported by a Hughes Aircraft Company Doctoral fellowship.

Fig. 1 and other images (subwindows are given in Fig. 2). In the Fourier transform results shown in Fig. 3 there is some indication of the regular structure in the urban area windows, but it is not as apparent (visually) as it is in the image. Other attempts to derive much of the structural information from the Fourier transform were only partially successful [4], so we felt other methods should be attempted.

The individual textural elements could be located and analyzed [5], but the simple regions seem to be unreliable when the textural elements are very small, which is the case in the urban areas. Another option is to analyze an edge image to find the structure. The patterns in the original image will cause related patterns to appear in the edge image, and those patterns should be more consistent and easier to analyze than the original image data.

To study textures which are composed of small basic elements, a small window size edge detector must be used. We are interested in the edges between adjacent textural elements and not so much in edges between adjacent textural patterns. The edge operator which we use has been used successfully for other types of analysis [8]. The operator is applied over a 3 x 3 window and generates an edge magnitude and direction (1 of 8 directions). The direction is defined so that the brighter side is to the right when facing in the direction of the edge. Figure 4 shows the result of applying this operator to each of the subwindows in Fig. 2. The edge data must be further processed before it is in a form useable in texture analysis. Since an edge in the image appears as a broad peak in the edge detector output (the width in this case is two for a perfect step edge), the edges must be thinned. For the experiments here a simple non-maximal suppression was applied in 2 directions (horizontal and vertical), but a more sophisticated suppression which considers the directions of the edge elements could also be applied [8].

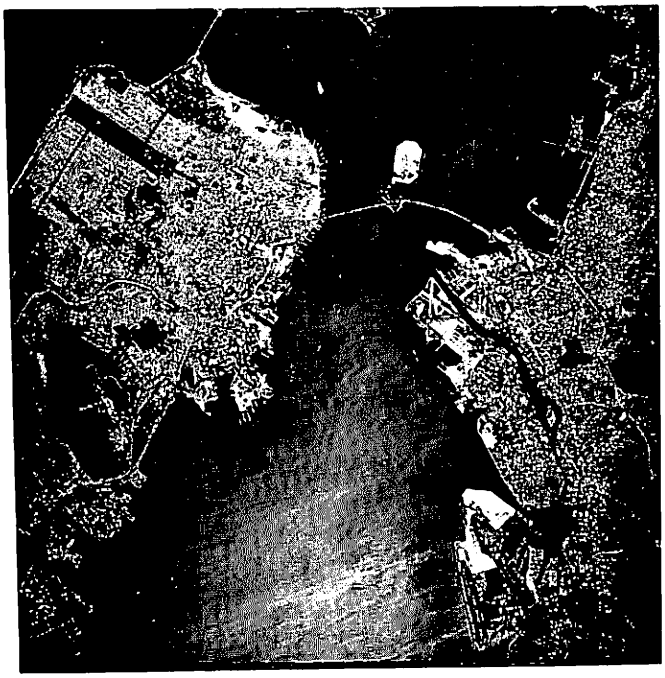


Fig.1. Aerial View of San Francisco Area.

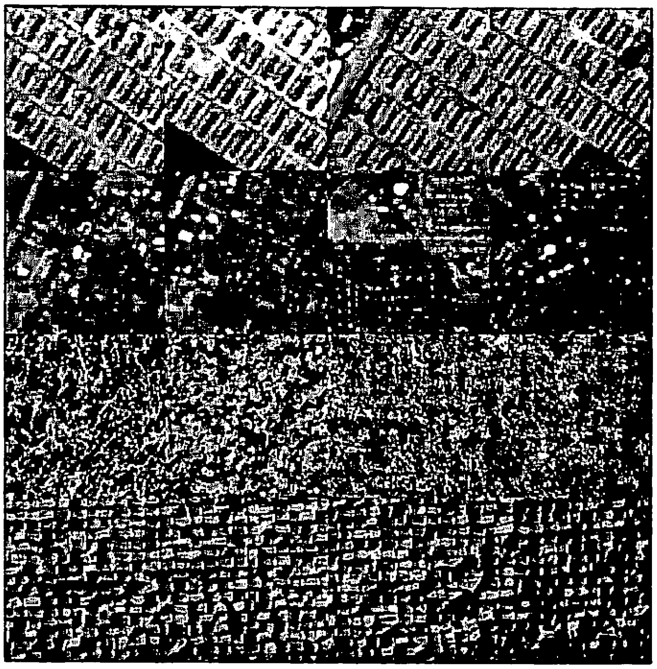


Fig. 2. 16 Subwindows for Texture Analysis

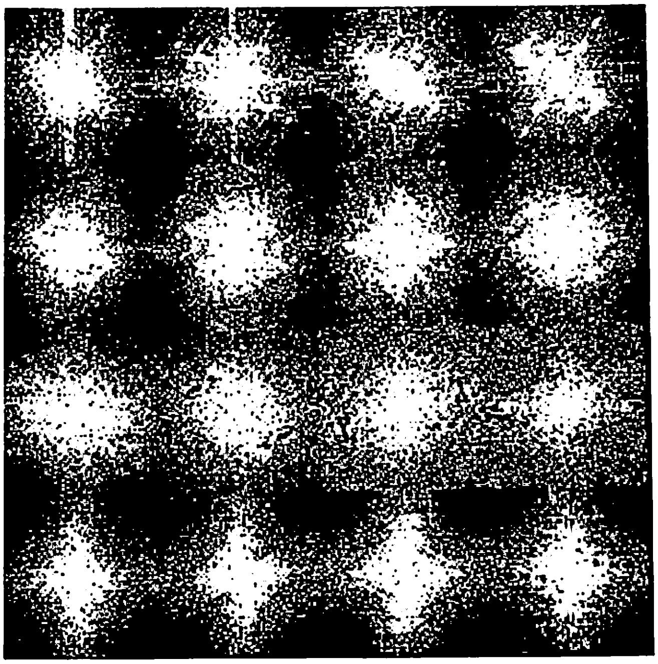


Fig. 3. Fourier Transforms of Subwindows in Fig. 2.

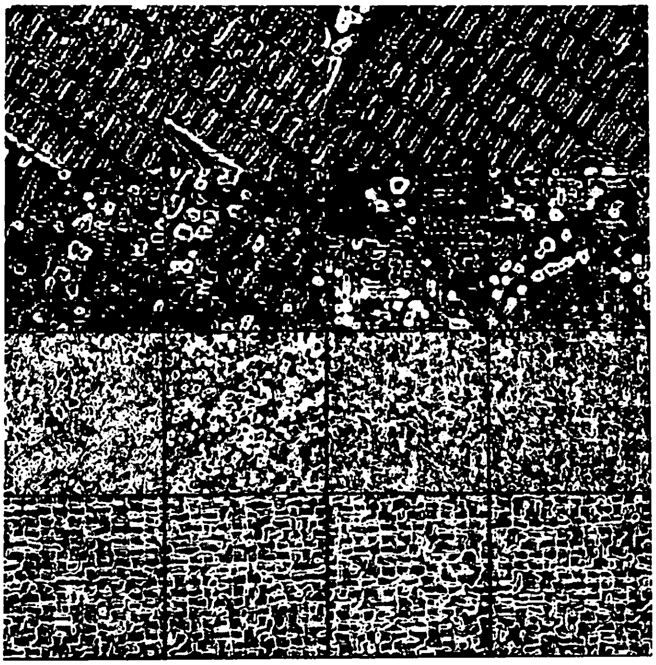


Fig. 4. Edge Magnitude for Subwindows in Fig. 2.

The suppressed edge images retain the regularity of the initial image, but now the regularity is in the spacing of edge elements not texture elements. A Fourier transform applied to this binary edge image would indicate the repetitive nature of the binary image, but is obscured by the degeneracies introduced by the binary nature of the input. Generalized gray level co-occurrence computations [1] have been studied for texture analysis, and were intended to indicate sizes of textural elements involved in the pattern. These can be applied more easily to a binary image than a general intensity image to indicate the spacing of edges.

Edge Co-occurrence Analysis

Generalized gray level co-occurrence matrix analysis is a basis for much of the statistical texture analysis. Basically, a set of matrices are computed for a portion of the image one for each selected spacing and angle. The entry in the matrix at row I and column J is incremented each time the first image point has the value I and the point at the given spacing and direction has the value J. Usually the image values are partitioned into a small set of values (8 rather than 256), so that it is even possible to compute the initial matrix. Also the computation is applied for many spacings (1,2,3,8, etc.) and several directions (0°,45°,90°,etc.) as shown in Fig. 6. Because of the large number of large matrices that are generated by this method various measures are computed on the matrix values, and the classification is performed using these measures [1]. The common and useful measures do not seem to capture the important feature in the edge images: the regular spacing of edge elements, but this is available in the co-occurrence matrix itself.

When binary edge images are used for co-occurrence analysis, many simplications in the computation can be made. We will use a 1 to indicate an edge at a given point, and 11 to indicate edges occurring at both the first point and the second point which is at some distance and angle from the first. The edge/no edge pair is indicated by 10

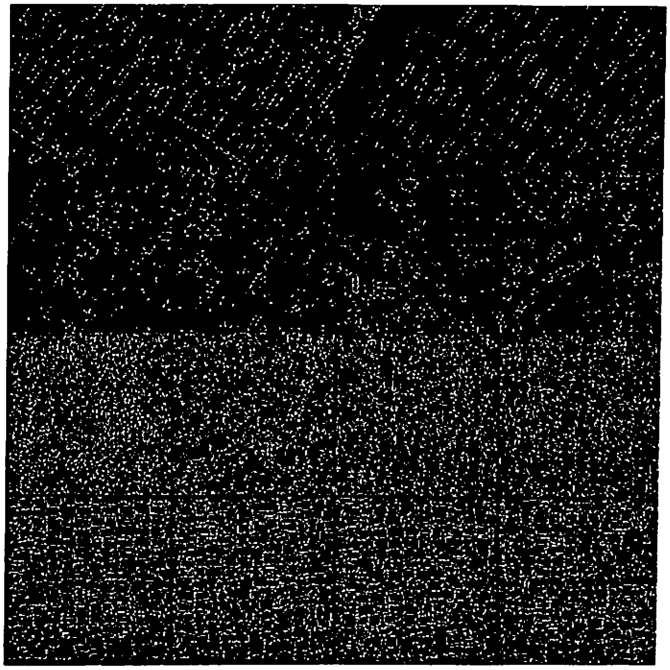


Fig. 5. Non-maximal Suppressed Edges from Fig. 4.

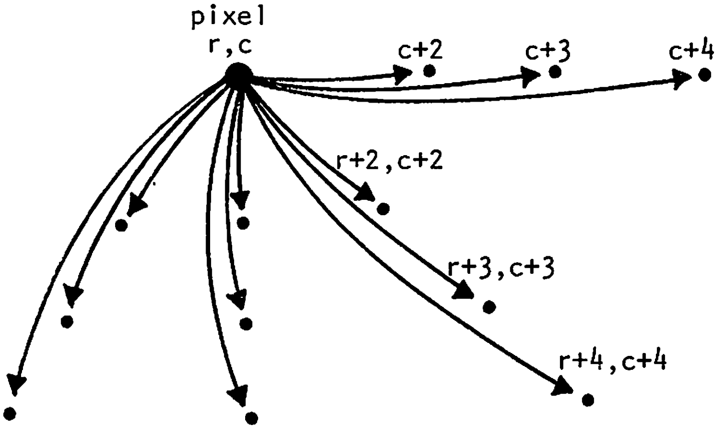


Fig. 6. Co-occurrence Matric Computation.

and no edge/edge by 01. Finally 00 means no edges at either point. The 10 and 01 combinations mean the same thing in terms of the image and thus are combined. The most important numbers are the 11 totals. The absolute magnitude is not very meaningful since this depends on the total number of edges and on the spacing being used (within a given image there are more opportunities for a co-occurrence edges with a small spacing than a large spacing) in addition to the actual frequency of occurrence of 11's. One good way to normalize the numbers seems to be to use the total of 10, 01, and 11. This gives the proportion of potential edges for co-occurrence that actually co-occur. We computed these values for 4 directions and spacings from 2 to 32 (at 45° and 135° a spacing of 2 is plotted at a distance of $2\sqrt{2}$). Some of these results are given in Fig. 7.

There are several ways to compare edges at two points, with different features indicated by the different comparison methods. Using all edges for every direction presents severe problems in the analysis of the output since long lines running in the same direction as the co-occurrence computation will be included along with lines running perpendicular to the direction. (Tamura et al. [6] used this feature to determine linear patterns in their texture experiments.) But, the edge element directions are availalbe and can be used to separate these two different patterns. The first step is to consider only those edge elements perpendicular to the direction of search, that is in the computation of co-occurrences in a horizontal direction only vertical edges are considered. There are an almost unlimited number of variations on this basic restriction which can either be derived from other variations or computed in a manner similar to the simple cases. The variations include: allow some freedom in the edge direction (45° either way), accept only perfect matches (up and up, down and down), accept only opposites (up and down, not up and up), and allow some freedom in the direction of the last two. The diferent combinations will all produce results with different information, so that several different ones can be computed.

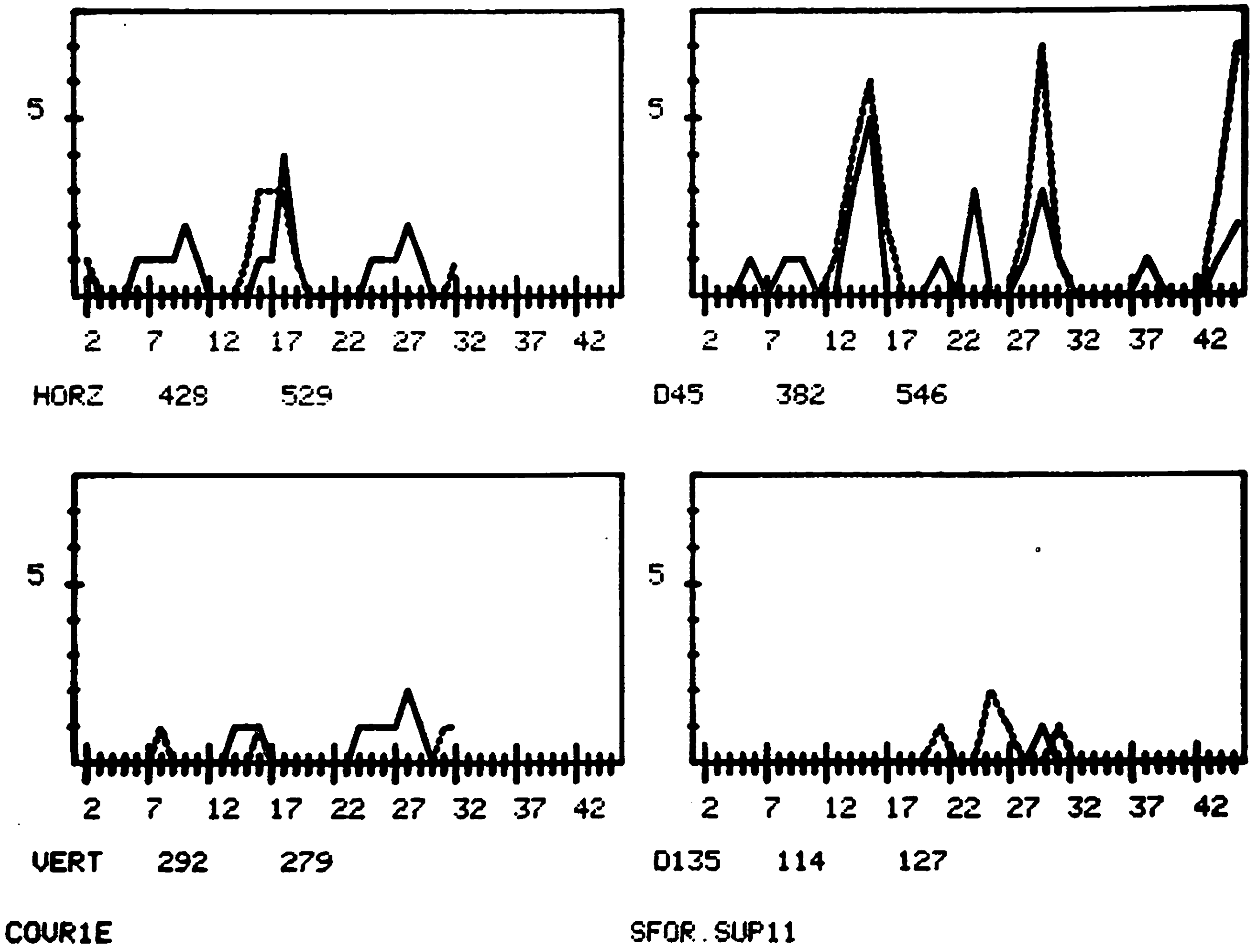
Discussion

None of this analysis would be worthwhile if it did not make the job of describing regular textures any easier. The highly regular patterns of the San Francisco urban area (the top row of Figs. 2-5 and Fig. 7a, 7b) and raffia (the bottom row and Fig. 7c, 7d) produce strong periodic patterns in the plot of the co-occurrence measure. A high value in the graphed measure indicates that edges frequently occur at that particular spacing. This spacing information can be used to determine the size and spacing of the textural elements, and the overall strength of the peak can be used to determine how regular the pattern is.

The spacing of pairs of textural elements is given by the peak to peak spacing using the measure which matches edges only in the exact same direction (as in Fig. 7a,c). The size of individual elements is best given by the measure which allows only edges in the opposite direction (as in Fig. 7b,d). The solid line in the graph indicates the size of dark objects and the dotted line the size bright objects. The size is from the first major peak, the succeeding peaks are caused by the repeated pattern. By comparing the results from the 4 directions, the orientation of the texture can be predicted. Since patterns usually do not line up with one of the 4 directions there will be some contribution to 2 of the directions. When these directions are 45^{o} apart the dominant direction is probably between them (as in San Francisco, Fig. 7a,b). But when they are 90^{o} apart there should be a regular pattern in two directions (as in Raffia, Fig. 7c,d). Thus, from the data we can say that the San Francisco subwindow has a regular pattern of bright and dark regions oriented in one direction, near 45^{o}, with the bright regions being larger (width about 10 pixels) than the dark ones (width about 4). Note that the size of the blocks in the other direction is near the size limit of the co-occurrence computation and also that very few of the edges at the ends of the blocks are detected.

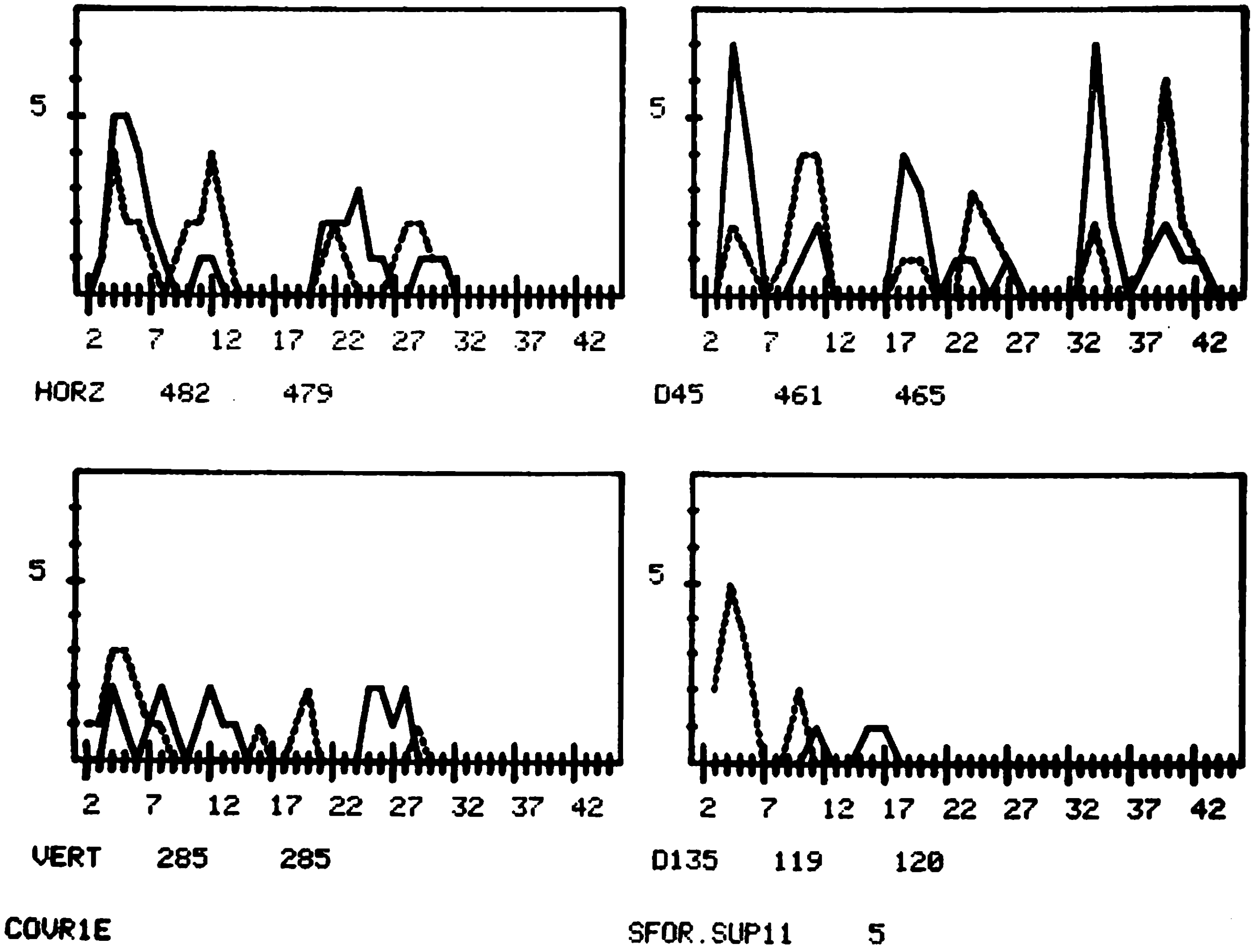
and no edge/edge by 01. Finally 00 means no edges at either point. The 10 and 01 combinations mean the same thing in terms of the image and thus are combined. The most important numbers are the 11 totals. The absolute magnitude is not very meaningful since this depends on the total number of edges and on the spacing being used (within a given image there are more opportunities for a co-occurrence edges with a small spacing than a large spacing) in addition to the actual frequency of occurrence of 11's. One good way to normalize the numbers seems to be to use the total of 10, 01, and 11. This gives the proportion of potential edges for co-occurrence that actually co-occur. We computed these values for 4 directions and spacings from 2 to 32 (at 45^{0} and 135^{0} a spacing of 2 is plotted at a distance of $2\sqrt{2}$). Some of these results are given in Fig. 7.

There are several ways to compare edges at two points, with different features indicated by the different comparison methods. Using all edges for every direction presents severe problems in the analysis of the output since long lines running in the same direction as the co-occurrence computation will be included along with lines running perpendicular to the direction. (Tamura et al. [6] used this feature to determine linear patterns in their texture experiments.) But, the edge element directions are availalbe and can be used to separate these two different patterns. The first step is to consider only those edge elements perpendicular to the direction of search, that is in the computation of co-occurrences in a horizontal direction only vertical edges are considered. There are an almost unlimited number of variations on this basic restriction which can either be derived from other variations or computed in a manner similar to the simple cases. The variations include: allow some freedom in the edge direction (45^{0} either way), accept only perfect matches (up and up, down and down), accept only opposites (up and down, not up and up), and allow some freedom in the direction of the last two. The diferent combinations will all produce results with different information, so that several different ones can be computed.



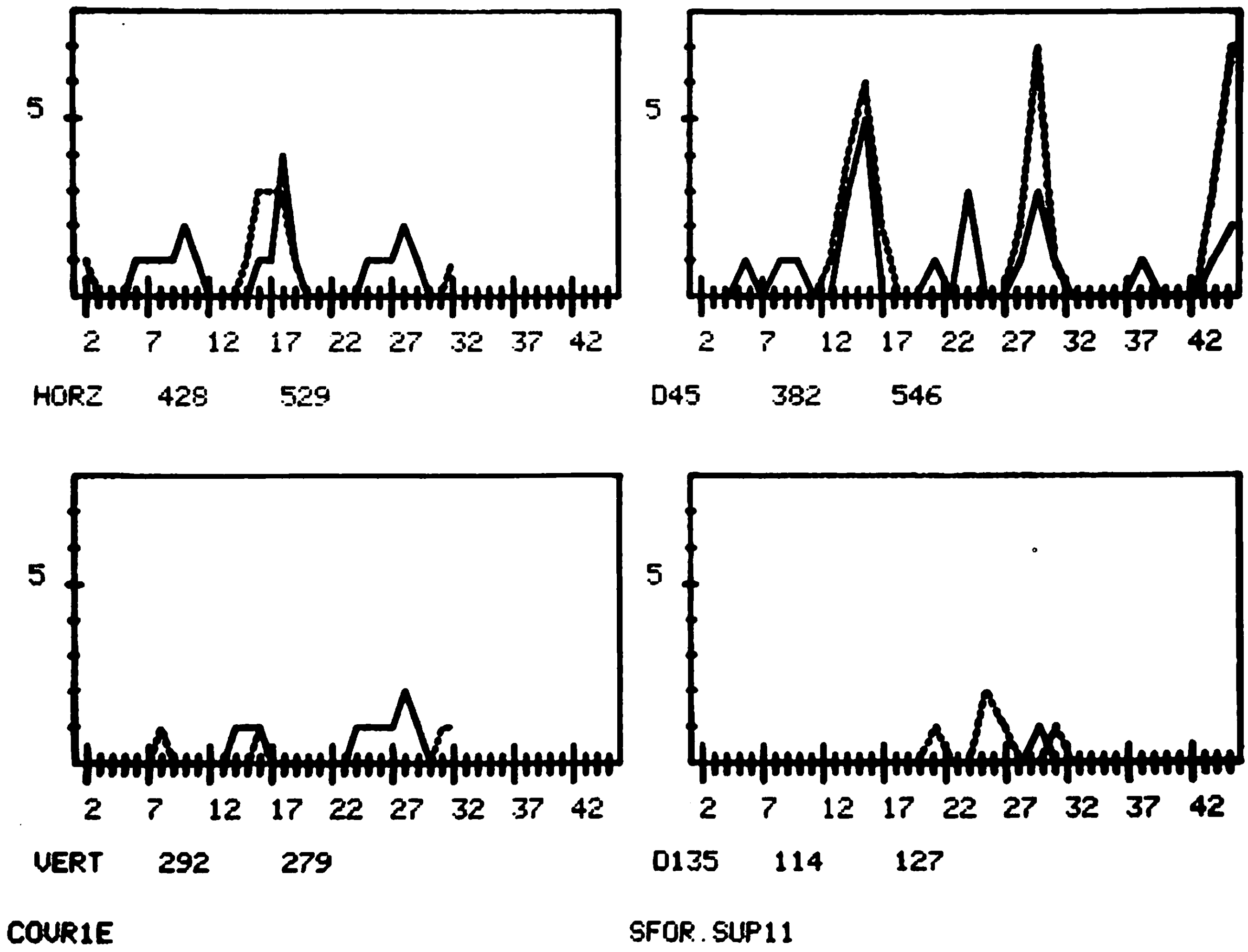
a) San Francisco Exact Edge Detection Matches Only.

Fig. 7. Co-occurrence Results Highly Regular Patterns.



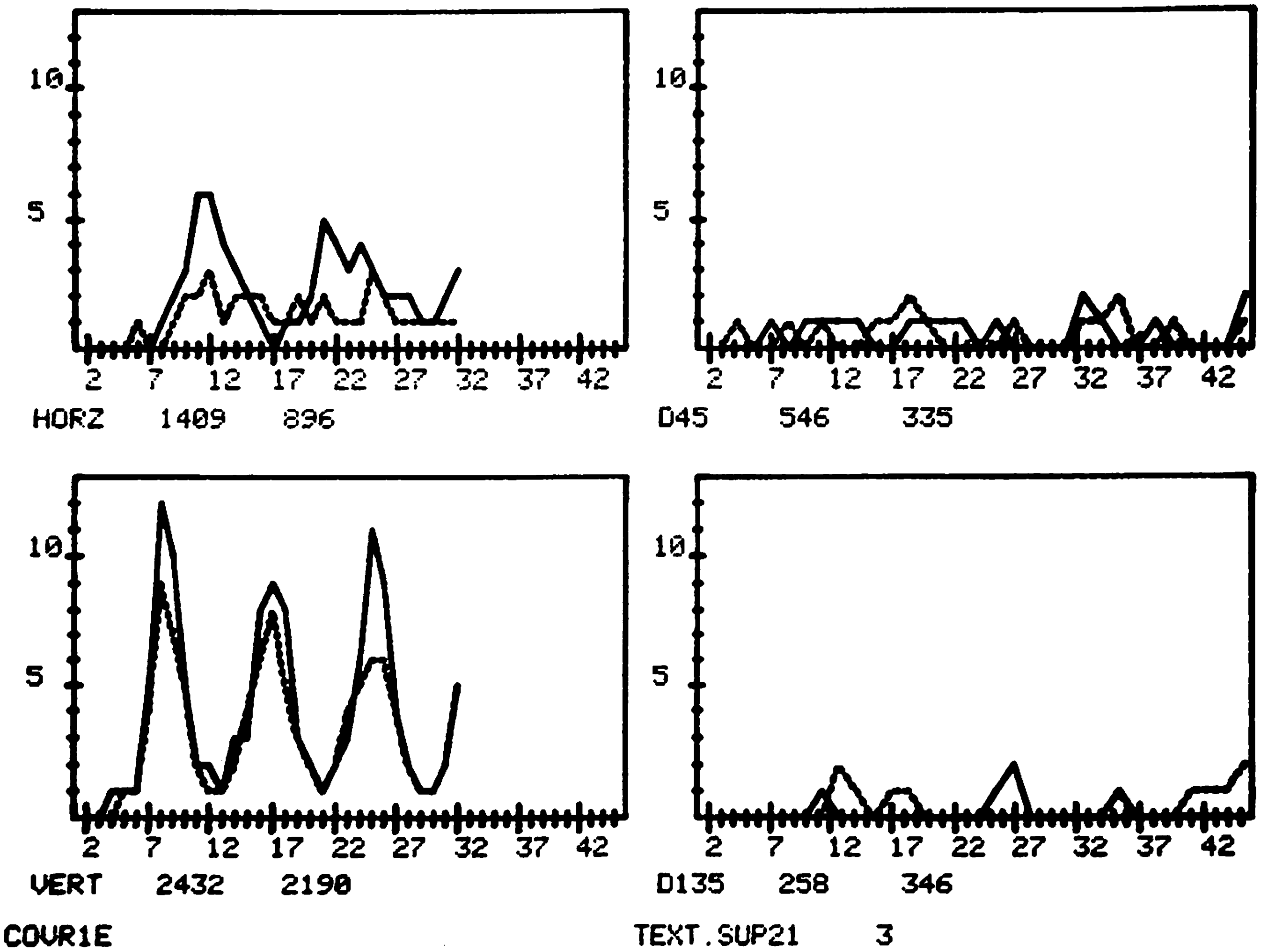
b) San Francisco Opposite Edge Direction Mathes Only.

Fig. 7. Continued.



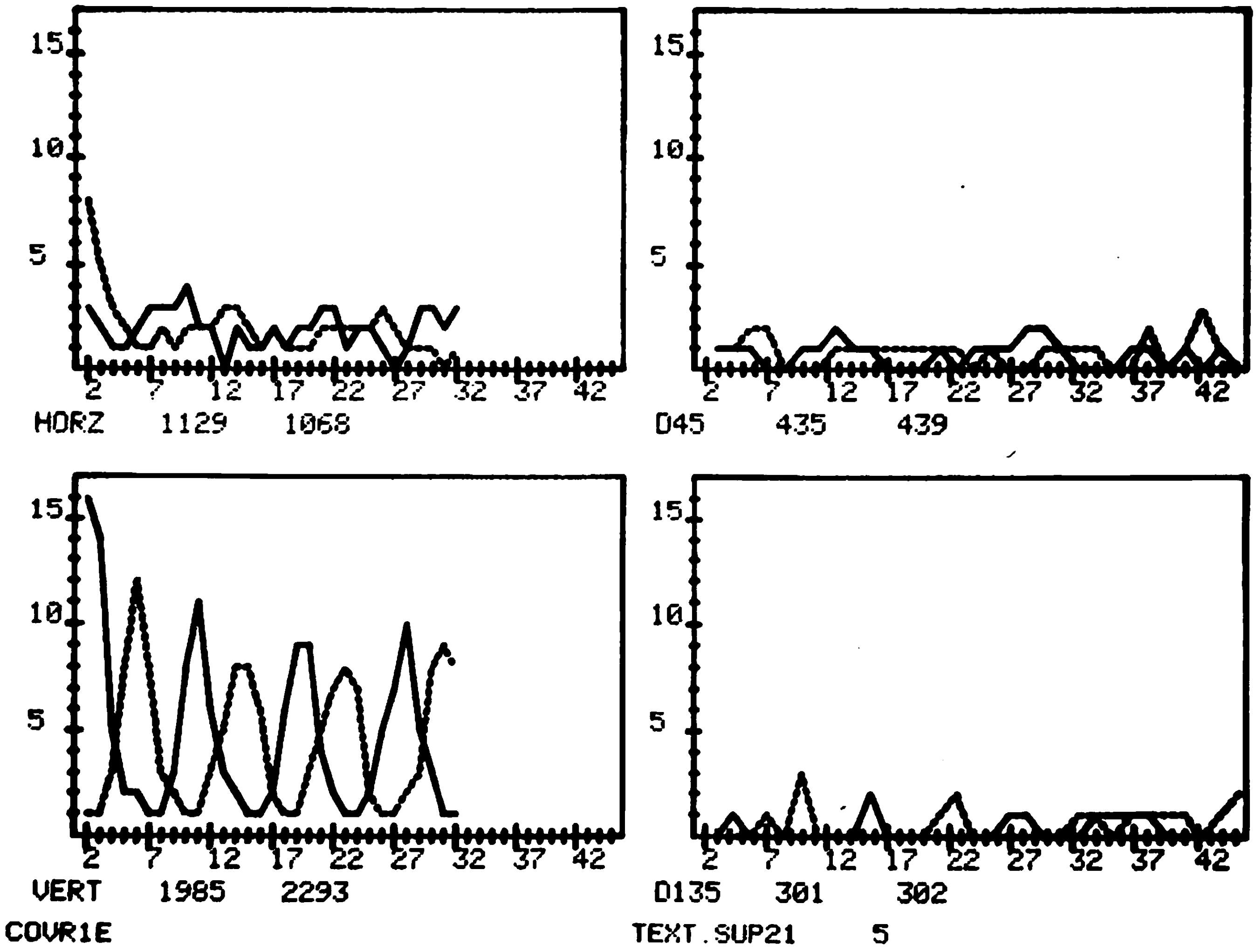
a) San Francisco Exact Edge Detection Matches Only.

Fig. 7. Co-occurrence Results Highly Regular Patterns.



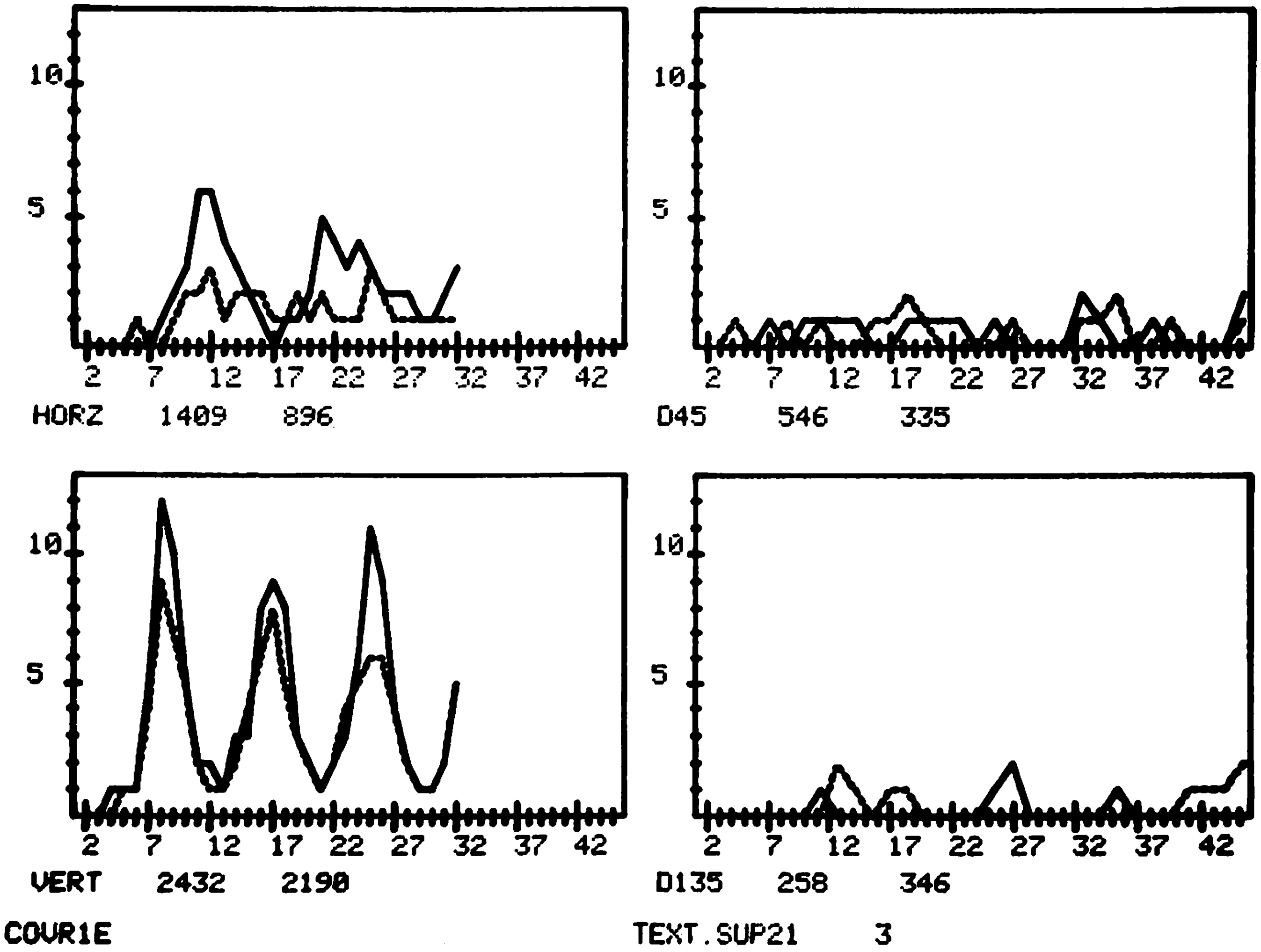
c) Raffia Exact Edge Direction Matches Only.

Fig. 7. Continued.



d) Raffia Opposite Edge Direction Matches Only.

Fig. 7. Continued.

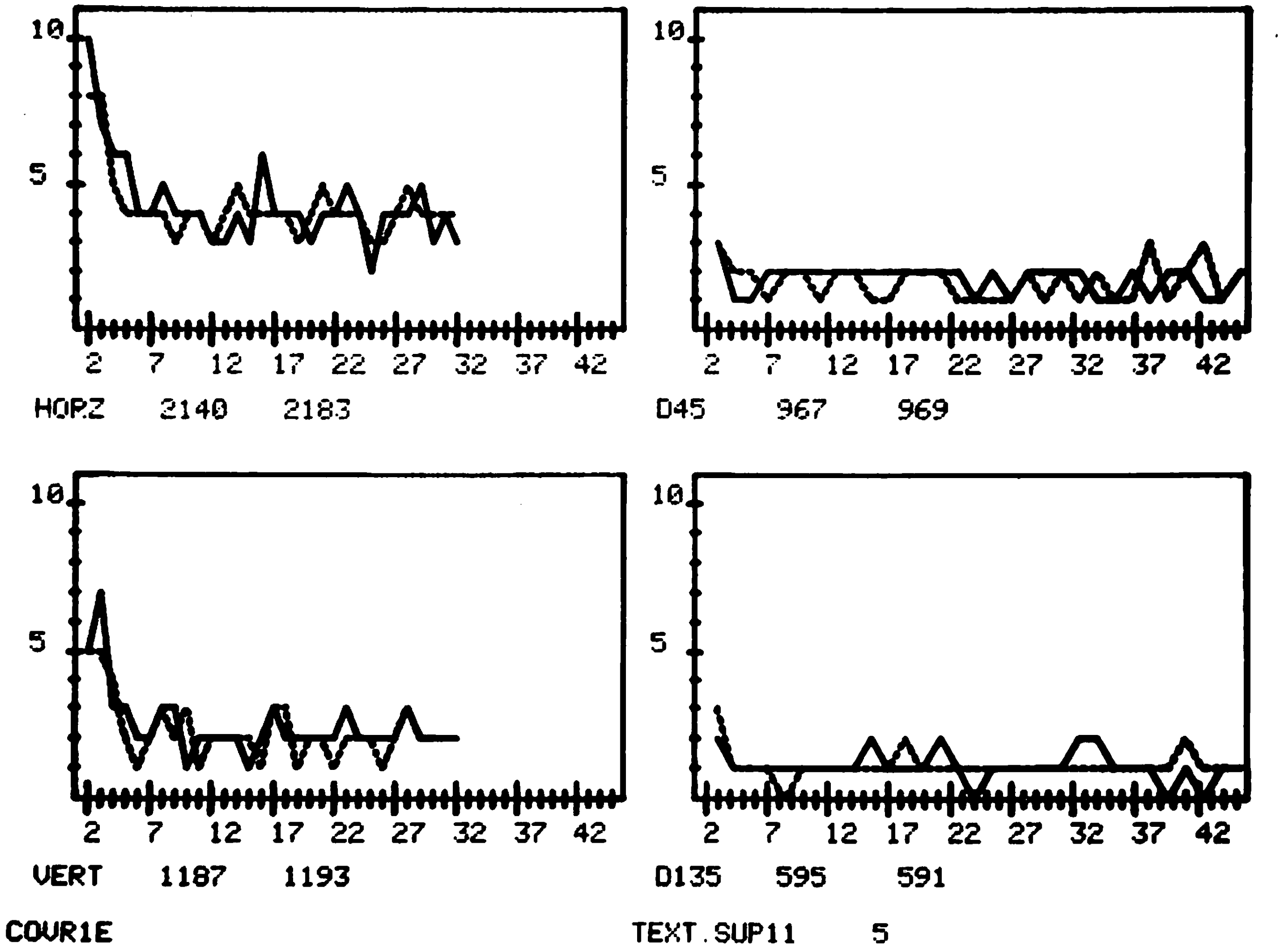


c) Raffia Exact Edge Direction Matches Only.

Fig. 7. Continued.

The irregular textural patterns (e.g.the suburban areas of the second row of Fig. 4, and the grass and sand of the third row, first and second windows) do not produce the same clearly periodic patterns of raffia as shown by Fig. 8a,b (for grass and suburban, respectively). But it is possible to derive certain useful features from these results, primarily that of the size of the textural elements. The strong peak near 3 for grass and 4 or 6 for suburban indicates a dominant size for textural elements (in the case of suburban probably 2 different sizes). The graphs indicate that the grass has thin dark and bright textural elements, predominately vertical and to a lesser extent, horizontal. The suburban area has only bright regions somewhat larger. These descriptions still leave open the question of whether the texturall elements are long and thin or small and round. The lack of a substantial peak in the 45^{0} or 135^{0} direction for grass indicates that it is probably long and thin and the small, though readily apparent peak in the graphs for the suburban windows indicates that the regions are probably small and round or more likely,rectangular).

Additional results are shown in Fig. 9-16. The first set are close up views of uniform texture patterns from [10], and the second set are aerial views of a variety of terrain. The results on these additional texture windows show that the procedure is robust enough to extract useful descriptors from a variety of textural patterns. There are some unexpected similarities in the case of a water and wood pattern (second and third rows in Figs. 9-11 and 12a,b). The edge images are very similar and there are only minor differences in the results of our processing. Structurally, the major differences between the two appears to be that the wood has its edge points in longer straight lines, a feature which may be derived through other processing. The herringbone (bottom row) cloth pattern is dominated by the regular pattern of the cloth and not the herringbone structure (Fig. 12c). The farmland (top row Fig. 13-15) does not have a repeated structure (at this resolution), so that the only results are the dominant sizes for the fields (i.e. the textural elements) as

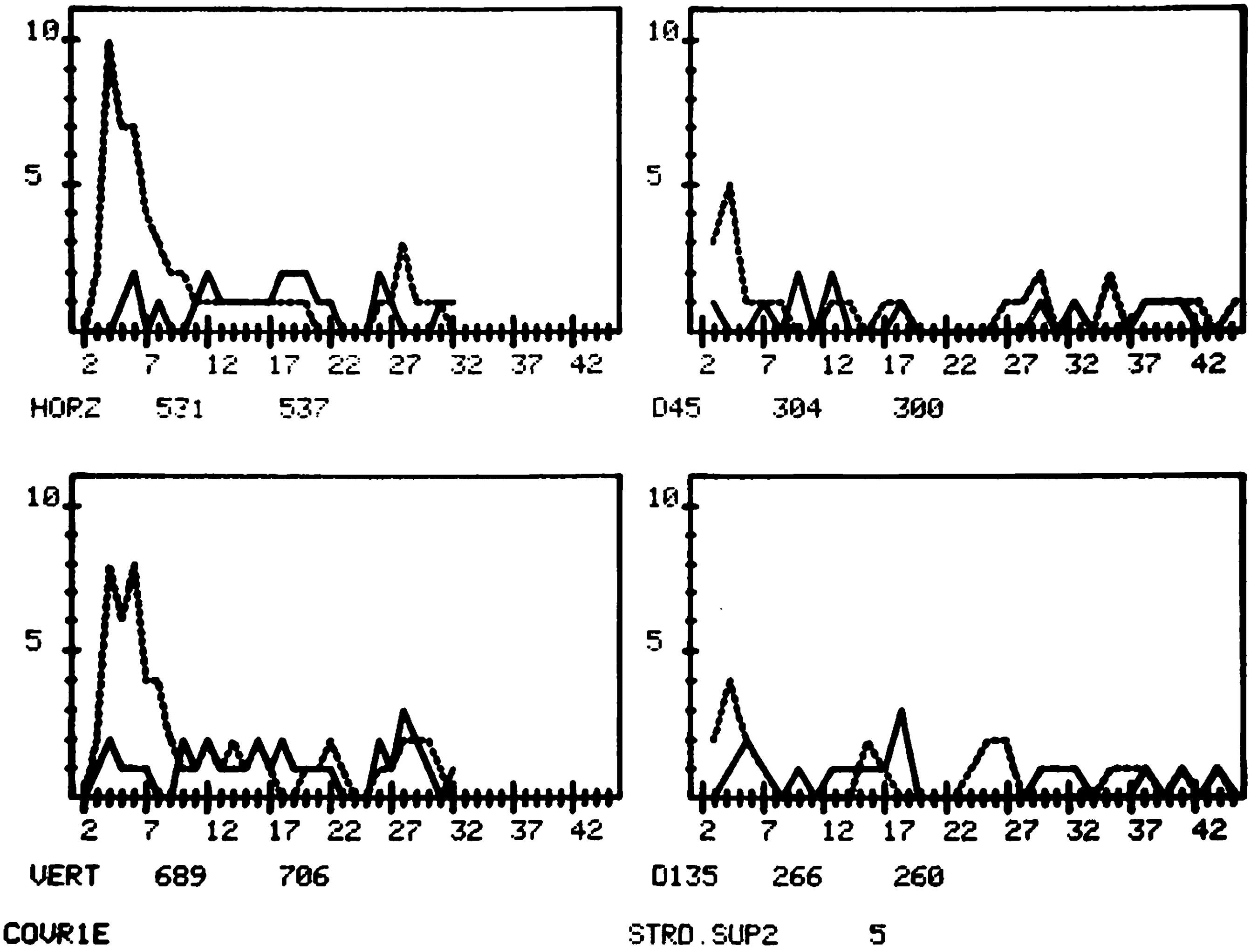


a) Grass Opposite Edge Direction Matches Only.

Fig. 8. Co-occurrence Results, Irregular Patterns.

The irregular textural patterns (e.g.the suburban areas of the second row of Fig. 4, and the grass and sand of the third row, first and second windows) do not produce the same clearly periodic patterns of raffia as shown by Fig. 8a,b (for grass and suburban, respectively). But it is possible to derive certain useful features from these results, primarily that of the size of the textural elements. The strong peak near 3 for grass and 4 or 6 for suburban indicates a dominant size for textural elements (in the case of suburban probably 2 different sizes). The graphs indicate that the grass has thin dark and bright textural elements, predominately vertical and to a lesser extent, horizontal. The suburban area has only bright regions somewhat larger. These descriptions still leave open the question of whether the texturall elements are long and thin or small and round. The lack of a substantial peak in the 45^{0} or 135^{0} direction for grass indicates that it is probably long and thin and the small, though readily apparent peak in the graphs for the suburban windows indicates that the regions are probably small and round or more likely,rectangular).

Additional results are shown in Fig. 9-16. The first set are close up views of uniform texture patterns from [10], and the second set are aerial views of a variety of terrain. The results on these additional texture windows show that the procedure is robust enough to extract useful descriptors from a variety of textural patterns. There are some unexpected similarities in the case of a water and wood pattern (second and third rows in Figs. 9-11 and 12a,b). The edge images are very similar and there are only minor differences in the results of our processing. Structurally, the major differences between the two appears to be that the wood has its edge points in longer straight lines, a feature which may be derived through other processing. The herringbone (bottom row) cloth pattern is dominated by the regular pattern of the cloth and not the herringbone structure (Fig. 12c). The farmland (top row Fig. 13-15) does not have a repeated structure (at this resolution), so that the only results are the dominant sizes for the fields (i.e. the textural elements) as



b) Suburban Opposite Edge Direction Matches Only.

Fig. 8. Continued.

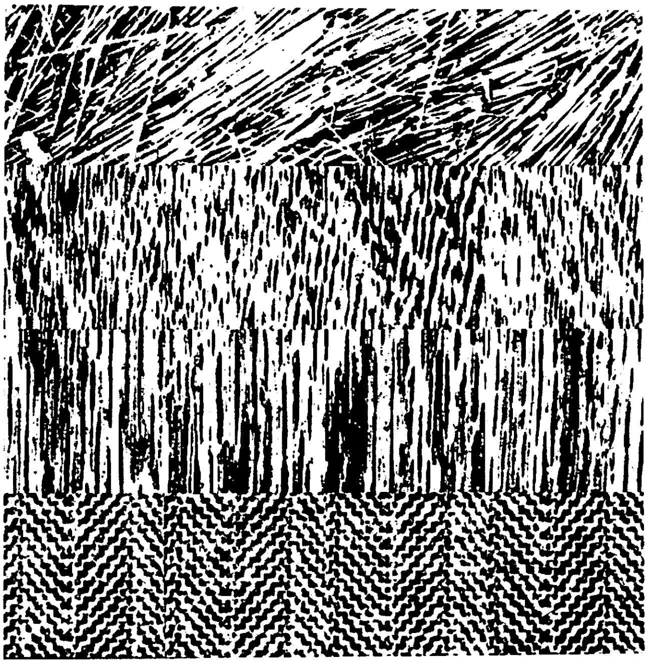


Fig. 9. Brodatz Texture Subwindows (Straw, Water, Wood, Herringbone).

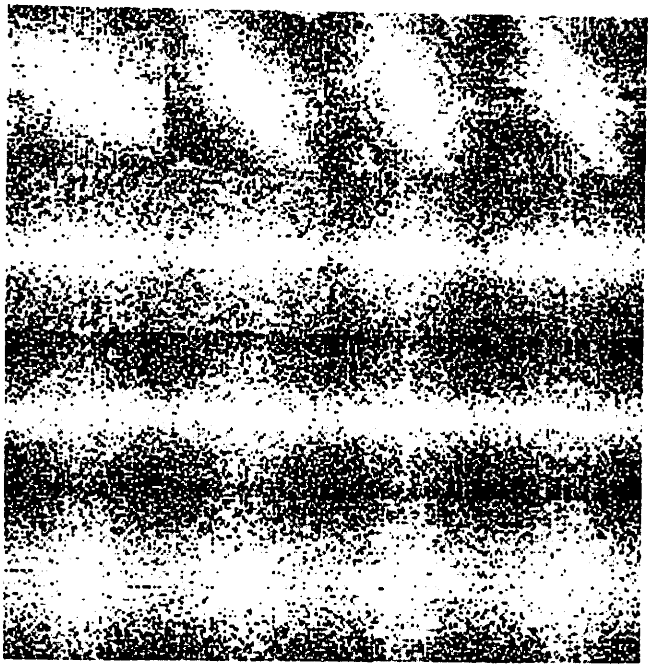


Fig. 10. Brodatz Texture Subwindows, Fourier Transform.

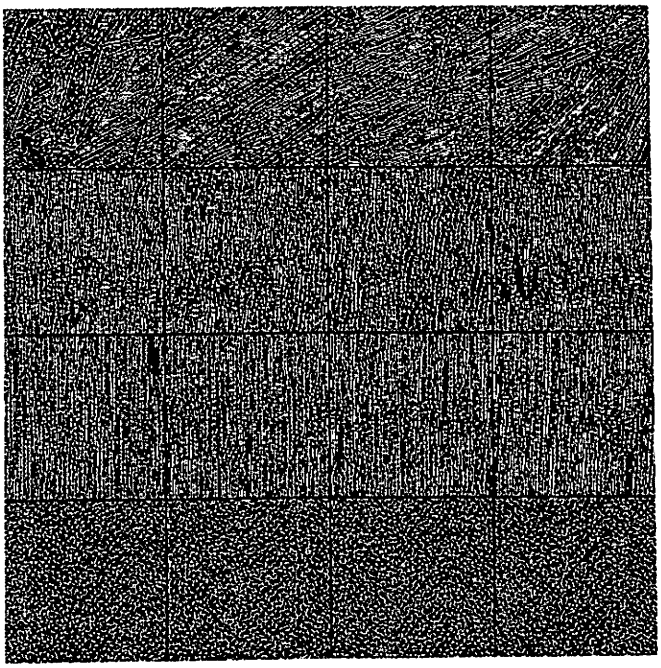
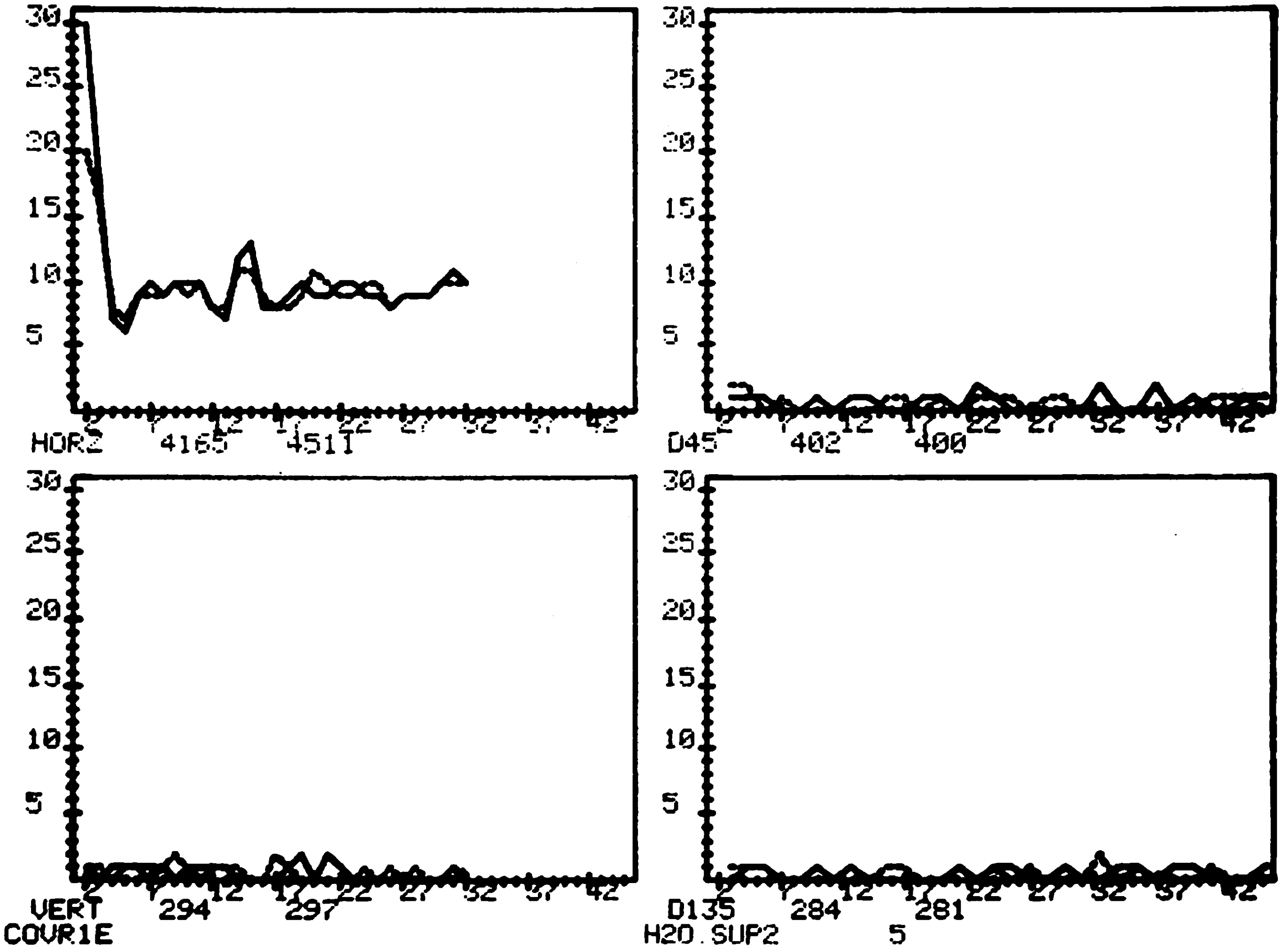
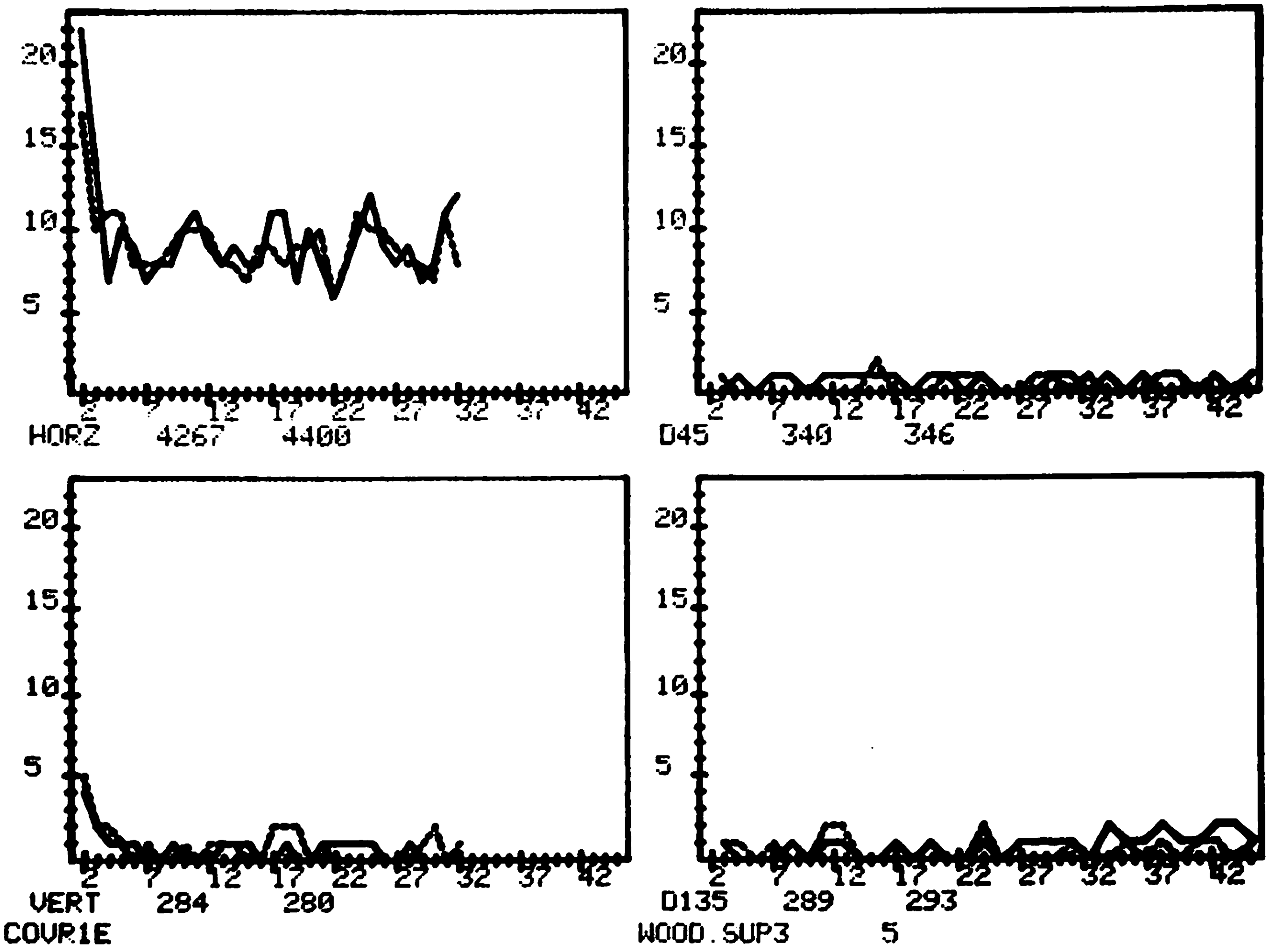


Fig. 11. Brodatz Texture Subwindows, Non-maximal Suppressed Edges.



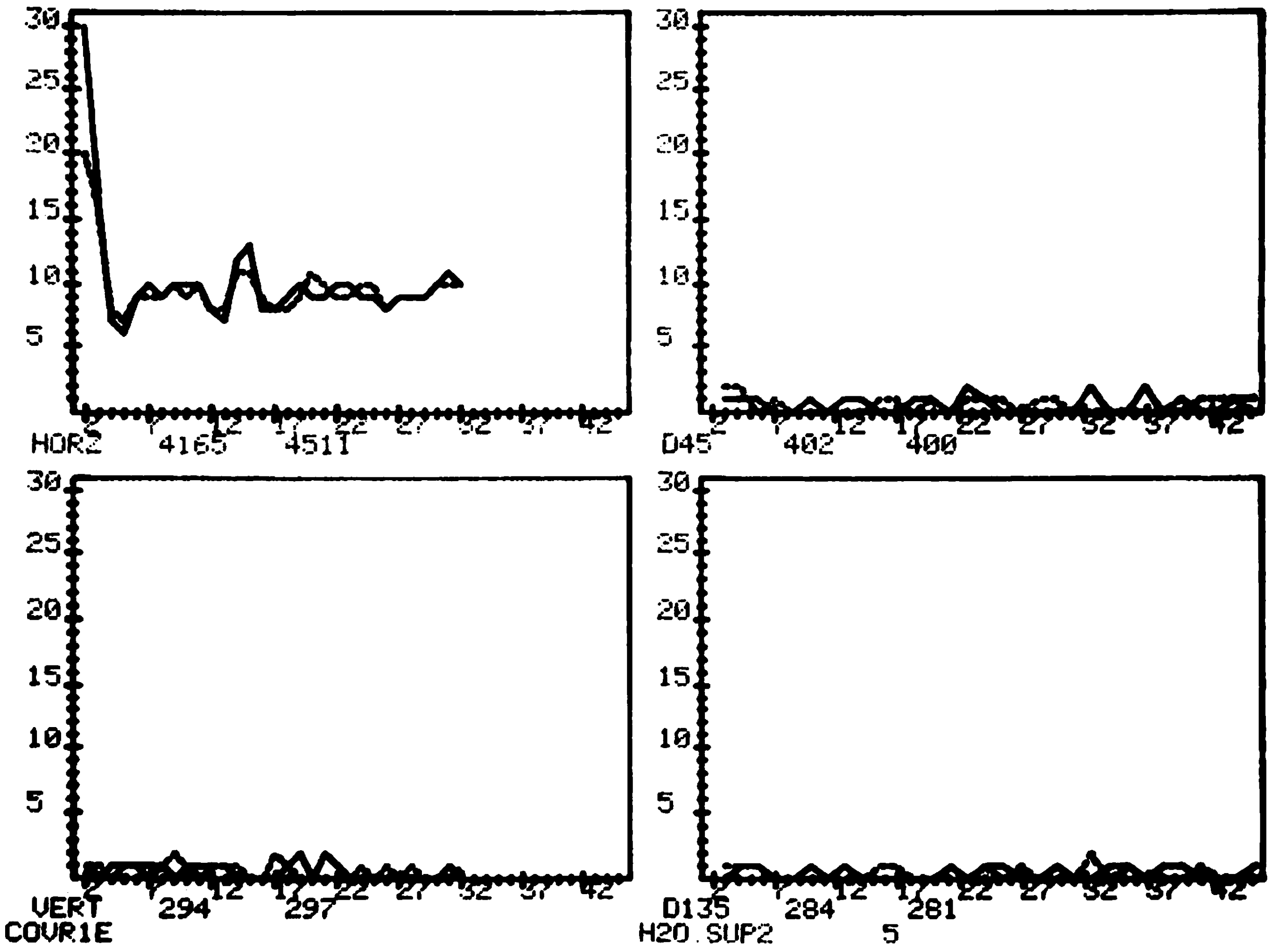
a) Water Opposite Edge Direction Matches Only.

Fig. 12. Brodataz Texture Subwindows, Co-occurrence Results.



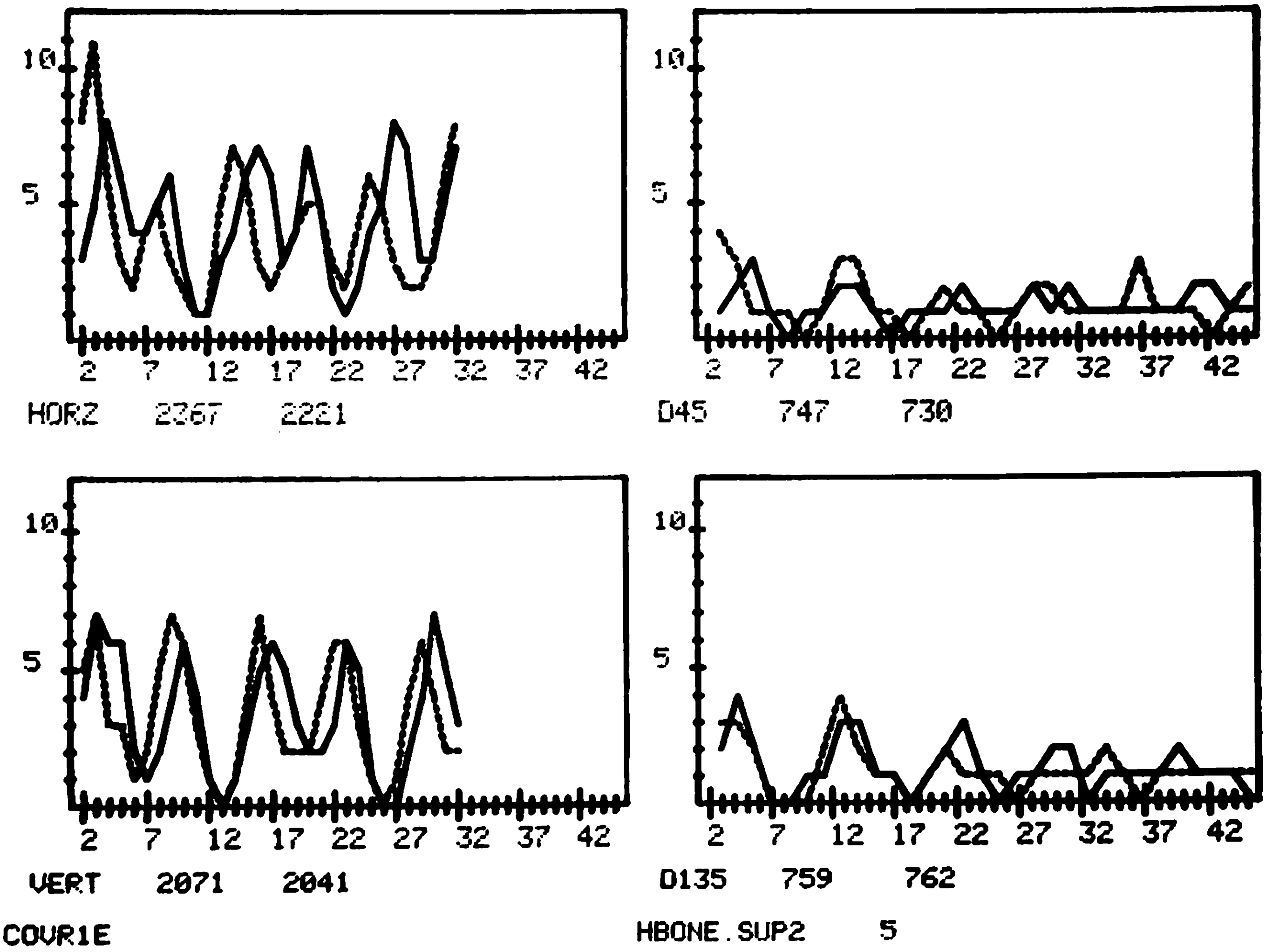
b) Wood Opposite Edge Direction Matches Only.

Fig. 12. Continued.



a) Water Opposite Edge Direction Matches Only.

Fig. 12. Brodataz Texture Subwindows, Co-occurrence Results.



c) Herringbone Material Opposite Edge Direction Matches Only.

Fig. 12. Continued.

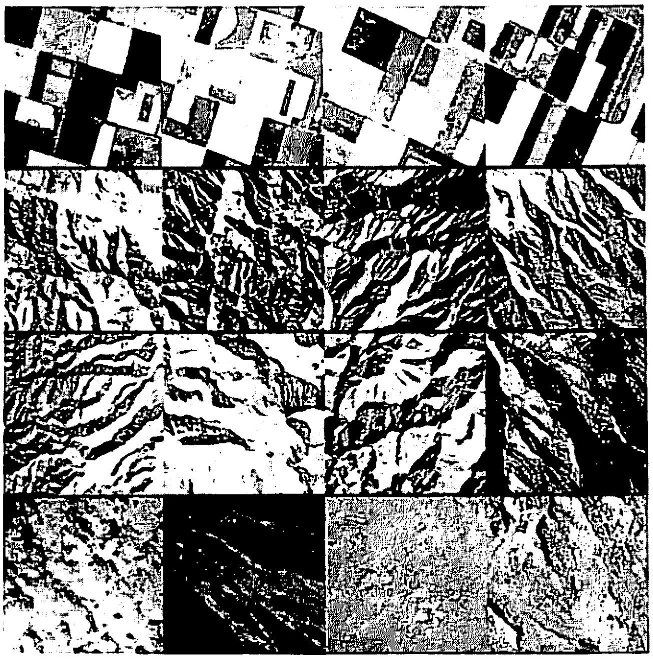


Fig. 13. Aerial Image Subwindows (Farm, Mountains,Desert and Clouds).

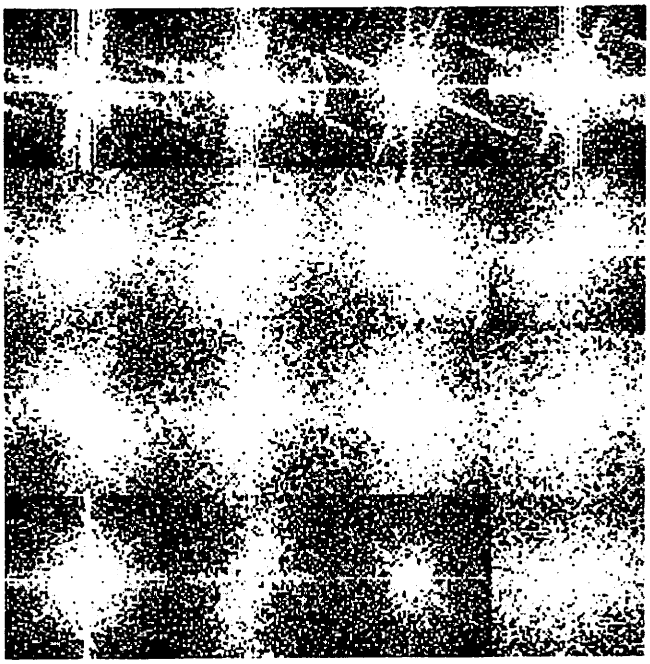


Fig. 14. Aerial Image Subwindows, Fourier Transform.

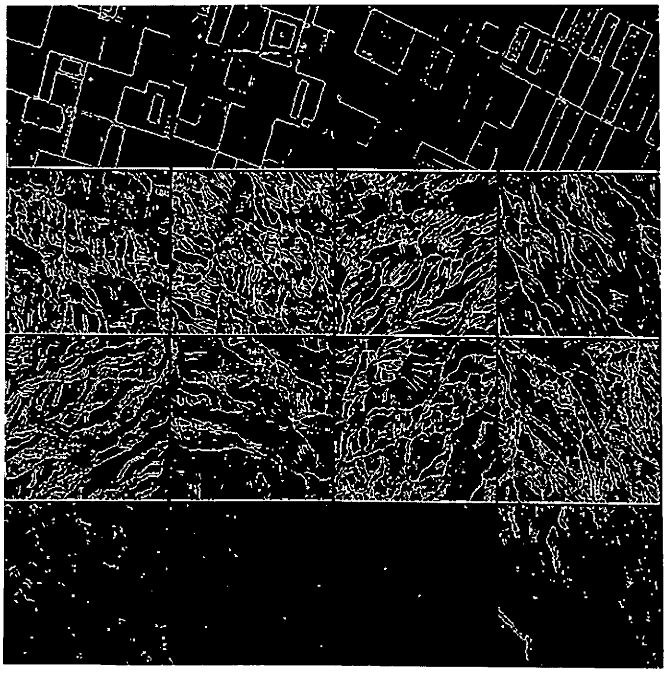
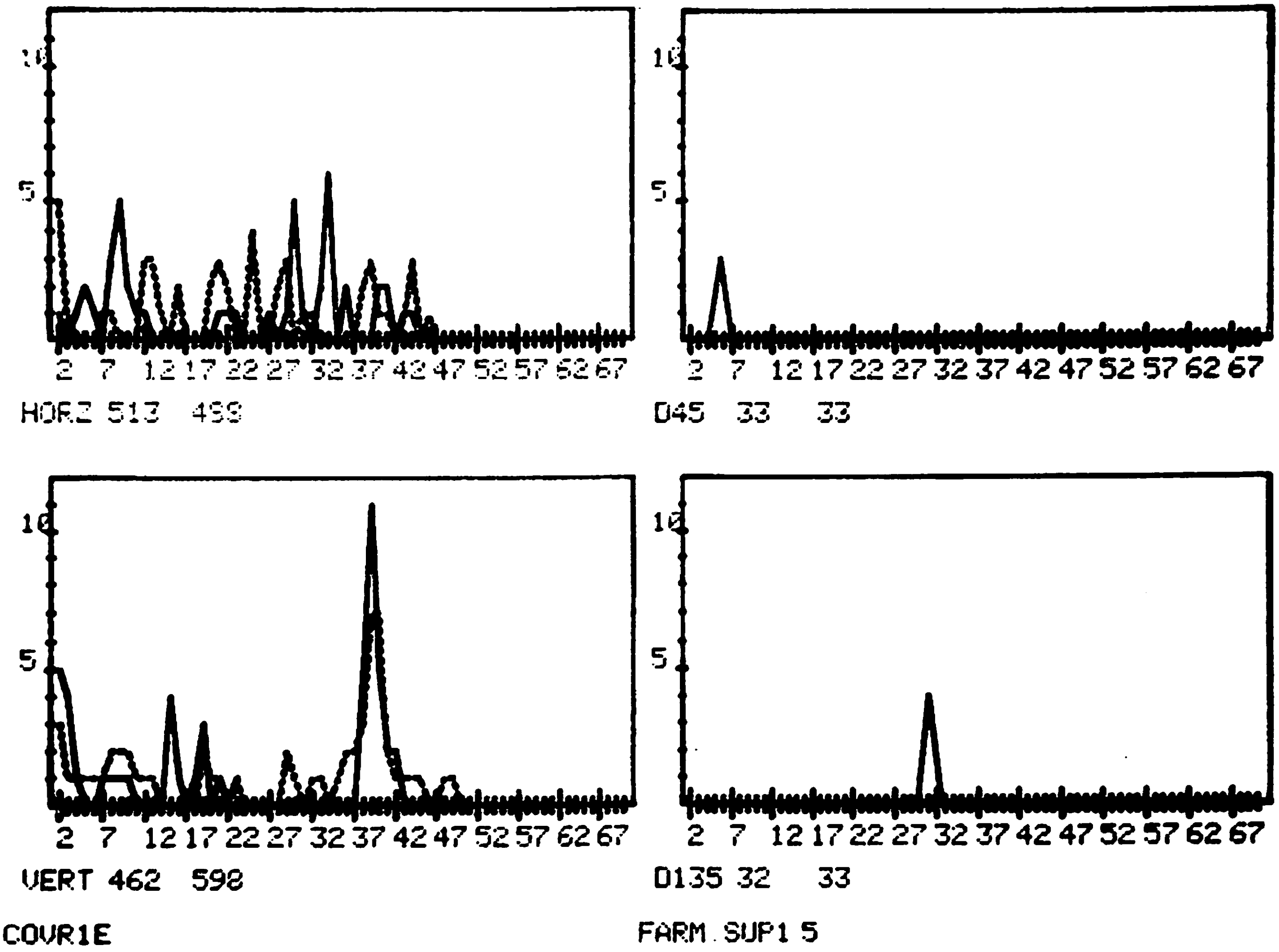
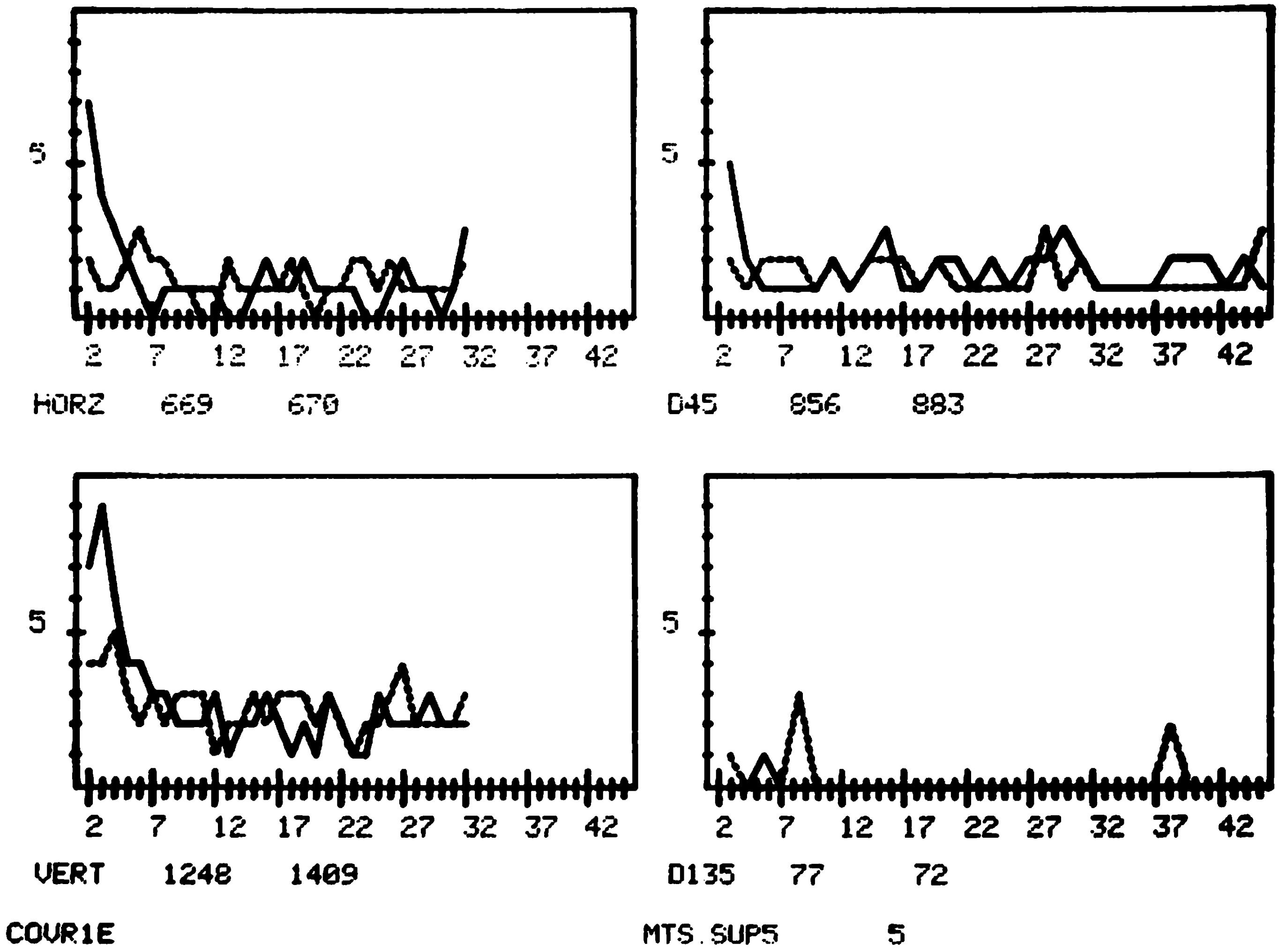


Fig. 15. Aerial Image Subwindows, Non-maximal Suppressed Edges.



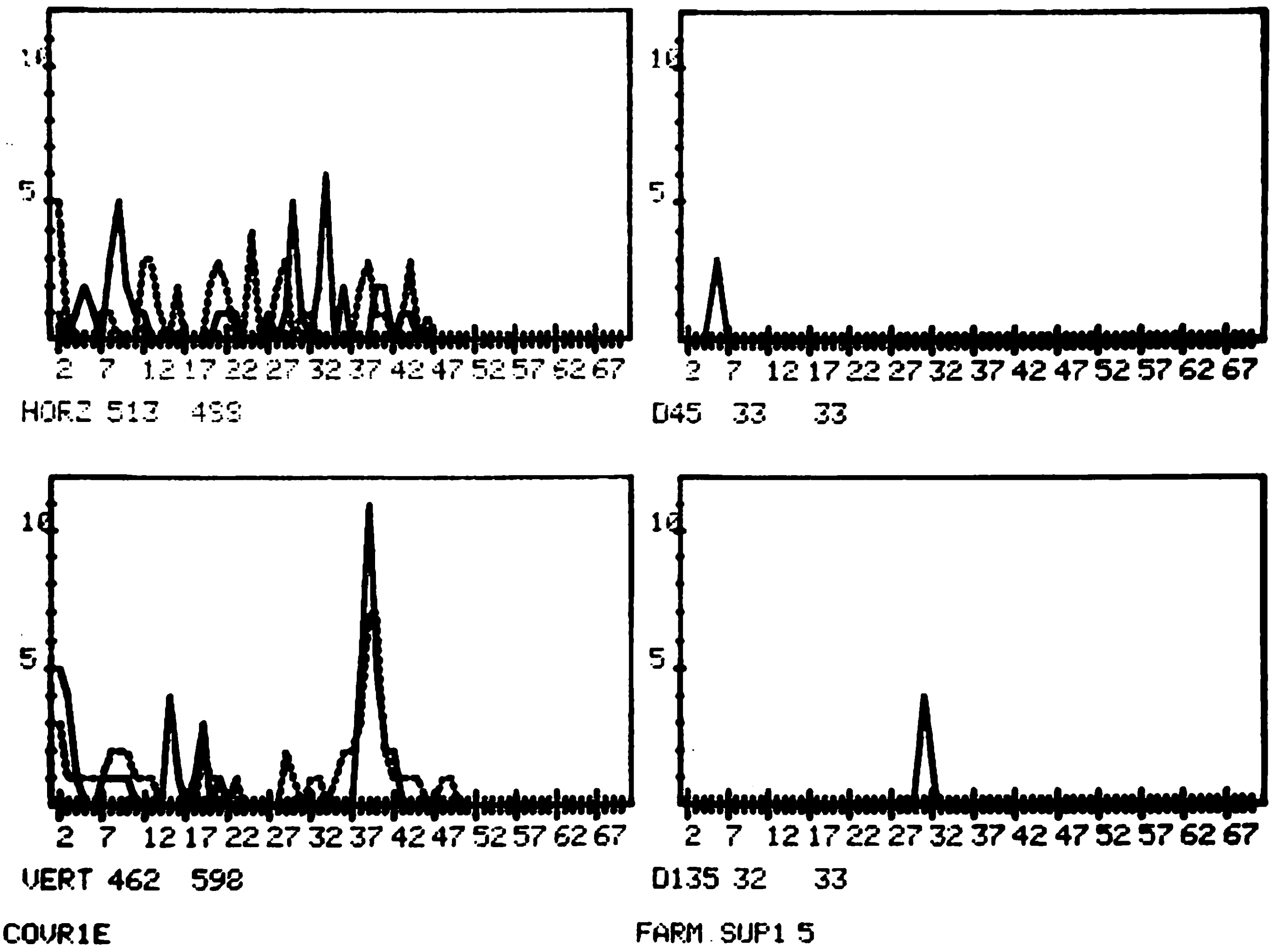
a) Farmland Opposite Edge Direction Matches Only.

Fig. 16. Aerial Image Subwindows, Co-occurrence Results.



b) Mountains Opposite Edge Direction Matches Only.

Fig. 16. Continued.



a) Farmland Opposite Edge Direction Matches Only.

Fig. 16. Aerial Image Subwindows, Co-occurrence Results.

shown in Fig. 16a. The mountains do not have a dominant size for the textural elements, except for some of the narrow dark (shadow) areas (Fig. 16b). The bottom row has so few edges that it can be simply described as a uniform region.

This is not a complete description of the textures, but serves as a good initial description of the patterns. There are still other important features of the textures which are not derived by this method, but could be computed by other techniques.

Conclusions

General texture analysis is a very difficult problem, but this analysis of edge images appears to be an effective method to extract many important structural features from the textural patterns. One major unanswered question is whether or not all of the information derived by the human user can be reliably derived by a program. We are still working on the automatic extraction of this information from the data which is produced by this textural analysis method.

References

1. R.M. Haralick, K. Shanmugam, and I. Dinstein, "Textural Features for Image Classification," IEEE Trans SMC-3, 1973, pp. 660-621.

2. J. Weszka, A. Rosenfeld, "A Comparative Study of Texture Measures for Terrain Classification," IEEE Trans SMC-6, April, 1976.

3. A. Rosenfeld and Kak, Digital Picture Processing, Academic Press: New York, 1976.

4. R. Bajcsy, "Computer Identification of Visual Surfaces," Computer Graphics and Image Processing-2, Oct. 1973, pp. 118-130.

5. J.T. Maleson, "Understanding Texture in Natural Images,"

University of Rochester, Ph.D. Thesis, to appear.

6. H. Tamura, S. Mori, T. Yamawaki, "Textural Features Corresponding to Visual Perception," IEEE Trans SMC-8, June, 1978, pp. 460-473.

7. D. Marr, "Early Processing of Visual Information," AI-M-340, Artificial Intelligence Laboratory, MIT, Cambridge, MA. 1975.

8. R. Nevatia, and K. R. Babu, "Linear Feature Extraction and Description," submitted for publication to IJCAI-79.

9. R. Nevatia, and K. Price, "Locating Structures in Aerial Images," Proc. of 4th Intl. Joint Conf. on Pattern Recognition, Kyoto, Japan, Nov. 1978, pp. 686-90.

10. P. Brodatz, Textures, New York, Dover, 1966.

2.4 Supervised Classification with Singular Value Decomposition Texture Measurement

Behnam Ashjari and William K. Pratt

In a previous report [1] on this subject, four facts about singular values were established:

i) The singular values of a matrix are measures or descriptors of inter-relationships of the matrix elements.

ii) The singular values can be considered measures of the pattern variation of image texture.

iii) The SVD is not useful as a measure or feature of structure.

shown in Fig. 16a. The mountains do not have a dominant size for the textural elements, except for some of the narrow dark (shadow) areas (Fig. 16b). The bottom row has so few edges that it can be simply described as a uniform region.

This is not a complete description of the textures, but serves as a good initial description of the patterns. There are still other important features of the textures which are not derived by this method, but could be computed by other techniques.

Conclusions

General texture analysis is a very difficult problem, but this analysis of edge images appears to be an effective method to extract many important structural features from the textural patterns. One major unanswered question is whether or not all of the information derived by the human user can be reliably derived by a program. We are still working on the automatic extraction of this information from the data which is produced by this textural analysis method.

References

1. R.M. Haralick, K. Shanmugam, and I. Dinstein, "Textural Features for Image Classification," IEEE Trans SMC-3, 1973, pp. 660-621.

2. J. Weszka, A. Rosenfeld, "A Comparative Study of Texture Measures for Terrain Classification," IEEE Trans SMC-6, April, 1976.

3. A. Rosenfeld and Kak, Digital Picture Processing, Academic Press: New York, 1976.

4. R. Bajcsy, "Computer Identification of Visual Surfaces," Computer Graphics and Image Processing-2, Oct. 1973, pp. 118-130.

5. J.T. Maleson, "Understanding Texture in Natural Images,"

iv) The singular value distribution tends toward uniformity for a less correlated image and toward skewness for higher correlation among pictorial data.

The above mentioned facts were supported by theoretical and experimental results. The experiments performed were on artificially generated, two dimensional, zero mean, unit variance, separable, first order Markov, Gaussian random fields. It was also shown that similar looking textural images possess relatively identical singular value curves.

In the present study, experiments on artificial texture are continued and also new experiments on natural texture are performed in order to pave the way for the application of the SVD as a means for an efficient texture measurement.

Experiments on Artificial Texture

Artificially generated, separable random fields, F, have been used for three sets of experiments, let

$$F = U\Lambda^{\frac{1}{2}}V^T$$

For each value of the spatial correlation ρ, there is a corresponding generated matrix F, and for each F, there are corresponding U, V, and UV^T. By transforming the U's, V's, and their outer products UV^T's to pictures, it is possible to visually detect any trend among each group.

The experiments are performed with considering different values for ρ i.e.,; 0.0, 0.5, 0.55, 0.6, ... , 1.0. The results show that most of the information is concentrated in the singular values rather than in the U and V or UV^T matrices.

Experiments on Natural Texture

In these set of experiments, four types of similar appearing natural textures are chosen from the Brodatz Texture Album [2], Grass, Raffia, Sand, and Wool. Fig. 1 shows the four texture fields. The original data is stored in a 512 x 512 array of pixels. By performing a "neighborhood averaging" operation on the 512 x 512 data, it is possible to obtain 256 x 256 and 128 x 128 version of the data. At this stage, the goal is to determine the best window size versus resolution. The windows are 16 x 16 or 32 x 32 non-overlapping squares taken from the 128 x 128, 256 x 256, or 512 x 512 images. With the help of the within class and between class plots of the singular value distributions, it is determined that the best window size is 32 x 32 and for the purpose of this study, the best resolution is in the 512 x 512 array. There are 256 non-overlapping blocks of 32 x 32 windows in a 512 x 512 picture. Through a random integer generating mechanism, 64 of the 256 possible windows are randomly chosen. Figure 2 contains sixteen 32 x 32 sample windows for Raffia and Wool. Each of these 32 x 32 squares has 32 singular values which, when arranged in descending order, can be considered as elements of a 32-dimensional vector. Therefore, the prototypes consist of 64 32-dimensional vectors for each of the 4 texture images.

Histogram Modification

To avoid any sort of bias in our 4 texture images, they must have the same mean and variance. This is achieved by subtracting the mean value of each 512 x 512 picture from each of its elements and then dividing the result by the standard deviation of the picture. Through this operation, the texture images become zero-mean and unit-variance. In order to eliminate any miscalculation due to the brightness levels of the pixels or biases due to the background lighting at the time of photography, a histogram Gaussianization is performed on all 4 texture images. After performing Histogram Gaussianization on the textures, a

iv) The singular value distribution tends toward uniformity for a less correlated image and toward skewness for higher correlation among pictorial data.

The above mentioned facts were supported by theoretical and experimental results. The experiments performed were on artificially generated, two dimensional, zero mean, unit variance, separable, first order Markov, Gaussian random fields. It was also shown that similar looking textural images possess relatively identical singular value curves.

In the present study, experiments on artificial texture are continued and also new experiments on natural texture are performed in order to pave the way for the application of the SVD as a means for an efficient texture measurement.

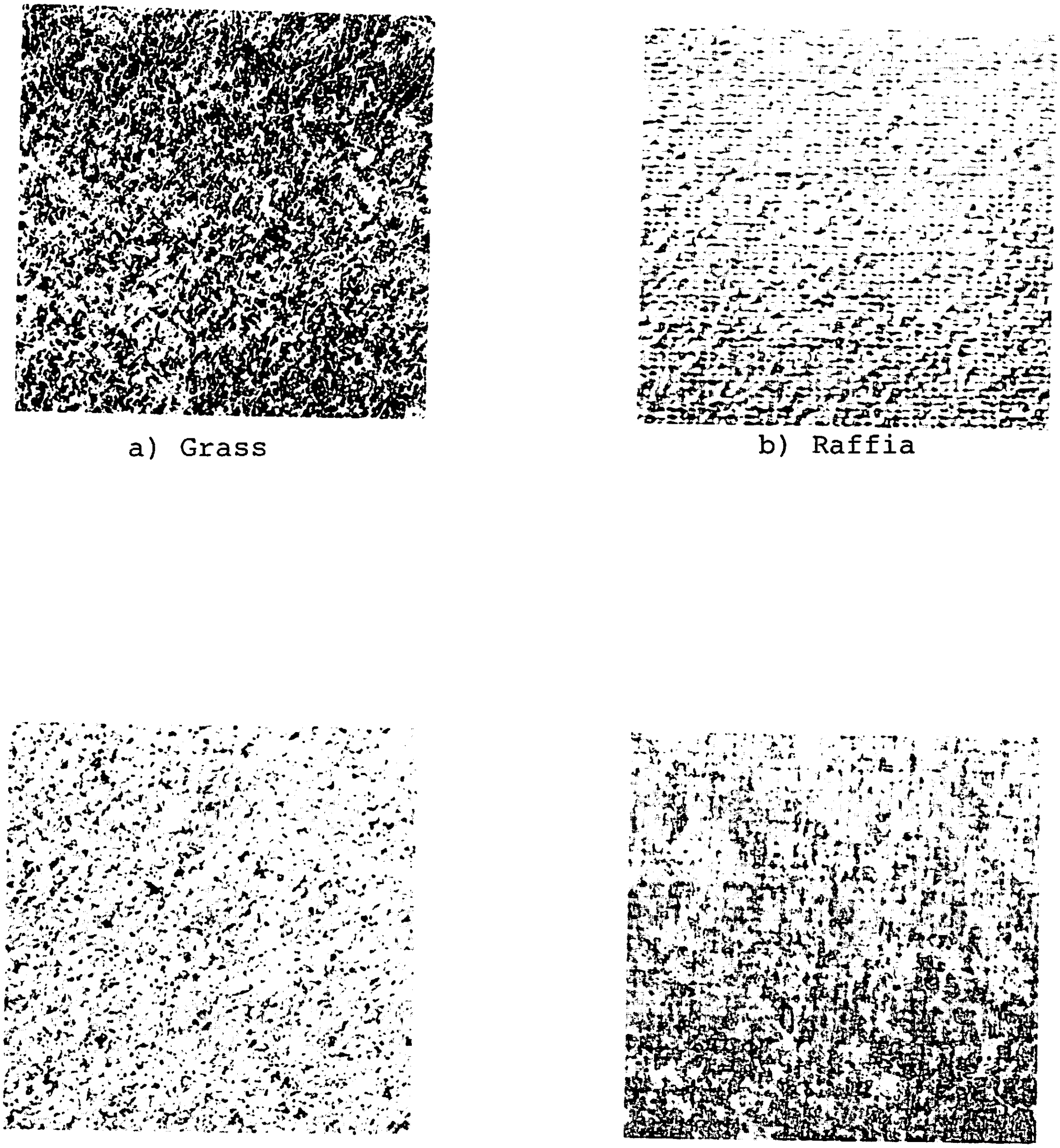
Experiments on Artificial Texture

Artificially generated, separable random fields, F, have been used for three sets of experiments, let

$$F = U\Lambda^{\frac{1}{2}}V^T$$

For each value of the spatial correlation ρ, there is a corresponding generated matrix F, and for each F, there are corresponding U, V, and UV^T. By transforming the U's, V's, and their outer products UV^T's to pictures, it is possible to visually detect any trend among each group.

The experiments are performed with considering different values for ρ i.e.,; 0.0, 0.5, 0.55, 0.6, ... , 1.0. The results show that most of the information is concentrated in the singular values rather than in the U and V or UV^T matrices.



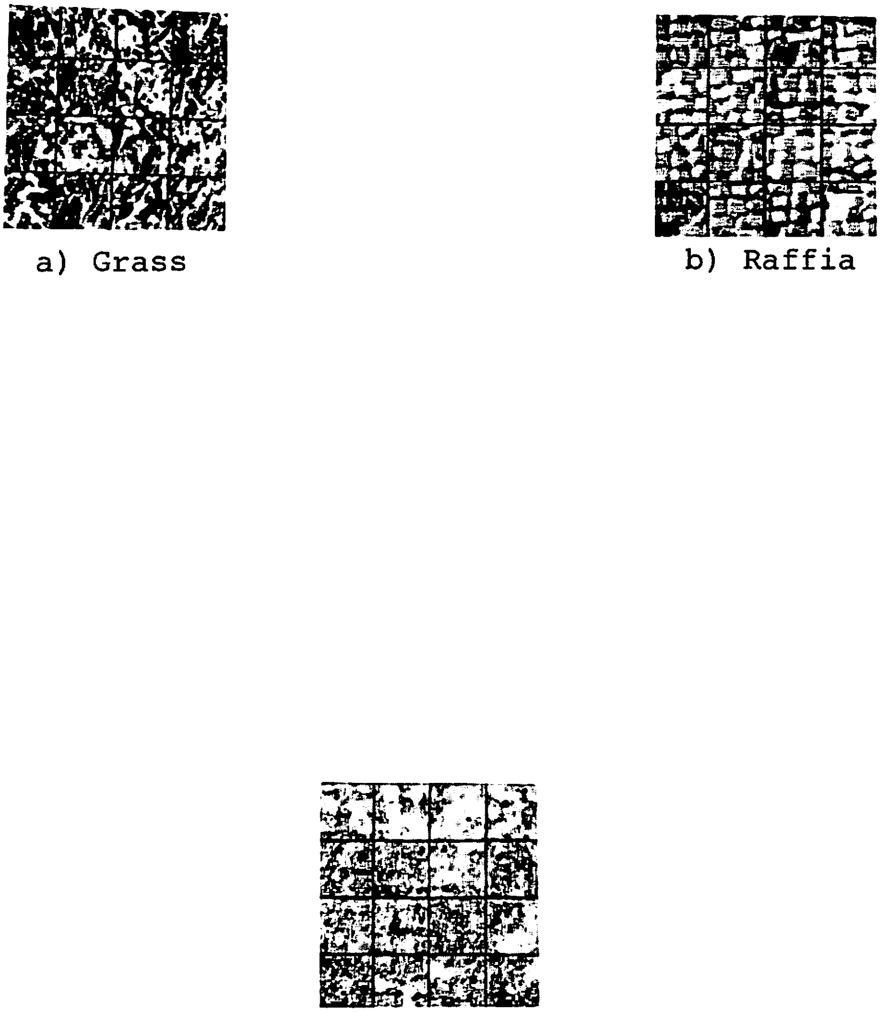
a) Grass

b) Raffia

c) Sand

d) Wool

Fig. 1. 4 original 512x512 texture fields.



a) Grass

b) Raffia

c) Wool

Fig. 2. Randomly selected 32x32 sample windows.

complete homogeneity on the data is reached and any result from the computation is solely due to the structural inter-relationship of the texture elements rather than unwanted biases.

Experimental Mean of Singular Value Vectors

As was previously mentioned, there are 64 32 x 32 sample windows from the zero-mean, unit-variance, Gaussian histogram version of each image. This, in turn, provides us with 64 32-dimensional S.V. prototype vectors. For each texture, therefore, there is a sample or experimental mean vector. Figure 3 shows the relative locations of the experimental-mean-singular value vector of the 4 texture fields.

Feature Extraction

Dividing each of the 64 32-dimensional S.V. prototype vector by the sum of its elements, will not change the relative size of each elements with the other; however, the result will be 64 32-dimensional first order S.V. histogram vector for each texture. A first order histogram vector has the property that the sum of its elements is one, and it can be concisely represented by its first 4 moments [3, P.472]. For a SV histogram vector S whose ith element is S(i), the first four moments, as explained thoroughly in [4], are as follows:

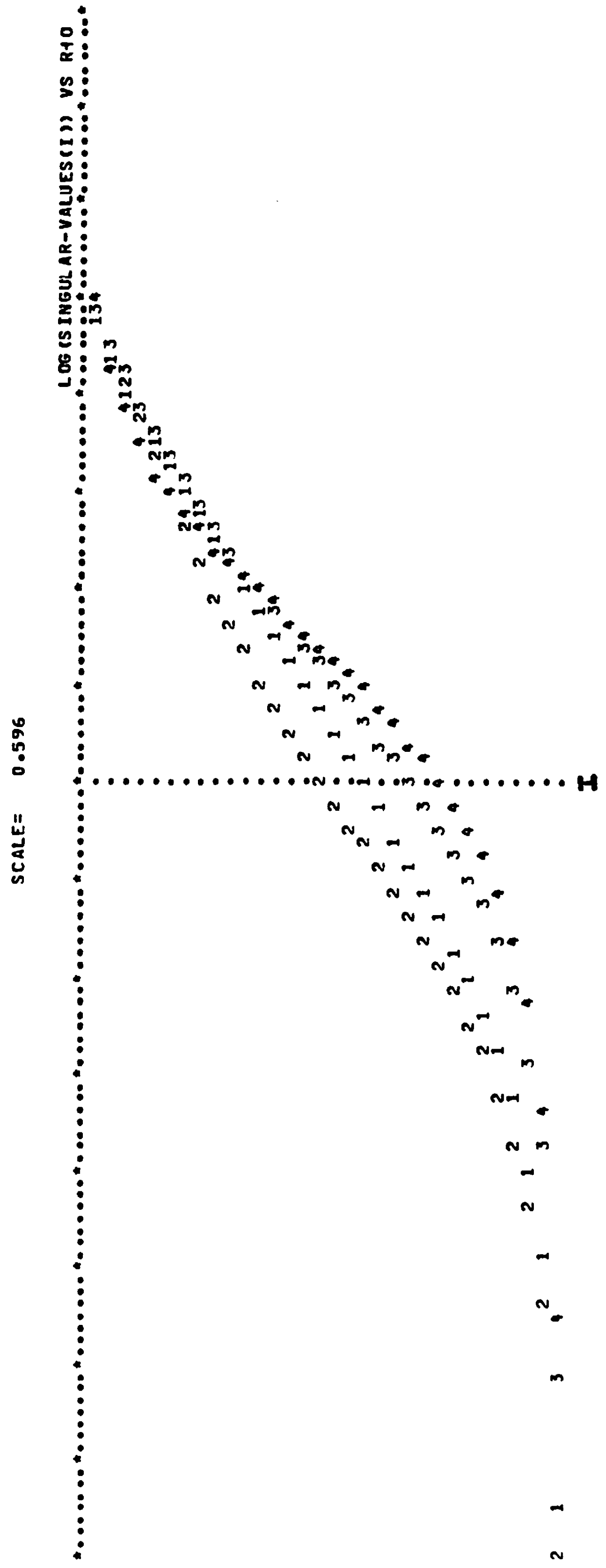
Average (First moment)

$$M_1 = \sum_{i=1}^{32} iS(i)$$

Deviation (square root of variance)

$$M_2 = \left[\sum_{i=1}^{32} (i-M_1)^2 S(i) \right]^{\frac{1}{2}}$$

Skewness (third moment)



grass ≙ 1

raffia ≙ 2

sand ≙ 3

wool ≙ 4

Fig. 3. Experimental mean of singular values of 64 sample windows for the 4 textures fields of Fig. 1.

complete homogeneity on the data is reached and any result from the computation is solely due to the structural inter-relationship of the texture elements rather than unwanted biases.

Experimental Mean of Singular Value Vectors

As was previously mentioned, there are 64 32 x 32 sample windows from the zero-mean, unit-variance, Gaussian histogram version of each image. This, in turn, provides us with 64 32-dimensional S.V. prototype vectors. For each texture, therefore, there is a sample or experimental mean vector. Figure 3 shows the relative locations of the experimental-mean-singular value vector of the 4 texture fields.

Feature Extraction

Dividing each of the 64 32-dimensional S.V. prototype vector by the sum of its elements, will not change the relative size of each elements with the other; however, the result will be 64 32-dimensional first order S.V. histogram vector for each texture. A first order histogram vector has the property that the sum of its elements is one, and it can be concisely represented by its first 4 moments [3, P.472]. For a SV histogram vector S whose ith element is S(i), the first four moments, as explained thoroughly in [4], are as follows:

Average (First moment)

$$M_1 = \sum_{i=1}^{32} iS(i)$$

Deviation (square root of variance)

$$M_2 = \left[\sum_{i=1}^{32} (i-M_1)^2 S(i) \right]^{\frac{1}{2}}$$

Skewness (third moment)

$$M_3 = \frac{1}{M_2^3} \sum_{i=1}^{32} (i-M_1)^3 S(i)$$

Kurtosis

$$M_4 = \frac{1}{M_2^4} \sum_{i=1}^{32} (i-M_1)^4 S(i)-3$$

The factor 3 in the kurtosis makes the kurtosis of a Gaussian histogram zero [4].

M_1, M_2, M_3 and M_4 features representing a SV histogrm vector can be utilized to give 64 4-dimensional feature vectors for each texture. From each set of these feature vectors, an experimental feature mean vector and an experimental feature covariance matrix can be computed for each texture field.

Evaluation of SVD Texture Measurement

In texture analysis, a set of measurements on texture is tested according to a 'goodness' criteria [5]. There are two quantitative methods for evaluation of statistical texture measures: the first is a classification method which involves measurement of classification error in classifying texture fields, and the second is the figure of merit method, which usually involves a distance function to provide a measure of separation between two texture classes [4]. The metric used for distance is usually related to classification accuracy. The larger the distance, the higher the classification accuracy. ClassifiThe cation method of evaluation will be used in future experiments to verify our figure of merit evaluation. The figure of merit technique is used in our experiments and will be described in the following section.

Bhattacharyya (B-) Distance

Using a Bayes classifier, B-distance is monotonically related to the Chernoff bound . The Chernoff bound is an upper bound on the probability of classification error [6].

For two classes with gaussian densities and equal likelihood, the B-distnace is

$$B(\text{class}_1,\text{class}_2) = \frac{1}{8}(\underline{m}_1-\underline{m}_2)^T\left[\frac{\underline{C}_1+C_2)}{2}\right](m_1-m_2)+\frac{1}{8}\ln\frac{|\frac{1}{2}(\underline{C}_1+\underline{C}_2)|}{|\underline{C}_1|^{\frac{1}{2}}|\underline{C}_2|^{\frac{1}{2}}}$$

where, m_J and C_J represent feature mean vector and feature covariance matrix of Jth class.

Table 1 contains the B-distances between 6 possible pair of the 4 texture images; Grass, Raffia, Sand and Wool. As can be seen in the table, Grass and Sand have the minimum distance. Hence, for this pair, the highest probability of classification error exists, and the Chernoff bound to this probability is about 10%.

It is possible to find a combinations out of M_1, M_2, M_3 and M_4 features which gives maximum B-distance. When such a combination is found, the features in the combination are the best ones to be utilized, in our experiments, for classification.

References

1. B. Ashjari and W.K. Pratt, "Singular Value Decomposition Image Feature Extraction," Semiannual Technical Report, USCIPI Report 800, Image Processing Institute, USC, Los Angeles, Calif. March 31, 1978, pp. 72-89.

2. P. Brodatz, Texture: A Photograph Album for Artists and Designers, New York, Dover, 1956.

$$M_3 = \frac{1}{M_2^3} \sum_{i=1}^{32} (i-M_1)^3 S(i)$$

Kurtosis

$$M_4 = \frac{1}{M_2^4} \sum_{i=1}^{32} (i-M_1)^4 S(i)-3$$

The factor 3 in the kurtosis makes the kurtosis of a Gaussian histogram zero [4].

M_1, M_2, M_3 and M_4 features representing a SV histogrm vector can be utilized to give 64 4-dimensional feature vectors for each texture. From each set of these feature vectors, an experimental feature mean vector and an experimental feature covariance matrix can be computed for each texture field.

Evaluation of SVD Texture Measurement

In texture analysis, a set of measurements on texture is tested according to a 'goodness' criteria [5]. There are two quantitative methods for evaluation of statistical texture measures: the first is a classification method which involves measurement of classification error in classifying texture fields, and the second is the figure of merit method, which usually involves a distance function to provide a measure of separation between two texture classes [4]. The metric used for distance is usually related to classification accuracy. The larger the distance, the higher the classification accuracy. ClassifiThe cation method of evaluation will be used in future experiments to verify our figure of merit evaluation. The figure of merit technique is used in our experiments and will be described in the following section.

Bhattacharyya (B-) Distance

Table 1

Field Pairs		B-Distance
Grass	Raffia	2.4790
Grass	Sand	1.4358
Grass	Wool	3.6898
Raffia	Sand	7.3492
Raffia	Wool	11.2446
Sand	Wool	3.8368

Bhattacharyya Distance of Texture
Feature Sets for Average, Deviation, Skewness, and Kurtosis.

3. W.K. Pratt, Digital Image Processing, Wiley-Interscience, New York, 1978.

4. O.D. Faugeras, W.K. Pratt, "Stochastic Based Feature Extraction," unpublished, January 1979.

5. S.K. Abdali, "Feature Extraction Algorithms," Pattern Recognition, Rergamon Press, Vol. 3, 1971, pp. 3-21.

6. K. Fukunaga, Introduction to Statistical Pattern Recognition, Wiley-Interscience, New York, 1972.

2.5 Use of Linear Features for Road Detection

K. Ramesh Babu and Ramakant Nevatia

Previously, we have described an edge detection and line finding technique that gives piecewise linear boundary segments [1]. Use of these linear features for extraction of roads and similar structures (e.g. airport runways and rivers) is described here. The described techniques are intended to be general and may be viewed as describing 2-D generalized cones [2]. This is in contrast to special techniques for road detection, such as in [3,4]. No attempt to compare the two approaches has been made here.

Basically, a 2-D generalized cone may be viewed as being bounded by locally linear and locally parallel boundaries of opposite orientations. We call such pairs of line segments as "anti-parallels" or "apars". The first step in our road detection techniques is to compute apars of a given maximum width (known from approximate scale of the image). The apar detection technique has been described

Table 1

Field Pairs		B-Distance
Grass	Raffia	2.4790
Grass	Sand	1.4358
Grass	Wool	3.6898
Raffia	Sand	7.3492
Raffia	Wool	11.2446
Sand	Wool	3.8368

Bhattacharyya Distance of Texture
Feature Sets for Average, Deviation, Skewness, and Kurtosis.

previously [1]. A line segment may form more than one apar with different pairing segments.

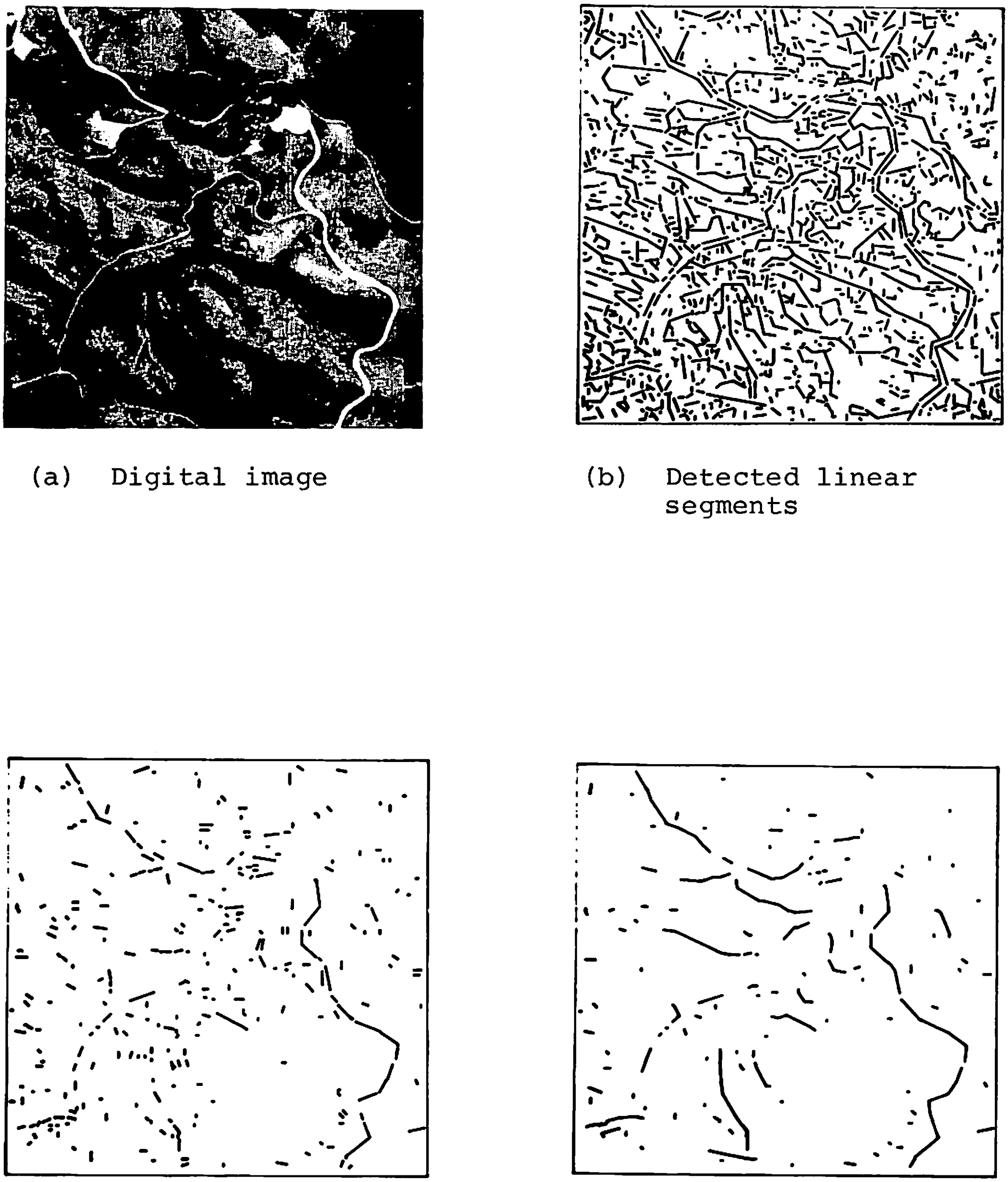
The above process generally leads to broken fragments of a road (and other structures), as many boundary segments are absent due to inadequacies of edge detection, sharp bends, road intersections and other causes. An example of detected segments and apars from fig. 1(a) is shown in fig. 1(b) and 1(c) respectively. Fortunately, some of the apars can be connected to form larger fragments, utilizing the connectivity apparent in the detected segments.

The connectivity criterion for connecting apars is that one of the segments of the apars be part of the same chain of connected segments, called the supersegment. In Fig. 3(a), the "curve" ABCDE is a supersegment while AB, BC, CD and DE are segments. The apars formed by these segments are connected and the collection is called "sap"-short for super antiparallel. The component apars occur in order, i.e., <1,2,3> or <3,2,1>. Note that, in Fig. 2(b), all 5 apars would be connected to form a single sap. The "color" (bright or dark) of the apars in a sap is also recorded. Fig. 1(d) shows the resulting saps from fig. 1(a).

The implementation of connecting apars for display purposes is complicated due to the presence of overlapping apars. The details are not described here.

Results

Results of processing another image, a 2048x2048 image of the Stockton, California area, are shown in Figs. 3(a)-(d). Note that fig. 3(d) shows saps, not all of which correspond to roads. The processing time to generate ordered saps is about 10% of the time required for previous processing to compute linear segments.



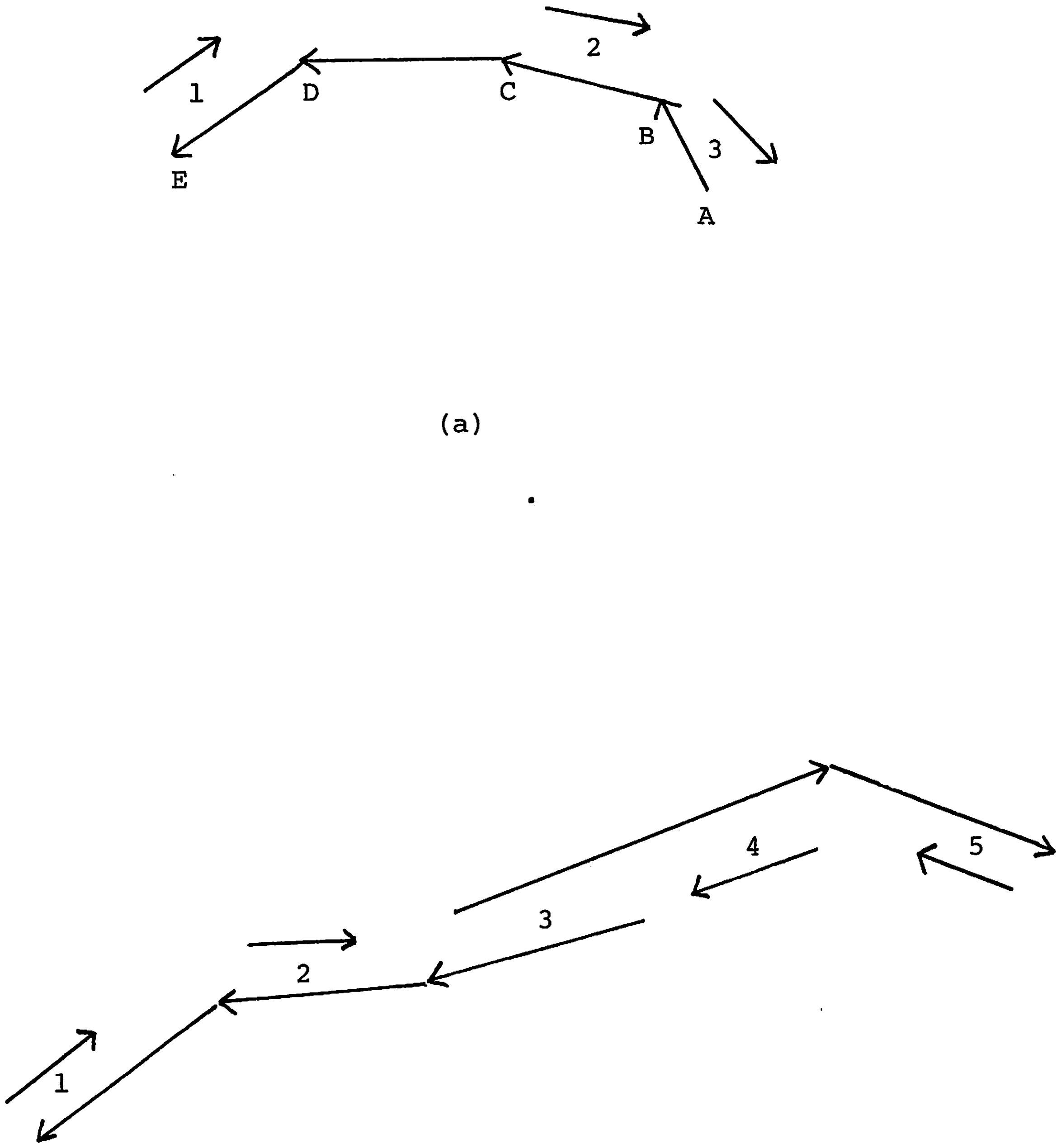
(a) Digital image

(b) Detected linear segments

(c) Detected apars

(d) Detected saps

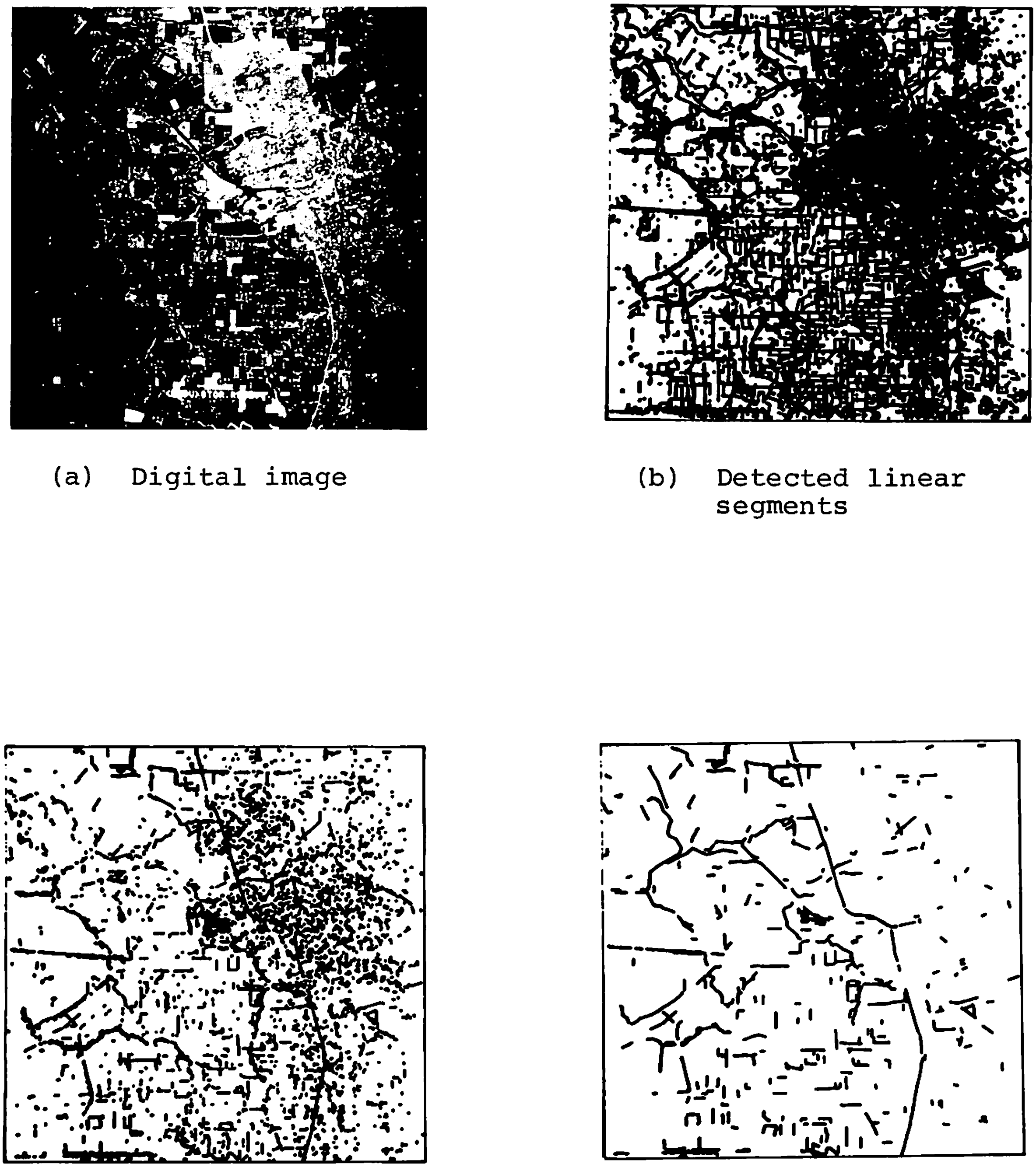
Figure 1. Steps in processing of an aerial image



(a)

(b)

Figure 2. Examples of "sap"



(a) Digital image

(b) Detected linear segments

(c) Detected apars

(d) Detected saps

Figure 3. Steps in processing of an image of the Stockton area

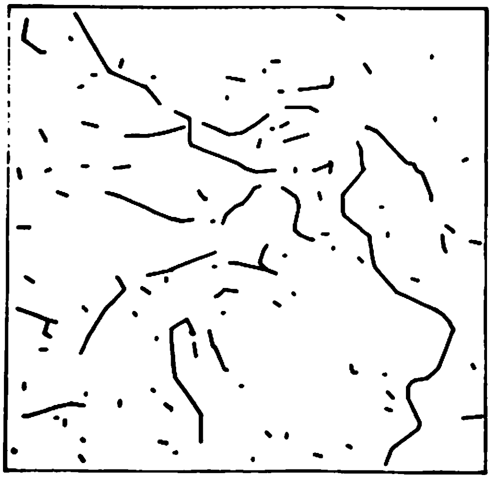


Figure 4. Detected road fragments from Fig. 1(a) after bridging

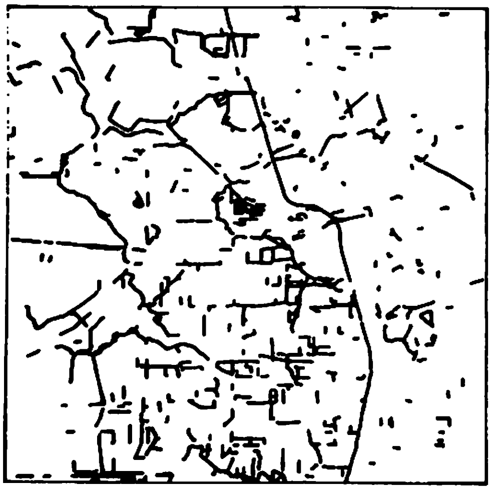


Figure 5. Detected road fragments from Fig. 3(a) after bridging

The detected roads in figs. 1 and 3 are fragmented. It was observed that a large number of gaps are due to a single missing edge. We bridge gaps between two segments if there exists a pixel between them that could be a predecessor of one segment and a successor of the other. Improvement in detected roads after bridging is easily discernible, compare figures 1(d) and 4. In the Stockton image, the results after bridging (Fig. 5) are less noticeable because of the scale of display.

References

1. R. Nevatia and K. Ramesh Babu, "Linear Feature Extraction", Proc. ARPA Image Understanding Workshop, Nov. 1978, pp. 73-78.

2. R.A. Brooks, R. Greiner, and T.O. Binford, "A Model Based Vision System", Proc. ARPA Image Understanding Workshop, May 1978, pp. 36-44.

3. R. Bajcsy and M. Tavakoli, "Computer Recognition of Roads from Satellite Pictures," IEEE Trans. Systems, Man and Cybernetics, SMC-6, 1976, pp. 623-637.

4. L.H. Quam, "Road Tracking and Anomaly Detection in Aerial Imagery", Proc. ARPA Image Understanding Workshop, May 1978, pp. 51-55.

2.6 Model Matching and Acquisition of Images

Keith Price

This note gives the current status of a system to locate complex structures in an image. Portions of the earlier wok have appeared in previous semi-annual reports, but a complete description will appear in a forth coming, separate USC IPI report.

This system uses both region based and edge based segmentation techniques. The edge based method is applied to extract the prominent linear objects. In the case of aerial images these would include roads, narrow rivers, etc. The region based method is used to segment the other objects in the image, those which should be easily used to segment the other objects represented as connected regions. Previously, regions and line based segmentations had not been effectively combined. These segments, lines and regions, are used as the basic elements in the symbolic description of the image. The symbolic description is completed by extracting features of the segments such as size, orientation, neighbors, etc.

The user describes the scene model through a dialog with the computer system. The various objects in the model are described in the same terms as the segments extracted from the image. This model is a general segment based description rather than a detailed pixel specification such as used in [1]. Additional information may be included in the model descriptions to account for the variability of the image data. For example, the size of objects may be reduced by segmentation errors, occlusions, and the location of the objects near the edge of the image. To handle this, and other similar problems, most features can be given as greater than (or less than) some value. The hand to locate (or hand to describe) objects which this system is intended to aid in finding are specified by the size of the object (either absolute size, in meters, or a percentage of another known object), and the location relative to known objects.

The automatically generated image description and the user derived scene model are used by the matching and location program to identify segments in the image as objects given in the model and to determine where in the image the hand to locate objects may be found. This model-image matching program is based on a general image-image previous semi-annual reports, but a complete description will appear in a forth coming, separate USC IPI report.

This system uses both region based and edge based segmentation techniques. The edge based method is applied to extract the prominent linear objects. In the case of aerial images these would include roads, narrow rivers, etc. The region based method is used to segment the other objects in the image, those which should be easily used to segment the other objects represented as connected regions. Previously, regions and line based segmentations had not been effectively combined. These segments, lines and regions, are used as the basic elements in the symbolic description of the image. The symbolic description is completed by extracting features of the segments such as size, orientation, neighbors, etc.

The user describes the scene model through a dialog with the computer system. The various objects in the model are described in the same terms as the segments extracted from the image. This model is a general segment based description rather than a detailed pixel specification such as used in [1]. Additional information may be included in the model descriptions to account for the variability of the image data. For example, the size of objects may be reduced by segmentation errors, occlusions, and the location of the objects near the edge of the image. To handle this, and other similar problems, most features can be given as greater than (or less than) some value. The hand to locate (or hand to describe) objects which this system is intended to aid in finding are specified by the size of the object (either absolute size, in meters, or a percentage of another known object), and the location relative to known objects.

The automatically generated image description and the user derived scene model are used by the matching and location program to identify segments in the image as objects given in the model and to determine where in the image the hand to locate objects may be found. This model-image matching program is based on a general image-image matching system [2], but has several extensions to handle the special properties of model descriptions (e.g. approximate feature values and the fact that some features are available only for a few objects). The matching procedure locates the segment in the image which most closely matches the description given in the model. These correspondences are then used by the location specifier for the structures whose locations are given as areas of the image. When these locations are known, the smaller area in the image can be further analyzed to extract detailed information.

Figure 1 is an image to illustrate how this system performs on a medium scale aerial image. The image is of San Francisco and covers about 15 by 15 miles, with 3 airports which we wish to locate. Figure 3 is a sketch of the scene from which much of the information required for the scene model can be extracted. Fig. 3 is a graph representation of the scene which illustrates the internal description of the model. Figure 4 shows the segmentation of the image-with regions outlined in white, and long lines features also marked in white. Figure 5 shows the results of applying the matching procedure, with the matched regions and road labeled. Figure 6 has the located areas (the airports) marked.

References

1. J.M Tenenbaum, "Locating Objects by Their Distinguishing Features in Multi-Sensory Images", Computer Graphics and Image Processing, Vol. 2, No. 3, December 1973.

2. K. Price, and R. Reddy, "Matching Segments of Images", IEEE Trans. PAMI, Vol. 1, No. 1, Jan. 1979.

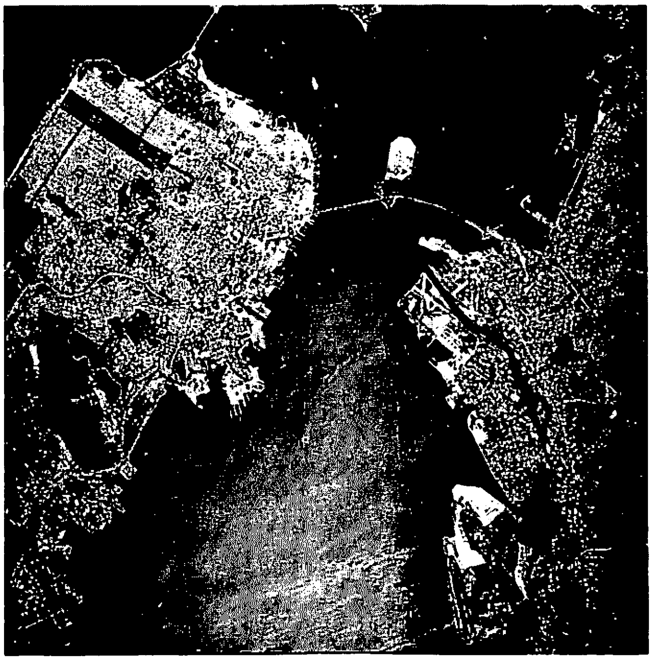


Figure 1. San Francisco Image

SAN FRANCISCO SCENE

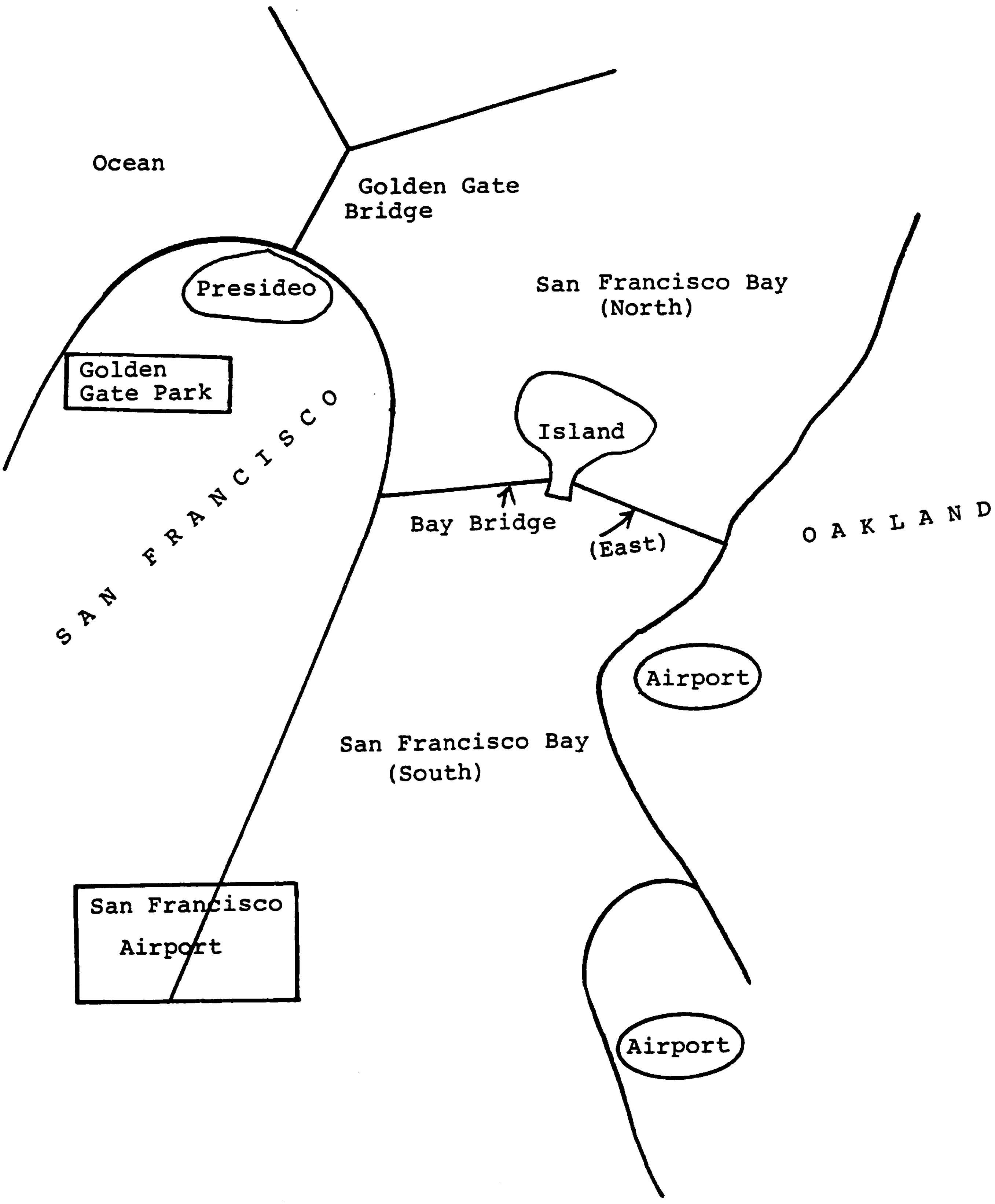


Figure 2. Sketch of San Francisco Area

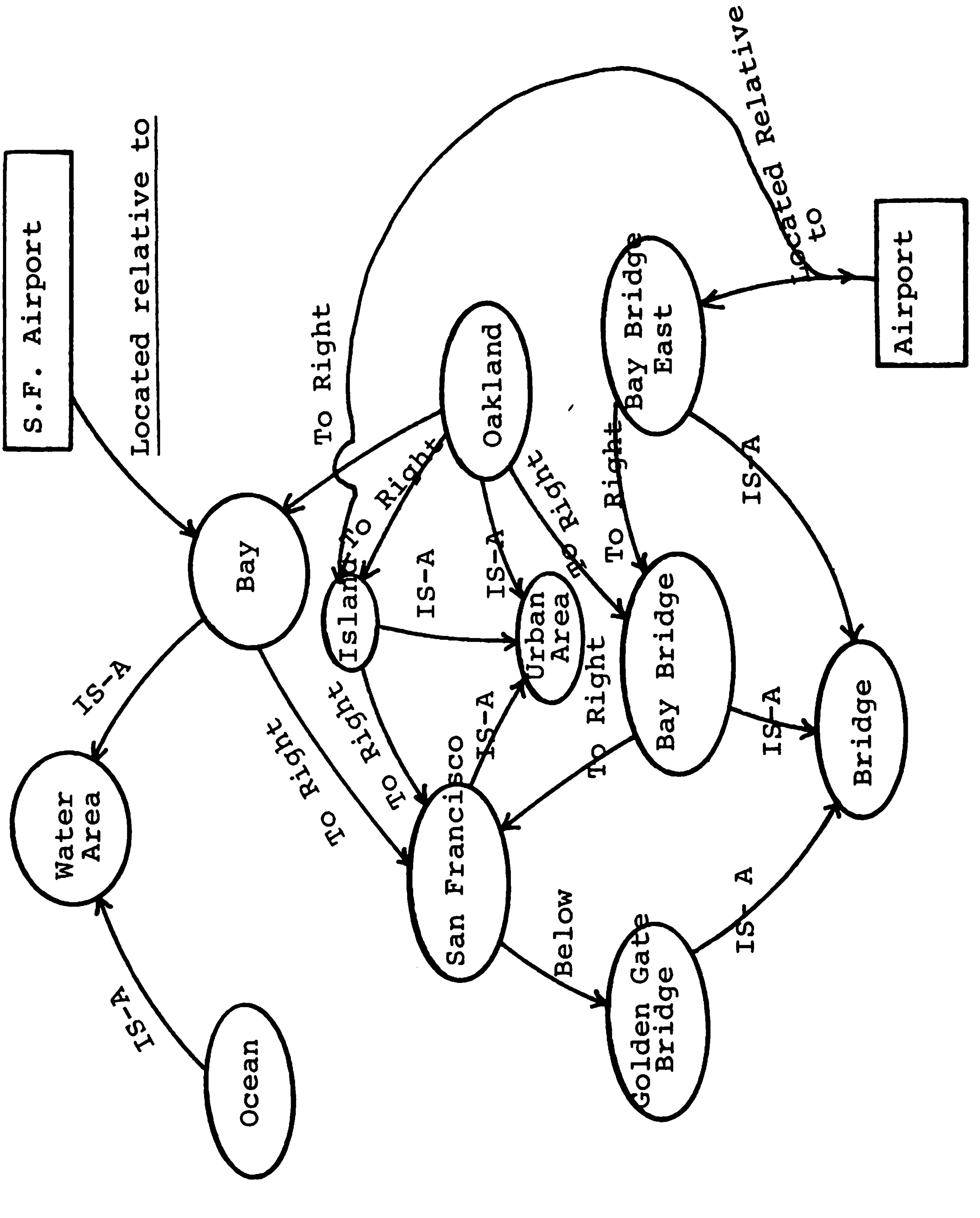


Figure 3. Graph Representation of San Francisco Area

SAN FRANCISCO SCENE

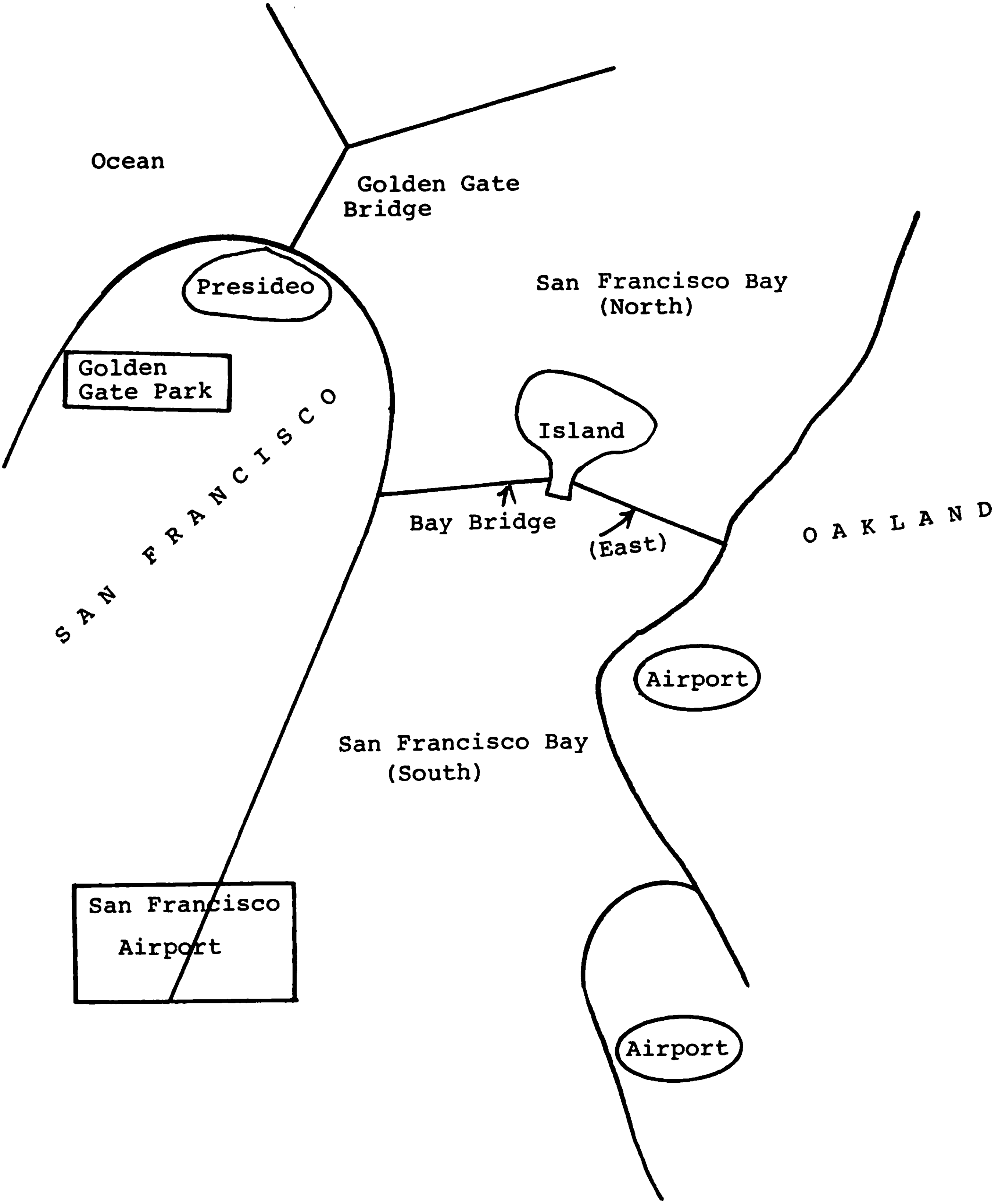


Figure 2. Sketch of San Francisco Area

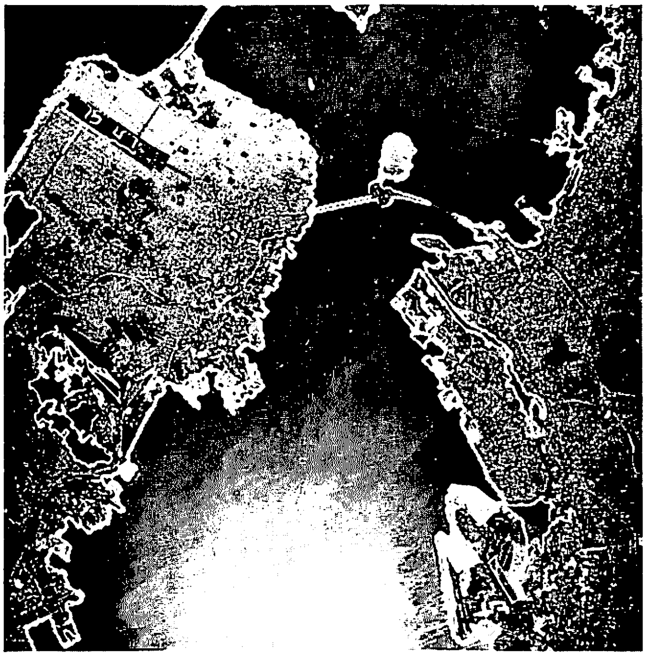


Figure 4. Segmented Regions and Linear Features of San Francisco

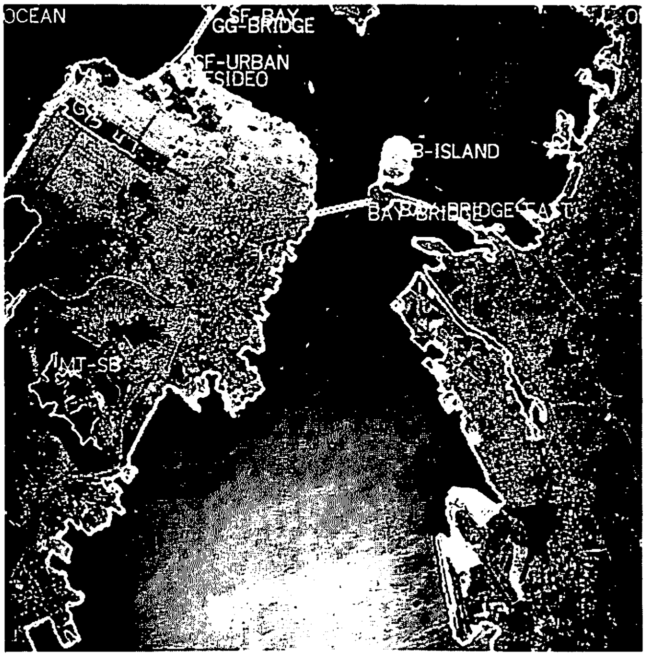


Figure 5. Recognized Regions

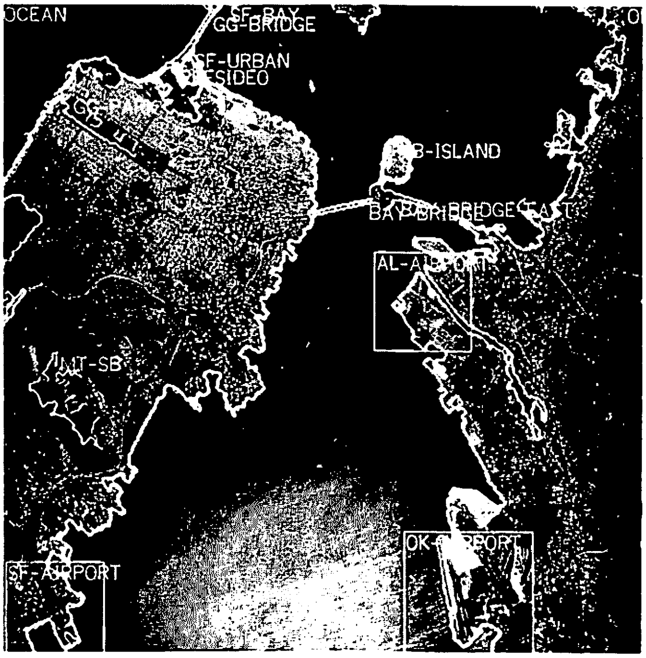


Figure 6. Airport Regions Located

3. Image Processing Projects

3.1 Two-Dimensional Small Generating Kernel Convolution

William K. Pratt, Jean F. Abramatic and Sang Uk Lee

Introduction

Small Generating Kernel (SGK) convolution is a processing technique for performing convolution on two-dimensional data arrays by sequentially convolving the array with small size convolution kernels. The processed output of the SGK system closely approximates the output obtained by convolution with a large kernel.

SGK Processing

Conventional discrete two-dimensional convolution is a linear computation defined by

$$G(j,k) \equiv F(j,k) \circledast H(j,k) = \sum_{m} \sum_{n} F(m,n) H(j-m+1, k-n+1) \tag{1}$$

where G(j,k) is an MxM output array, F(j,k) is an NxN input array, and H(j,k) is an LxL convolution kernel array called an impulse response function. The array dimensions are related by M=N+L-1. The number of multiplications required for conventional computation, in general, is N^2L^2.

In the SGK concept, the computation of eq.(1) is replaced by a sequential convolution operation. Referring to Figure 1, let

$$A_q(j,k) = A_{q-1}(j,k) \circledast K_q(j,k) \tag{2}$$

represent the convolution output of the q-th stage of the process where $K_q(j,k)$ is the q-th KxK small generating kernel, (typically 3x3). At the zeroth stage, $A_0(j,k)=F(j,k)$. In the basic SGK

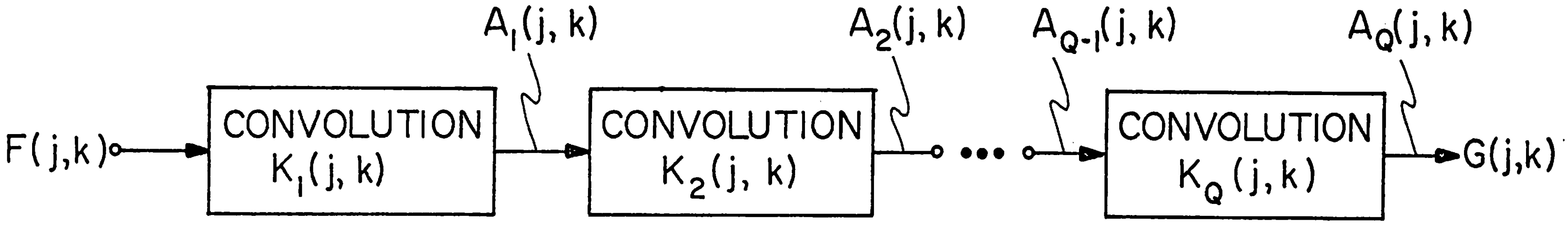


FIGURE 1. BASIC SMALL GENERATING KERNEL CONVOLUTION SYSTEM

3. Image Processing Projects

3.1 Two-Dimensional Small Generating Kernel Convolution

William K. Pratt, Jean F. Abramatic and Sang Uk Lee

Introduction

Small Generating Kernel (SGK) convolution is a processing technique for performing convolution on two-dimensional data arrays by sequentially convolving the array with small size convolution kernels. The processed output of the SGK system closely approximates the output obtained by convolution with a large kernel.

SGK Processing

Conventional discrete two-dimensional convolution is a linear computation defined by

$$G(j,k) \equiv F(j,k) \circledast H(j,k) = \sum_{m} \sum_{n} F(m,n)H(j-m+1,k-n+1) \tag{1}$$

where G(j,k) is an MxM output array, F(j,k) is an NxN input array, and H(j,k) is an LxL convolution kernel array called an impulse response function. The array dimensions are related by M=N+L-1. The number of multiplications required for conventional computation, in general, is N^2L^2.

In the SGK concept, the computation of eq.(1) is replaced by a sequential convolution operation. Referring to Figure 1, let

$$A_q(j,k) = A_{q-1}(j,k) \circledast K_q(j,k) \tag{2}$$

represent the convolution output of the q-th stage of the process where $K_q(j,k)$ is the q-th KxK small generating kernel, (typically 3x3). At the zeroth stage, $A_0(j,k)=F(j,k)$. In the basic SGK convolution system, the convolution output is

$$\hat{G}(j,k) = A_Q(j,k) = [K_1(j,k) \circledast K_2(j,k) \circledast \ldots \circledast K_Q(j,k)] \circledast F(j,k) \quad (3)$$

An alternate system, shown in figure 2, produces a convolution output from the cumulative sum of the SGK stages. In this top ladder configuration

$$\hat{G}(j,k) = \sum_{q=0}^{Q} \lambda_q A_q(j,k) \quad (4a)$$

or equivalently

$$\hat{G}(j,k) = \sum_{q=0}^{Q} \lambda_q [K_1(j,k) \circledast K_2(j,k) \circledast \ldots \circledast K_q(j,k)] \circledast F(j,k) \quad (4b)$$

Figure 3 contains a dual formulation, called the bottom ladder configuration in which the convolution and weighting stages are reversed in order. The cumulative formulations offer more convolution design freedom through independent specification of the weighting constants λ_q.

The values of the Q small generating kernels $K_q(j,k)$, and in the case of the cumulative SGK system, the Q weighting constants λ_q, are found by an optimization procedure that seeks to minimize the error between the conventional output G(j,k) of eq.(1) and the SGK output given by eq.(3) or (4). Techniques have been developed for mean square error and absolute error (Chebyshev error) minimization.

With SGK convolution, a total of

$$Q = \frac{L-1}{K-1} \quad (5)$$

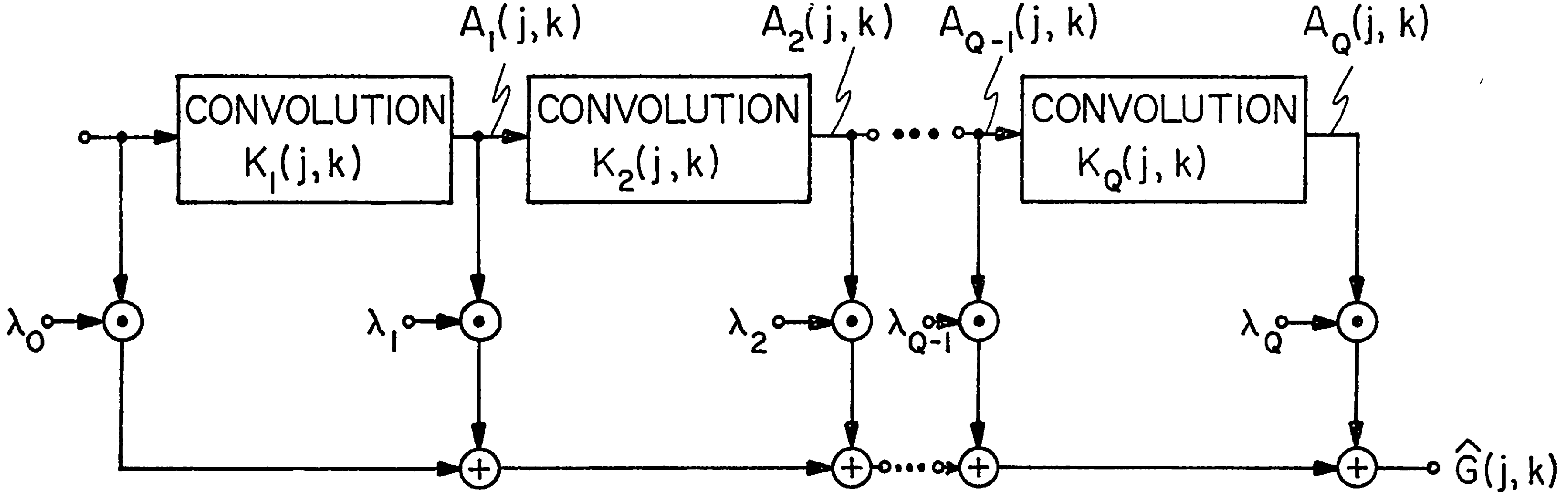


FIGURE 2. CUMULATIVE SMALL GENERATING KERNEL CONVOLUTION SYSTEM-TOP LADDER CONFIGURATION.

convolution system, the convolution output is

$$\hat{G}(j,k) = A_Q(j,k) = [K_1(j,k) \circledast K_2(j,k) \circledast \ldots \circledast K_Q(j,k)] \circledast F(j,k) \tag{3}$$

An alternate system, shown in figure 2, produces a convolution output from the cumulative sum of the SGK stages. In this top ladder configuration

$$\hat{G}(j,k) = \sum_{q=0}^{Q} \lambda_q A_q(j,k) \tag{4a}$$

or equivalently

$$\hat{G}(j,k) = \sum_{q=0}^{Q} \lambda_q [K_1(j,k) \circledast K_2(j,k) \circledast \ldots \circledast K_q(j,k)] \circledast F(j,k) \tag{4b}$$

Figure 3 contains a dual formulation, called the bottom ladder configuration in which the convolution and weighting stages are reversed in order. The cumulative formulations offer more convolution design freedom through independent specification of the weighting constants λ_q.

The values of the Q small generating kernels $K_q(j,k)$, and in the case of the cumulative SGK system, the Q weighting constants λ_q, are found by an optimization procedure that seeks to minimize the error between the conventional output G(j,k) of eq.(1) and the SGK output given by eq.(3) or (4). Techniques have been developed for mean square error and absolute error (Chebyshev error) minimization.

With SGK convolution, a total of

$$Q = \frac{L-1}{K-1} \tag{5}$$

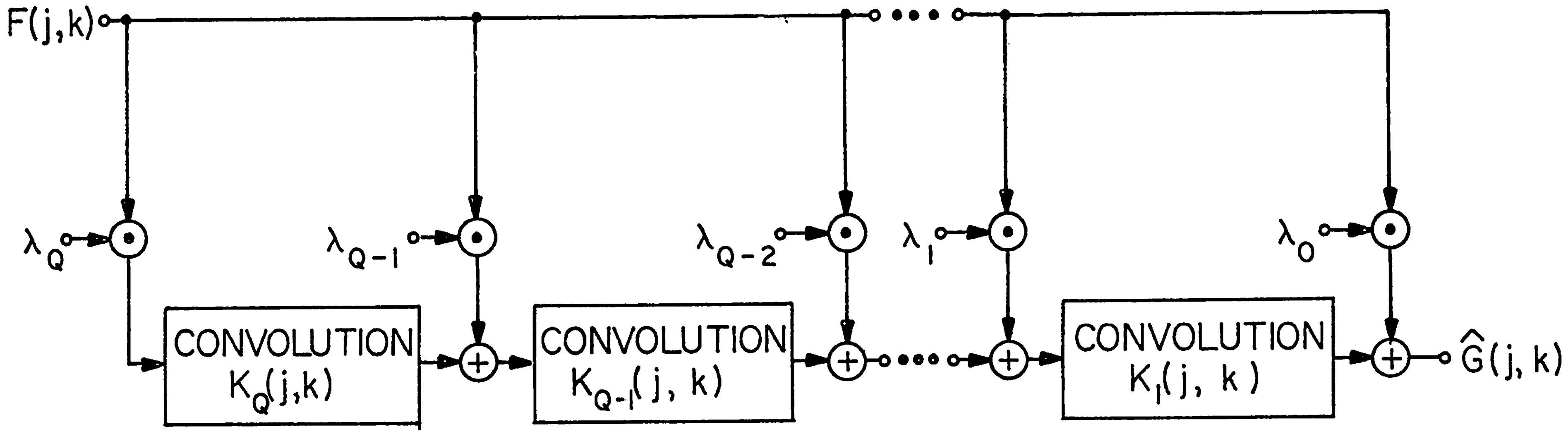


FIGURE 3. CUMULATIVE SMALL GENERATING KERNEL CONVOLUTION SYSTEM — BOTTOM LADDER CONFIGURATION.

stages of SGK convolution with an KxK kernel producing the equivalent of an LxL kernel conventional convolution. Thus, 15x15 convolution requires 7 stages of convolution with a 3x3 SGK. The number of SGK operations required is QK^2N^2. The saving in computation over conventional convolution is by the ratio

$$S = \frac{L^2N^2}{QK^2N^2} = \frac{K-1}{K^2} \frac{L^2}{L-1} \tag{6}$$

For the 15x15 convolution example using a 3x3 SGK, the computational saving is by a factor of about 3.6:1.

SVD/SGK Processing

If an impulse response operation matrix $\underline{H}$ is orthogonally separable such that is can be expressed in the outer product form

$$\underline{H} = \underline{a}\ \underline{b}^T \tag{7}$$

where $\underline{a}$ and $\underline{b}$ are column and row operator vectors, respectively, then the convolution operation of eq.(1) can be performed by sequential convolution on the rows and columns of F(j,k). This reduces the number of computations to 2NL instead of N^2L^2 in the two-dimensional case.

Any impulse response operator can be expressed as a sum of sepa:able operators by a singular value decomposition (SVD) by which

$$\underline{H} = \sum_{i=1}^{R} \psi_i \underline{a}_i \underline{b}_i^T \tag{8}$$

stages of SGK convolution with an KxK kernel producing the equivalent of an LxL kernel conventional convolution. Thus, 15x15 convolution requires 7 stages of convolution with a 3x3 SGK. The number of SGK operations required is QK^2N^2. The saving in computation over conventional convolution is by the ratio

$$S = \frac{L^2N^2}{QK^2N^2} = \frac{K-1}{K^2} \quad \frac{L^2}{L-1} \tag{6}$$

For the 15x15 convolution example using a 3x3 SGK, the computational saving is by a factor of about 3.6:1.

SVD/SGK Processing

If an impulse response operation matrix $\underline{H}$ is orthogonally separable such that is can be expressed in the outer product form

$$\underline{H} = \underline{a}\ \underline{b}^T \tag{7}$$

where $\underline{a}$ and $\underline{b}$ are column and row operator vectors, respectively, then the convolution operation of eq.(1) can be performed by sequential convolution on the rows and columns of F(j,k). This reduces the number of computations to 2NL instead of N^2L^2 in the two-dimensional case.

Any impulse response operator can be expressed as a sum of sepa: able operators by a singular value decomposition (SVD) by which

$$\underline{H} = \sum_{i=1}^{R} \psi_i \underline{a}_i \underline{b}_i^T \tag{8}$$

where $R/(R \leq L)$ is the rank of $\underline{H}$, ψ_i is the i-th eigenvalue of $\underline{HH}^T(\underline{H}^T\underline{H})$, and $\underline{a}_i$ and $\underline{b}_i$ are the i-th eigenvectors of $\underline{HH}^T$ and $\underline{H}^T\underline{H}$ respectively. If $\underline{H}$ is of full rank, the number of operations required to perform the convolution is of the order of $2NL^2$. In many practical cases, the rank R of $\underline{H}$ is much less than L, and the number of operations can be reduced accordingly. Also, it is possible to truncate the expansion of eq.(8) if the ψ_i are small.

It is possible to perform a two-dimensional SGK expansion on each submatrix $\underline{H}_i=\underline{a}_i\underline{b}_i^T$ of eq.(8). Another approach, shown in figure 4, is to sequentially expand each one-dimensional operator $\underline{a}_i$ and $\underline{b}_i$ by the SGK method into a sequence of Kx1 kernels. The latter approach is particularly attractive for two reasons. First, large kernel size two-dimensional convolution can be performed by sequential one-dimensional convolution with small operators, say 3x1, resulting in considerable savings in processing complexity. Second, there is no approximation error associated with the one-dimensional SGK expansion, and therefore, the convolution operation is theoretically perfect.

Experimental Results

Several experiments have been performed to evaluate the SGK design procedure. Figure 5 contains an example of image deblurring by conventional and SGK processing. The original image before blurring is shown in figure 5a and after blurring in figure 5b. Figure 5c shows the result of convolution with a 15x15 prototype deblurring operation. The result of deblurring with seven stage 3x3 SGK processing is presented in figure 5d. There are no observable difference between the conventional and SGK processing methods.

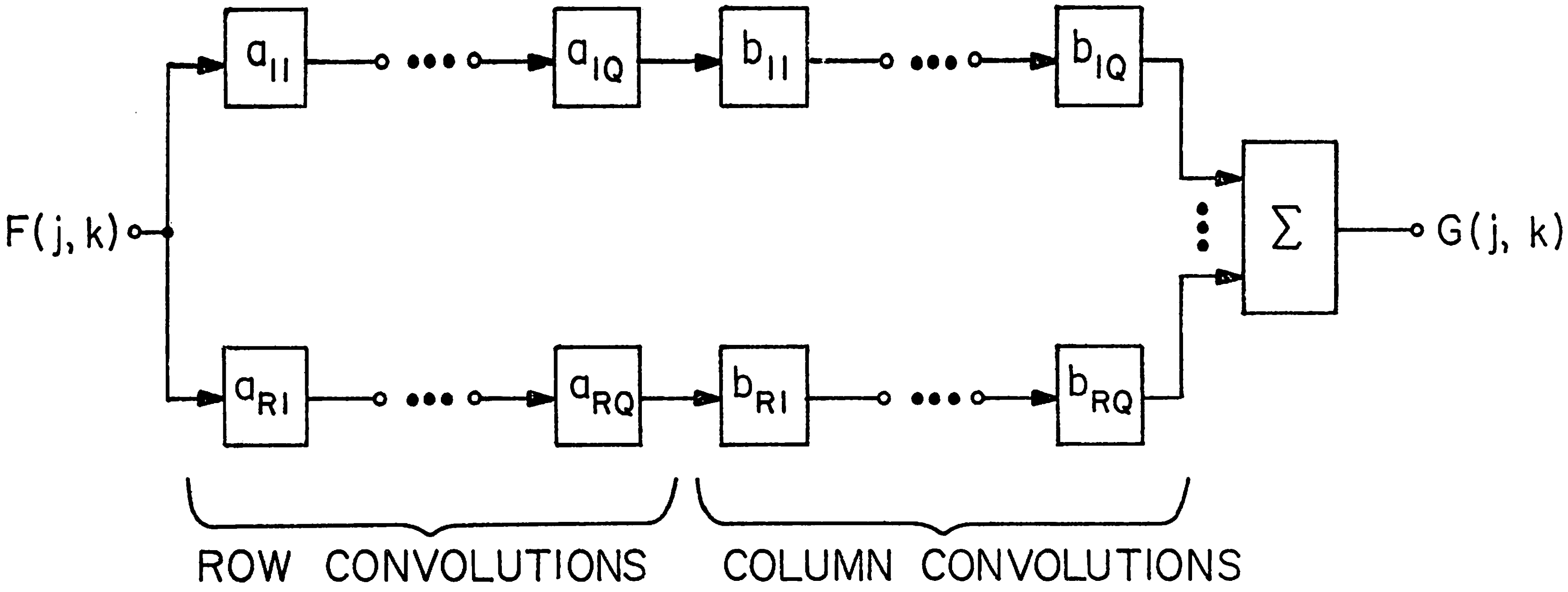


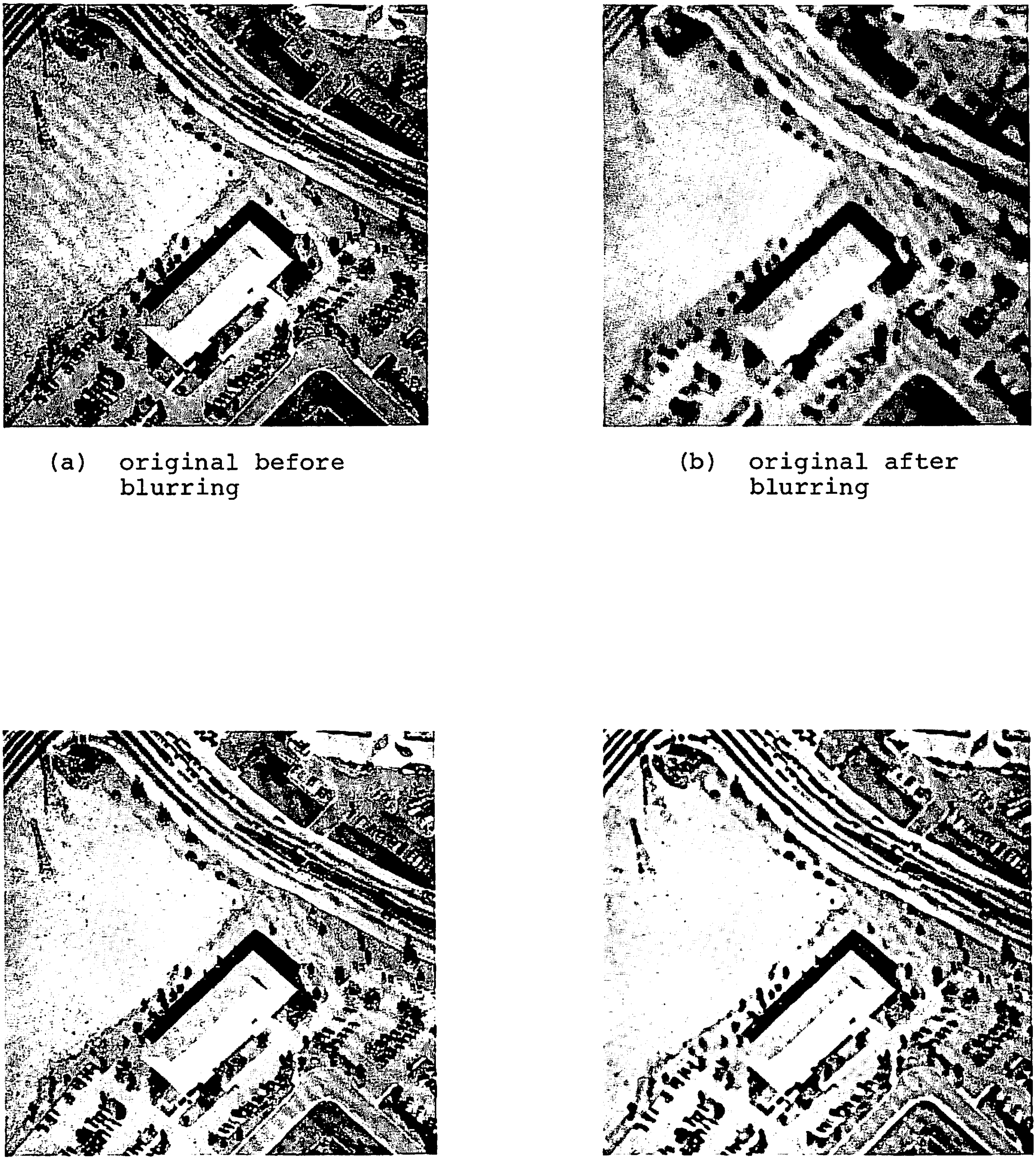
FIGURE 4. SVD/SGK CONVOLUTION

where R/(R<L) is the rank of $\underline{H}$, ψ_i is the i-th eigenvalue of $\underline{HH}^T(\underline{H}^T\underline{H})$, and $\underline{a}_i$ and $\underline{b}_i$ are the i-th eigenvectors of $\underline{HH}^T$ and $\underline{H}^T\underline{H}$ respectively. If $\underline{H}$ is of full rank, the number of operations required to perform the convolution is of the order of $2NL^2$. In many practical cases, the rank R of $\underline{H}$ is much less than L, and the number of operations can be reduced accordingly. Also, it is possible to truncate the expansion of eq.(8) if the ψ_i are small.

It is possible to perform a two-dimensional SGK expansion on each submatrix $\underline{H}_i=\underline{a}_i\underline{b}_i^T$ of eq.(8). Another approach, shown in figure 4, is to sequentially expand each one-dimensional operator $\underline{a}_i$ and $\underline{b}_i$ by the SGK method into a sequence of Kx1 kernels. The latter approach is particularly attractive for two reasons. First, large kernel size two-dimensional convolution can be performed by sequential one-dimensional convolution with small operators, say 3x1, resulting in considerable savings in processing complexity. Second, there is no approximation error associated with the one-dimensional SGK expansion, and therefore, the convolution operation is theoretically perfect.

Experimental Results

Several experiments have been performed to evaluate the SGK design procedure. Figure 5 contains an example of image deblurring by conventional and SGK processing. The original image before blurring is shown in figure 5a and after blurring in figure 5b. Figure 5c shows the result of convolution with a 15x15 prototype deblurring operation. The result of deblurring with seven stage 3x3 SGK processing is presented in figure 5d. There are no observable difference between the conventional and SGK processing methods.



(a) original before blurring

(b) original after blurring

(c) deblurring with 15x15 prototype operator

(d) deblurring with seven stage 3x3 SGK

Figure 5. Examples of image deblurrinq with conventional and SGK convolution

Conclusions

The SGK and SDV/SGK convolution methods are attractive techniques for simplifying the computational requirements of two-dimensional convolution. Studies are now underway to assess the effects of computational errors of the processing procedures.

3.2 Wiener Image Restoration Condition Number

Sang Uk Lee and William K. Pratt

In a previous paper [1], Pratt has introduced an easily computed formulation of the condition number of a finite length convolution operator. operator. This formulation is extended here to the Wiener filter operator.

Problem Statement

Assume a linear, signal-independent noise model of image formation and recording. Also for simplicity, initial consideration will be given to the one-dimensional overdetermined model

$$\underline{g} = \underline{H}\,\underline{f} + \underline{n} \qquad (1)$$

where $\underline{g}$ and $\underline{n}$ are mx1 vectors of the observed image with M=N+L-1, L is the length of the impulse response length, $\underline{f}$ is a Nx1 vector of the ideal image and $\underline{H}$ is an MxN blur matrix associated with the impulse response h(n), for n=0,1,...,L-1.

The Wiener filter deconvolution operator for this model is [2]

$$\underline{W} = (\underline{H}^T\underline{K}_n^{-1}\underline{H} + \underline{K}_f^{-1})^{-1}\underline{H}^T\underline{K}_n^{-1} \qquad (2)$$

where $\underline{K}_n$ and $\underline{K}_f$ are covariance matrices of the noise and ideal image, respectively. In numerical analysis, the accuracy of the deconvolution estimate can be bounded in terms of the noise pertubation $\underline{n}$. It has been shown [3] that

$$\frac{\|\Delta\underline{f}\|}{\|\underline{f}\|} \leq \|\underline{H}\| \; \|\underline{W}\| \; \frac{\|\underline{n}\|}{\|\underline{g}\|} \tag{3}$$

and the condition number of Wiener filter operator is

$$C(\underline{W}) = \|H\| \|\underline{W}\| \tag{4}$$

where $\|\cdot\|$ denotes matrix Euclidean norm defined as

$$\|\underline{H}\| = \sum_{i=1}^{M} \sum_{j=1}^{N} \left[|h(i,j)|^2 \right]^{\frac{1}{2}} \tag{5}$$

Computation of $\underline{W}$ is not simple because of the matrix inversion operation in eq. (2).

Pratt [1] shown that a close approximation can be obtained by replacing the generalied inverse norm by the less restrictive conditional inverse norm. So our problem leads to the computation of $\|\underline{W}^{\#}\|$.

Formulation

A finite-length operator $\underline{D}$ can be extracted from the circular superposition operator $\underline{C}$ by use of a selection matrix according to the relation

$$\underline{D} = [\underline{S1}_J^N] \; \underline{C} \; [\underline{S1}_J^M]^T \tag{6}$$

where the selection matrix $[\underline{S1}_J^K]$ is defined as

Conclusions

The SGK and SDV/SGK convolution methods are attractive techniques for simplifying the computational requirements of two-dimensional convolution. Studies are now underway to assess the effects of computational errors of the processing procedures.

3.2 Wiener Image Restoration Condition Number

Sang Uk Lee and William K. Pratt

In a previous paper [1], Pratt has introduced an easily computed formulation of the condition number of a finite length convolution operator. operator. This formulation is extended here to the Wiener filter operator.

Problem Statement

Assume a linear, signal-independent noise model of image formation and recording. Also for simplicity, initial consideration will be given to the one-dimensional overdetermined model

$$\underline{g} = \underline{H}\ \underline{f} + \underline{n} \qquad (1)$$

where $\underline{g}$ and $\underline{n}$ are mx1 vectors of the observed image with M=N+L-1, L is the length of the impulse response length, $\underline{f}$ is a Nx1 vector of the ideal image and $\underline{H}$ is an MxN blur matrix associated with the impulse response h(n), for n=0,1,...,L-1.

The Wiener filter deconvolution operator for this model is [2]

$$\underline{W} = (\underline{H}^T\underline{K}_n^{-1}\underline{H} + \underline{K}_f^{-1})^{-1}\underline{H}^T\underline{K}_n^{-1} \qquad (2)$$

where $\underline{K}_n$ and $\underline{K}_f$ are covariance matrices of the noise and ideal image,

$$[\underline{S}|_J^K] = [\underbrace{I_N}_{K} \vdots \underbrace{\underline{0}}_{J-K}]\}K \tag{7}$$

It is easily verified that the conditional inverse of matrix $\underline{D}$ is

$$\underline{D}^\# = [\underline{S1}_J^N]\ \underline{C}^{-1}\ [\underline{S1}_J^M]^T \tag{8}$$

but the generalized inverse $\underline{D}$ does not satisfy such a relation. That is,

$$\underline{D}^- \neq [\underline{S1}_J^N]\ \underline{C}^{-1}\ [\underline{S1}_J^M]^T \tag{9}$$

The conditional inverse norm $\underline{D}$ can be computed in terms of $\underline{C}$ which can be computed easily by Fourier methods since $\underline{C}$ is also a circulant matrix. Pratt shown that the conditional inverse norm $\|\underline{D}^\#\|$ is

$$\|\underline{D}^\#\|^2 = \sum_{\mu=0}^{M-1} \frac{N}{M^2}\ [|\tilde{h}(u)|^2]^- \tag{10}$$

where $\tilde{h}(u)$ is the Fourier transformation of $h(n)$, $n=0,1,\ldots,L-1$.

The next step is to use an adjoint Wiener model, which was proposed in the development of a fast Wiener restoration algorithm [4]. The resulting adjoint Wiener model estimate is

$$\hat{\underline{f}} = [S|_J^N]\ \underline{V}_x\ [\underline{S}|_J^M]^T\ \underline{g} \tag{11}$$

and

$$\underline{V}_x = [\underline{C}^T\underline{K}_{nx}^{-1}\underline{C} + \underline{K}_{fx}^{-1}]^{-1}\ \underline{C}^T\ \underline{K}_{fx}^{-1} \tag{12}$$

where $\underline{K}_{nx}$ and $\underline{K}_{fx}$ are extended covariance matrix of noise and ideal image, respectively. The adjoint Wiener model operator is

$$\underline{W}_x = [\underline{S1}^N_J][\underline{C}^T K^{-1}_{nx}\underline{C} + \underline{K}^{-1}_{fx}]^{-1}\underline{C}^T \underline{K}^{-1}_{fx}[\underline{S1}^M_J]^T \tag{13}$$

Assuming a white noise process with noise energy σ_n^2, a Markov process with image energy σ_f^2, and also using a fact that matrix $\underline{V}_x$ is a circulant matrix, leads immediately to

$$\|\underline{W}_x\|^2 = \sum_{\mu=0}^{M-1} \frac{N}{M^2} [|\nu_x(\mu,\mu)|^2] = \sum_{\mu=0}^{M-1} \frac{N}{M^2} \left[\left| \frac{\tilde{h}(u)}{|\tilde{h}(u)|^2 + \frac{1}{\lambda R \lambda(u)}} \right|^2 \right] \tag{14}$$

where $\tilde{h}(u)$ is a Fourier transform of impulse response, R is the signal to noise ratio σ_f^2/σ_n^2, and

$$\lambda(u) = \frac{(1-\rho^2)[1-(-1)^{\mu-1}\rho^{N-1}]}{1-2\rho\cos\theta + \rho^2} \tag{15}$$

with the approximate condition number for the adjoint Wiener model operator then becomes

$$C(\underline{W}) = \|H\| \ \|\underline{W}_x\| \tag{16}$$

where the norm of the condition operator is given by eq. (5) and the norm of the condition inverse of the operator by eq. (14).

$$[\underline{S}|_J^K] = [\underbrace{I_N}_{K} \vdots \underbrace{\underline{0}}_{J-K}]\}K \tag{7}$$

It is easily verified that the conditional inverse of matrix $\underline{D}$ is

$$\underline{D}^{\#} = [\underline{S1}_J^N]\ \underline{C}^{-1}\ [\underline{S1}_J^M]^T \tag{8}$$

but the generalized inverse $\underline{D}$ does not satisfy such a relation. That is,

$$\underline{D}^{-} \neq [\underline{S1}_J^N]\ \underline{C}^{-1}\ [\underline{S1}_J^M]^T \tag{9}$$

The conditional inverse norm $\underline{D}$ can be computed in terms of $\underline{C}$ which can be computed easily by Fourier methods since $\underline{C}$ is also a circulant matrix. Pratt shown that the conditional inverse norm $\|\underline{D}^{\#}\|$ is

$$\|\underline{D}^{\#}\|^2 = \sum_{\mu=0}^{M-1} \frac{N}{M^2}\ [|\tilde{h}(u)|^2]^- \tag{10}$$

where $\tilde{h}(u)$ is the Fourier transformation of $h(n)$, $n=0,1,\ldots,L-1$.

The next step is to use an adjoint Wiener model, which was proposed in the development of a fast Wiener restoration algorithm [4]. The resulting adjoint Wiener model estimate is

$$\hat{\underline{f}} = [S|_J^N]\ \underline{V}_x\ [\underline{S}|_J^M]^T\ \underline{g} \tag{11}$$

and

$$\underline{V}_x = [\underline{C}^T \underline{K}_{nx}^{-1} \underline{C} + \underline{K}_{fx}^{-1}]^{-1}\ \underline{C}^T\ \underline{K}_{fx}^{-1} \tag{12}$$

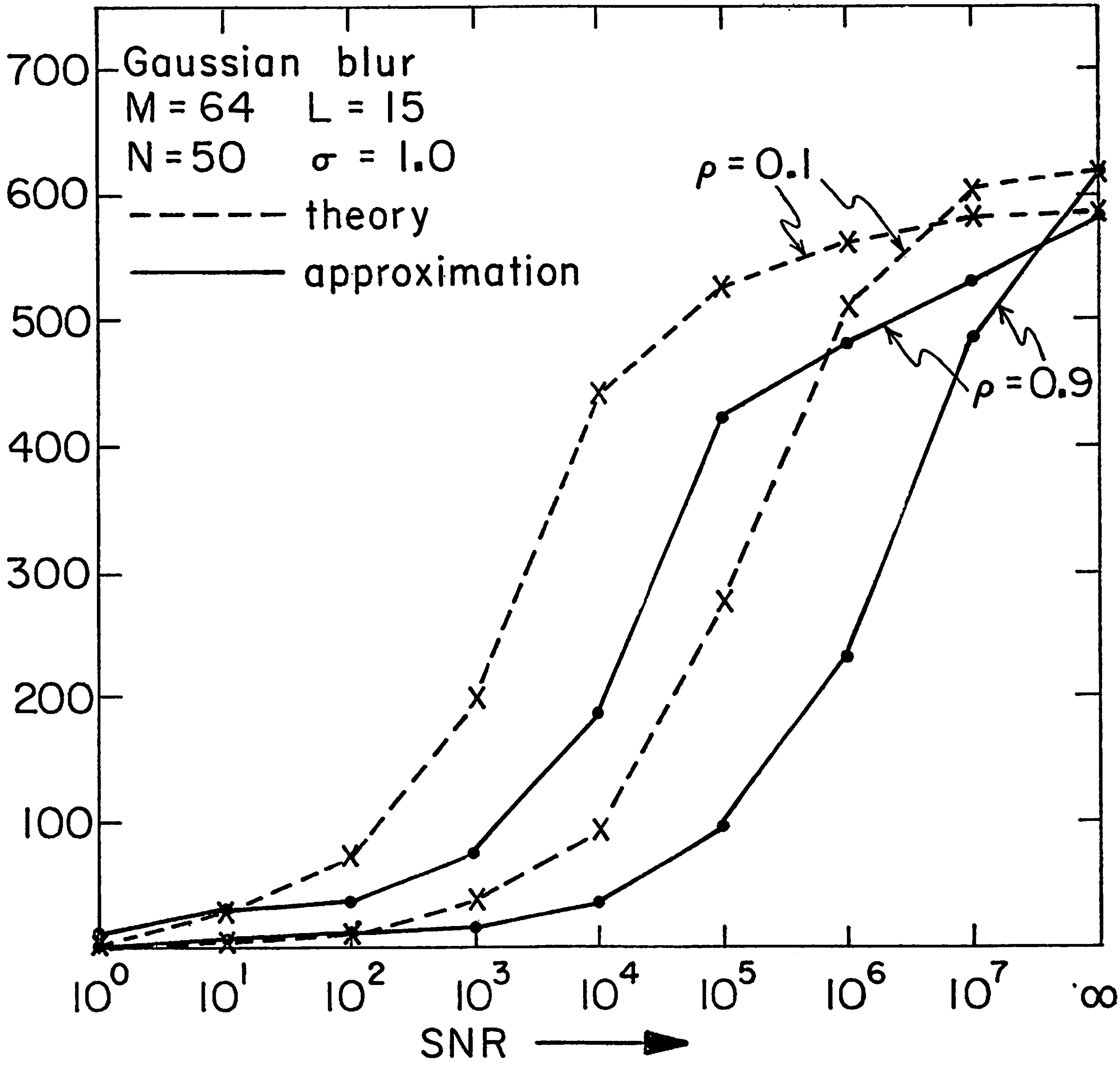


Figure 1. Comparison of approximated and exact condition number for Wiener filter.

Evaluation and Summary

Equation (16) has been evaluated and compared with an exact condition number of the Wiener filter deconvolution operator given by eq.(2) using the singular value decomposition formula of a previous report [1]. A typical result for Gaussian blur (σ^2=1.0), M=64, N=50, and L=15 is presented in Figure 1. Two correlation coefficients, ρ=0.9, 0.1, are chosen in this experiment. As the ratio of image energy to noise energy (σ_f^2/σ_n^2) approaches infinity, the Wiener operator norm of eq.(14) becomes equivalent to the conditional inverse norm of eq.(10) as shown in Figure 1.

References

1. W.K. Pratt, "Condition Number Computation of A Discrete Deconvolution Operator," University of Southern California, Image Processing Institute, USCIPI Report 840, September 78, pp. 123-135.

2. W.K. Pratt, Digital Image Processing, Wiley-Interscience, New York, 1978.

3. B.W. Rust and W.R. Burrus, Mathematical Programming and the Numerical Solution of Linear Equation, American Elsevier, New York, 1972.

4. W.K. Pratt and F. Davarian, "Fast Computational Techniques for Pseudoinverse and Wiener Image Restoration," IEEE Transactions on Computers, Vol. C-26, No. 6, June 1977, pp. 571-580.

3.3 Estimation of Blurred Image Signals with Poisson Noise

Chun Moo Lo and A.A. Sawchuk

Evaluation and Summary

Equation (16) has been evaluated and compared with an exact condition number of the Wiener filter deconvolution operator given by eq.(2) using the singular value decomposition formula of a previous report [1]. A typical result for Gaussian blur ($\sigma^2=1.0$), M=64, N=50, and L=15 is presented in Figure 1. Two correlation coefficients, $\rho=0.9$, 0.1, are chosen in this experiment. As the ratio of image energy to noise energy (σ_f^2/σ_n^2) approaches infinity, the Wiener operator norm of eq.(14) becomes equivalent to the conditional inverse norm of eq.(10) as shown in Figure 1.

References

1. W.K. Pratt, "Condition Number Computation of A Discrete Deconvolution Operator," University of Southern California, Image Processing Institute, USCIPI Report 840, September 78, pp. 123-135.

2. W.K. Pratt, *Digital Image Processing*, Wiley-Interscience, New York, 1978.

3. B.W. Rust and W.R. Burrus, *Mathematical Programming and the Numerical Solution of Linear Equation*, American Elsevier, New York, 1972.

4. W.K. Pratt and F. Davarian, "Fast Computational Techniques for Pseudoinverse and Wiener Image Restoration," *IEEE Transactions on Computers*, Vol. C-26, No. 6, June 1977, pp. 571-580.

3.3 Estimation of Blurred Image Signals with Poisson Noise

Chun Moo Lo and A.A. Sawchuk

Introduction

This report discusses more general, realistic, and interesting cases of restoration which include degradations due to blurring and Poisson noise. In many practical problems of interest, the detected image data arises from a blurred image of the object. Examples include linear motion degradation in which the object suffers significant motion during the detection interval T, Gaussian blurring degradations in which the detected image is seriously degraded by the spatial and temporal fluctuations of refractive index of the atmosphere, and aberrations, which arises in focusing error or in inherent properties of spherical lenses. These blurring effects can be lumped together as a blurring matrix H. The block diagram of this system is shown in Fig. 1.

The formulation of MAP estimation equation and its solution is derived in section 2. The implementation of the MAP filter with one- and two-dimensional blurring and their experimental results will be illustrated and discussed in section 3 and section 4, respectively.

MAP Estimation Equations with Blurring Matrix H

As derived previously [17], the estimation equation for the MAP estimate is

$$\lambda H^T(\underline{q}-\underline{1})-R_f^{-1}(\underline{f}-\underline{\bar{f}}) = 0 \tag{1}$$

where

$$\underline{q} = [q_1,q_2,\ldots q_N]^T,\ q_i = \frac{d_i}{b_i} = \frac{d_i}{\sum_j H_{ij}f_j} \tag{2}$$

and

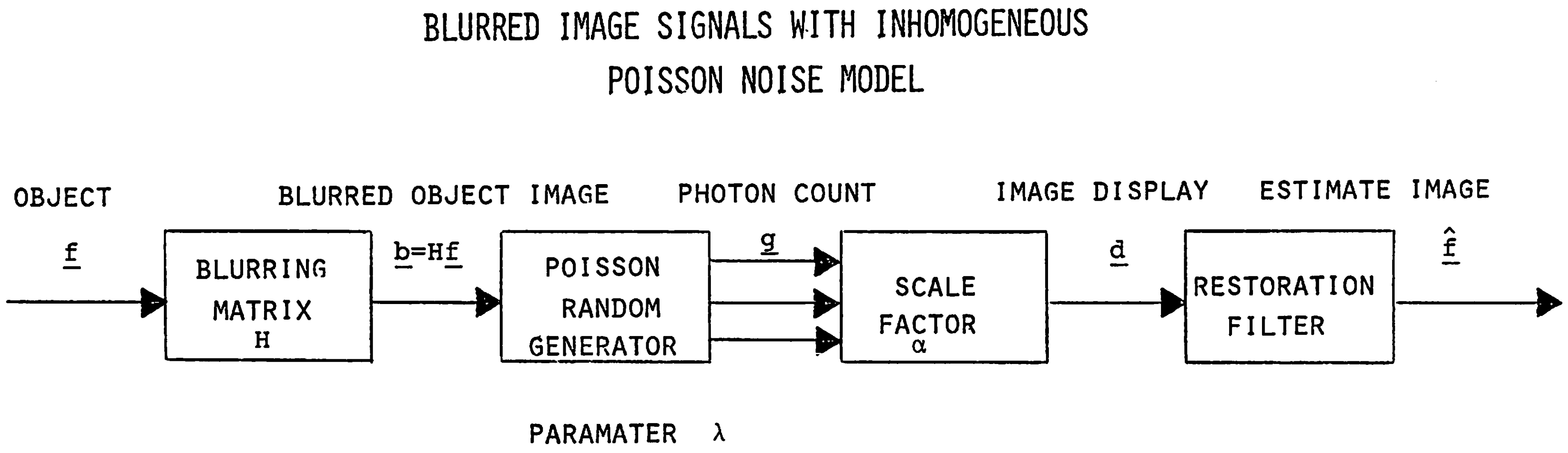


Fig. 1. Block diagram of blurred image signals with inhomogeneous Poisson noise model.

Introduction

This report discusses more general, realistic, and interesting cases of restoration which include degradations due to blurring and Poisson noise. In many practical problems of interest, the detected image data arises from a blurred image of the object. Examples include linear motion degradation in which the object suffers significant motion during the detection interval T, Gaussian blurring degradations in which the detected image is seriously degraded by the spatial and temporal fluctuations of refractive index of the atmosphere, and aberrations, which arises in focusing error or in inherent properties of spherical lenses. These blurring effects can be lumped together as a blurring matrix H. The block diagram of this system is shown in Fig. 1.

The formulation of MAP estimation equation and its solution is derived in section 2. The implementation of the MAP filter with one- and two-dimensional blurring and their experimental results will be illustrated and discussed in section 3 and section 4, respectively.

MAP Estimation Equations with Blurring Matrix H

As derived previously [17], the estimation equation for the MAP estimate is

$$\lambda H^T(\underline{q}-\underline{1})-R_f^{-1}(\underline{f}-\underline{\bar{f}}) = 0 \tag{1}$$

where

$$\underline{q} = [q_1,q_2,\ldots q_N]^T, \ q_i = \frac{d_i}{b_i} = \frac{d_i}{\sum\limits_j H_{ij}f_j} \tag{2}$$

and

$$H \triangleq \begin{bmatrix} H_{11} & H_{12} & \text{------} & H_{1N} \\ H_{21} & H_{22} & \text{------} & H_{2N} \\ \vdots & \vdots & & \vdots \\ H_{N1} & H_{N2} & & H_{NN} \end{bmatrix} \quad \text{and} \quad \underline{1} = [1,1,\ldots,1]^T \qquad (3)$$

Here the H matrix is not necessarily a square matrix, but depends on the model of blurring degradation. For simplicity, we assume here that the H matrix is square matrix.

Equation (1) is the most important key equation of MAP estimation with a blurring matrix. The first term is the ML solution and the second term is an a priori solution. Thus, the MAP filter tries to balance the inverse solution with some smoothness contraint. However, Eq. (2) is a nonlinear MAP estimate equation. The nonlinearity is buried in the $\underline{g}$ function.

Due to the larger dimensionality and nonlinearity of the MAP estimate equation it uses a sectioning method with a Newton-Raphson technique to obtain a suboptimal solution [10,12,21]. There are two sectioning methods, one is the overlap-add sectioning method, the other is the overlap-save sectioning method. Sectioning methods generally give rise to boundary edge effects, hence, it is necessary to investigate which method will be applicable to this MAP estimate equation. From equation (1), we have

$$\lambda H^{T}(\underline{q} + \underline{1}) - R_{f}^{-1}(\underline{f} - \overline{\underline{f}}) = 0 \tag{4}$$

where

$$q_i = \frac{d_i}{\sum\limits_{k} H_{ik} f_k} \tag{5}$$

In the overlap-add method of sectioning filtering the mth and (m+1)th section are added together in the region of overlap to create the final correct output. This method will be suitable only for the linear function case. However, the convolution with H is imbedded inside of the $\underline{q}$ function of equation (4). Since $\underline{q}$ is a nonlinear function of $(H\underline{f})$, then

$$\underline{q}(H\underline{f}^{(m)}+H\underline{f}^{(m+1)} \neq \underline{q}(H\underline{f}^{(m)})+\underline{q}(H\underline{f}^{(m+1)}) \tag{6}$$

If $\underline{f}^{(m)}$ belongs to the overlapped portion of a section m, and $\underline{f}^{(m+1)}$ belongs to the overlapped portion of an adjacent section m+1, then obviously overlap-add section method is not valid in the presence of the nonlinear function in the MAP estimate equation. Fortunately, the overlap-save method remains valid for the nonlinear case and can be used in our MAP estimate equation. Since incorrect points in the overlap region are discarded, rather than being corrected by addition, the overlap-save sectioning method with the Newton-Raphson technique can reduce the boundary edge effects because it discard erroneous processed data of the overlapped region.

Implementation of MAP filter with One Direction Blurring Degradation and its Experimental Results

The most interesting blurring degradations are linear motion blurring and atmospheric turbulence blurring. The linear blurring is

$$H \triangleq \begin{bmatrix} H_{11} & H_{12} & - - - - - - & H_{1N} \\ H_{21} & H_{22} & - - - - - - & H_{2N} \\ \vdots & \vdots & & \vdots \\ H_{N1} & H_{N2} & & H_{NN} \end{bmatrix} \quad \text{and} \quad \underline{1} = [1,1,\ldots,1]^T \qquad (3)$$

Here the H matrix is not necessarily a square matrix, but depends on the model of blurring degradation. For simplicity, we assume here that the H matrix is square matrix.

Equation (1) is the most important key equation of MAP estimation with a blurring matrix. The first term is the ML solution and the second term is an a priori solution. Thus, the MAP filter tries to balance the inverse solution with some smoothness contraint. However, Eq. (2) is a nonlinear MAP estimate equation. The nonlinearity is buried in the $\underline{g}$ function.

Due to the larger dimensionality and nonlinearity of the MAP estimate equation it uses a sectioning method with a Newton-Raphson technique to obtain a suboptimal solution [10,12,21]. There are two sectioning methods, one is the overlap-add sectioning method, the other is the overlap-save sectioning method. Sectioning methods generally give rise to boundary edge effects, hence, it is necessary to investigate which method will be applicable to this MAP estimate equation. From equation (1), we have models as a square blurring, while atmospheric turbulence blurring is modeled as Gaussian blurring for a long exposure time. This section will implement the MAP filter with one directional linear motion blurring case. The linear motion blurring is most general and complex case of the blurring degradation, because its amplitude response has singularities and phase reversal. We also assume that R_f has a first order Markovian covariance matrix.

Using the overlap-save section method with the iterative Newton-Raphson technique obtains the solution to the MAP estimate Eq. (4). The convergence is very fast; about 2 to 3 iterative steps. The detailed Newton-Raphson method is described in [18]. The discrete point spread function of h(x,y) is 5 pixels width of linear motion blurring degradation. The simulation is same techniques as previous report. The nonstationary mean is estimated by a 1-dimensional moving average on 11 pixels of observation data and its variance is estimated globally by an unbiased estimate of population. The linear system equations of gradient function of Eq. (4), which obtains increment value of iterative roots, is heavily dependent on structure of the blurring matrix H.

When H matrix is symmetrical matrix, the computing time of Eq. (4) with Newton-Raphson technique can be lesser since it uses the symmetrical properties of linear system equation. This simulation is done with one directional linear motion blurring (5 pixels blurring) and different $(\overline{SNR})_{rms}$. The processing of sectional MAP filter is done by a section size 36 pixels with overlapping 8 pixels. The restored images of MAP filter are shown in Fig. 2 for different $(\overline{SNR})_{rms}$. The layout of the result pictures are as follows:

The upper left picture is an original image.

The upper right picture is a Poisson noisy image.

The lower left picture is the restored image with $\rho = 0$.

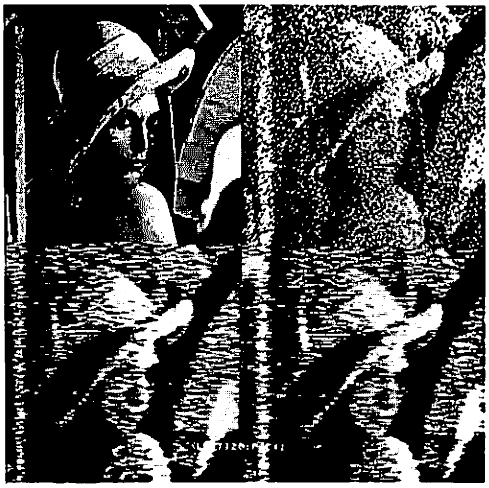


Fig. 2a. The restored image of the MAP filter with $(\overline{\text{SNR}})_{\text{rms}} = \sqrt{10}$.

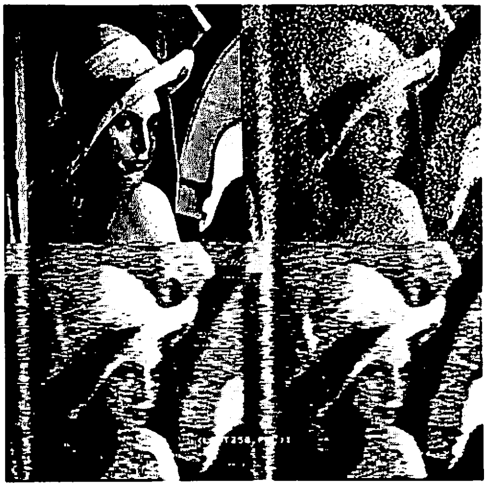


Fig. 2b. The restored image of the MAP filter with $(\overline{\text{SNR}})_{\text{rms}} = \sqrt{15}$.

The lower right picture is the restored image with $\rho = 0.9$

From Fig. 3, it is clear that the ill-condition of restored image with $\rho = 0$ more severe for the higher $(\overline{SNR})_{rms}$ image signal. The reason why is more correlation between pixels for higher $(\overline{SNR})_{rms}$ and the singularity of amplitude response of blurring matrix H which amplified the Poisson noise.

For a local adaptive MAP filter, the equation is

$$W\cdot[\lambda H^T(\underline{q}-1)]+(I-W)\cdot[-R_f^{-1}(\underline{f}-\underline{\overline{f}})] = 0 \tag{7}$$

Where $W = \{\text{diag } W_i\}$. W is called weighting matrix. W_i is the weight of ith section which can be varied on the first moment and second moment of local properties of image. The local adaptive MAP filter also can be used for the restoration of image degraded by spatially variant point spread function. This has long been a problem for real world of image processing. For instance, the point spread function of each photon detector is not identical in the whole array detectors.

Since the image can be divided into section image and treat each section with space invariant assumption. For simplicity, it will simulate the global adaptive MAP sectioning filter which set $W_i = W_h$ for any i,j.

The simulation is done with $\rho = 0.95$ for different weight of section and for different $(\overline{SNR})_{rms}$. The experimental results are shown in Fig. 4. The layout of picture is as follows:

The upper left picture is an original image.

The upper right picture is a noisy image.

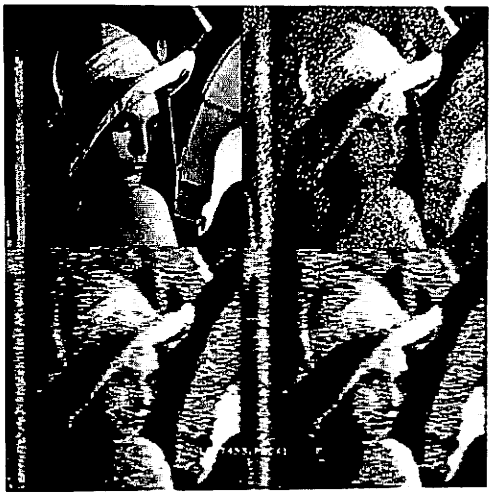


Fig. 3a. The restored image of global adaptive MAP filter with weight Wi = 0.3.

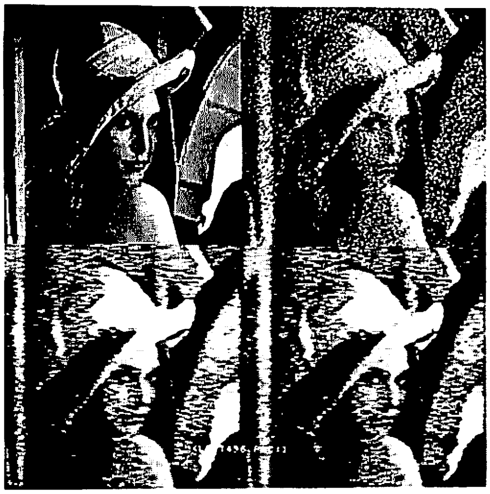


Fig. 3b. The restored image of global adaptive MAP filter with weight Wi = 0.6.

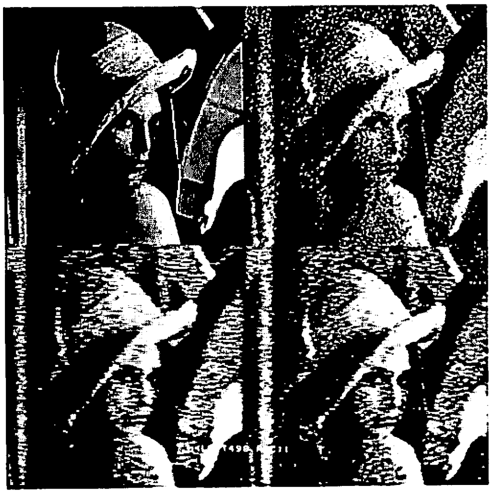


Fig. 3c. The restored image of global adaptive MAP filter with weight Wi = 0.8.

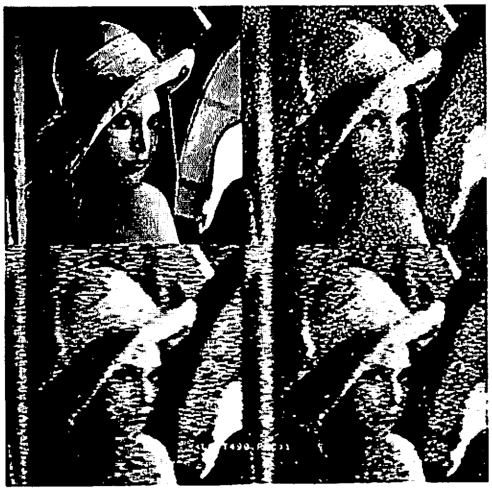


Fig. 3d. The restored image of global adaptive MAP filter with weight Wi = 0.9.

The lower left picture is the restored image of MAP Filter with $W_i = 0.5$.

The lower right picture is the restored image of MAP Filter with different W_i value.

The $W_i = 0.5$ is equal weight between maximum likelihood (ML) solution and a priori solution.

From the experimental results, it is illustrated that the more weight on the ML term, the higher frequency information can be extracted. Also, the overweight on the ML solution results in ill-conditioning of some solutions of the MAP estimate. Since the overweight on the ML solution, the MAP estimate will be asymptotically approached to ML estimate. The ML estimate indeed, is the inverse filter of the image restoration with blurring degradation case. Since the amplitude response of PSF of a linear motion blurring has the singularities and also it seriously distorted by the Poisson noisy degradation. Therefore, the global adaptive filter has an optimal weight filter. Consequently, the local adaptive filter has an optimal weight filter in the blurring degradation cases, too.

Implementation of the MAP Filter with 2-Dimensional Blurring Degradation and its Experimental Results

This section will discuss the implementation and experimental results of the MAP filter with 2-dimensional blurring degradation which often encounters in many practical interested cases. The overlap-save sectioning MAP filter will be implemented with separable assumption and nonseparable assumption, respectively. The blurring degradation is simulated by 3 x 3 pixels blurring. The non-stationary mean is estimated by 2-dimensional moving average with window size 7 x 7 pixels over the detected image intensity. The separable of H

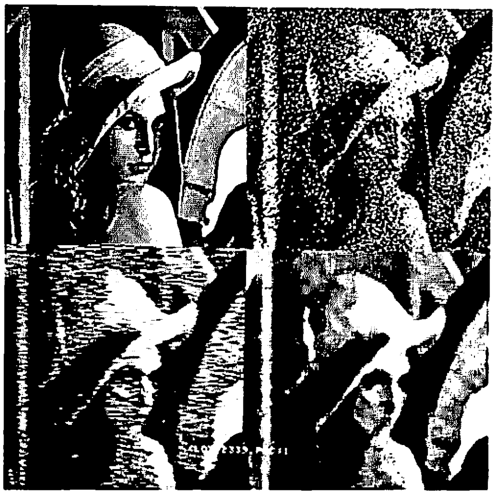


Fig. 4a. The restored image of the MAP filter with $(\overline{SNR})_{rms} = \sqrt{15}$ for separable 2-D MAP filter.

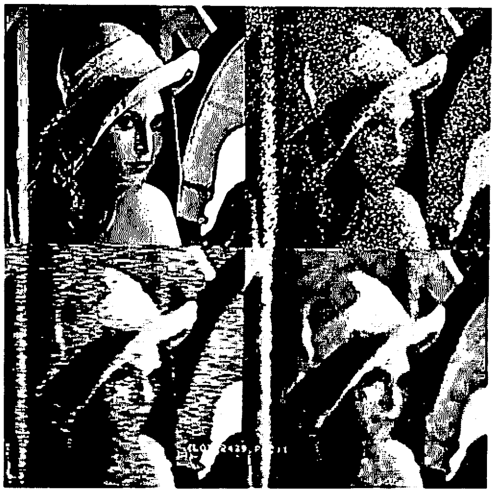


Fig. 4b. The restored image of the MAP filter with $(\overline{SNR})_{rms} = \sqrt{20}$ for separable 2-D MAP filter.

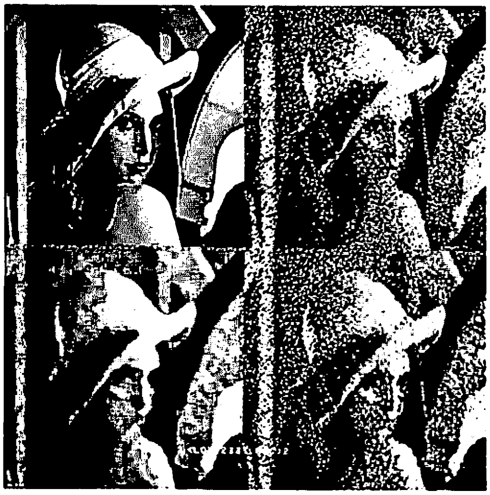


Fig. 5a. The restored image of the MAP filter with $(\overline{SNR})_{rms} = \sqrt{15}$ for nonseparable 2-D MAP filter.

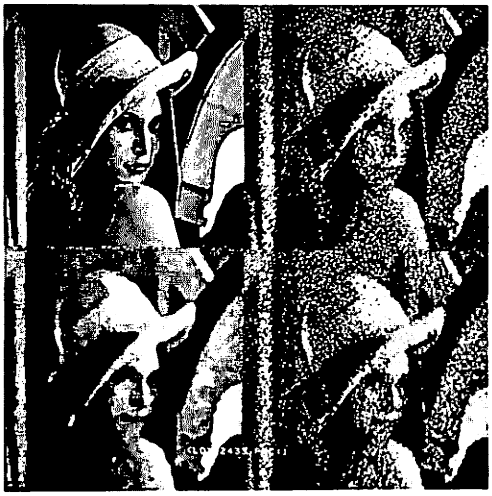


Fig. 5b. The restored image of the MAP filter with $(\overline{SNR})_{rms} = \sqrt{20}$ for nonseparable 2-D MAP filter.

matrix means space-invariant separable. When this is the case, the MAP estimate Eq. (4) can be implemented as a column processor first and then implemented as a row processor. The solution to this MAP estimate is employed the same sectioning method with Newton-Raphson technique as before. Of course, the processing time of 2-dimensional MAP filters is twice as much as that of 1-dimensional case. The experimental results are shown in Fig. 5 for different $(\overline{SNR})_{rms}$ and $\rho = 0.95$. The layout of resulting picture is as follows:

The upper left picture is an original image.

The upper right picture is a blurred and noisy image.

The lower left picture is the restored image of 1-dimensional MAP filter.

The lower right picture is the restored image of 2-dimensional separable MAP filter.

Figure 4 illustrates that the restored image of 2-dimensional separable filter is overly smoothing the Poisson noise degradation. The restored images are some improvement over blurred and noisy image. Although the separable assumption is probably good first-order approximation of well-correct linear system, the image field itself is not separable at all. Therefore, it is necessary to try 2-dimensional nonseparable MAP sectioning filter

When the PSF is a nonseparable space-invariant function and assuming R_f is the identity matrix. Using 2-dimensional nonseparable sampled infinite area superposotion operator H models 2-dimensional blurring degradation. H matrix is $M^2 \times N^2$ matrix. Where M and N are the observed data size, the processed data size of sectioning MAP filter, respectively. Thus, it needs solve N^2 order linear system equation of Eq. (4) in order to find the incremental value of the root in each iterative step. In spite of the small size of the section, it needs tremendous amount of computing time for this sectioning MAP filter.

The simulation is done with a 9 x 9 pixel section size with overlapping 4 x 4 pixels. Since blurring degradation is 3 x 3 pixels blurring, the wraparound data is 2(L-1)x2(L-1) pixels which is 4 x 4 pixels. The nonstationary mean is estimated by Rolling Window Moving Average (RWMA) method which is a very easy, fast algorithm for the 2-dimensional moving average over any size of rolling window. Because the cpu time of this sectioning MAP filter takes about 100 minutes for processing the 256 x 256 size picture, it only processes the last half size of noisy picture with two different $(\overline{SNR})_{rms}$ signals for demonstration. The experimental results are shown in Fig. 5. The layout of pictures is as follows:

The upper left picture is an original image.

The upper right picture is a blurred and noisy image.

The lower left picture is a nonstationary mean image.

The lower right picture is a restored image of MAP filter.

From Fig. 5 it is clearly seen that the noticeable better results over the restored image of Fig. 5. However, the cpu time of the nonseparable assumption is about 2 order longer than that of separable assumption. Therefore, it is trade-off between performance and computing time.

Conclusion

The MAP filter with blurring degradation for Poisson noise model has been developed. The implementation and solution to the MAP filter are heavily dependent on the scheme of blurring degradation matrix H and covariance matrix of the object image. It has been shown that the matrix means space-invariant separable. When this is the case, the MAP estimate Eq. (4) can be implemented as a column processor first and then implemented as a row processor. The solution to this MAP estimate is employed the same sectioning method with Newton-Raphson technique as before. Of course, the processing time of 2-dimensional MAP filters is twice as much as that of 1-dimensional case. The experimental results are shown in Fig. 5 for different $(\overline{SNR})_{rms}$ and $\rho = 0.95$. The layout of resulting picture is as follows:

The upper left picture is an original image.

The upper right picture is a blurred and noisy image.

The lower left picture is the restored image of 1-dimensional MAP filter.

The lower right picture is the restored image of 2-dimensional separable MAP filter.

Figure 4 illustrates that the restored image of 2-dimensional separable filter is overly smoothing the Poisson noise degradation. The restored images are some improvement over blurred and noisy image. Although the separable assumption is probably good first-order approximation of well-correct linear system, the image field itself is not separable at all. Therefore, it is necessary to try 2-dimensional nonseparable MAP sectioning filter

When the PSF is a nonseparable space-invariant function and assuming R_f is the identity matrix. Using 2-dimensional nonseparable sampled infinite area superposotion operator H models 2-dimensional blurring degradation. H matrix is $M^2 \times N^2$ matrix. Where M and N are the observed data size, the processed data size of sectioning MAP filter, respectively. Thus, it needs solve N^2 order linear system equation of Eq. (4) in order to find the incremental value of the root in each iterative step. In spite of the small size of the overlap-save sectioning method with Newton-Raphson technique is a good fast approach to find the suboptimal solution of MAP estimate. The 1-dimensional blurring and 2-dimensional blurring degradation with different Poisson noise degradations have been simulated.

From the experimental results, it has been found that the estimate nonstationary mean carries the most structured background low frequencies information and also the covariance matrix gives the higher frequency information and the stable solution of Newton-Raphson iterative method specially in the higher $\overline{(SNR)}_{rms}$ of image signals. It also has been known that the global adaptive MAP filter has an optimal weight over the best quality of image criterion, since overweight on the ML term solution will give rise to the ill-condition. From Fig. 7, it can be concluded that the quality of the restored image of MAP filter for nonseparable case is better than that of MAP filter for the separable case. However, the cpu time of the nonseparable case is much longer than that of the separable case. It is hope that the fast algorithm by the same concept of RWMA method can be developed in the future.

Nevertheless, this report has built the solid framework for image restoration of blurred images with the Poisson noise model. It has been learned that the overlap-save sectioning MAP filter with Newton-Raphson iterative technique can be used for solving larger dimensionality and nonlinearity MAP estimate equation, and that the nonstationary mean and variance cam be accurately estimated from the observation data-photon counts.

References

1. Harry L. Van Trees, Detection, Estimation, and Modulation Theory, Wiley 1958.

2. John B. Thomas, An Introduction to Statistical Communication Theory, Wiley 1959.

3. Nasser E. Nahi, Estimation Theory and Application, Wiley 1976.

4. Ralph Deutsch, Estimation Theory, Prentice-Hall 1955.

5. William K. Pratt, Digital Image Processing, Wiley Interscience, 1978.

6. Harry C. Andrews, and B.R. Hunt, Digital Image Restoration, Prentice-Hall 1977.

7. Harry C. Andrews, Tutorial and Selected Papers in Digital Image Processing, IEEE Computer Society 1978.

8. B.R. Hunt, "Bayesian Methods in Nonlinear Digital Image Restoration," IEEE Trans. on Computer, March 1977.

9. B.R. Hunt, "Approximate Error Bounds for Bayesian Methods of Nonlinear Digital Image Restoration," University of California, Los Alamos Scientific Laboratory.

10. H.J. Trussell and B.R. Hunt, "Sectional Method for Image Restoration," IEEE Trans. on ASSP,April 1978.

11. H.J. Trussell and B.R. Hunt, "Notes on Linear Image Restoration by Maximizing the A Posteriori Probability," IEEE Trans. on ASSP, April 1978.

12. H.J. Trussell, "Improved Method of Maximum A Posteriori Image Restoration," Ph.D. Dissertation, The University of New Mexico, 1976.

13. Gary K. Froehlich, John F. Walkup, Robert B. Asher, "Optimal Estimation in Signal-Dependent Noise," Texas Tech. University. Submitted to JOSA 1978.

overlap-save sectioning method with Newton-Raphson technique is a good fast approach to find the suboptimal solution of MAP estimate. The 1-dimensional blurring and 2-dimensional blurring degradation with different Poisson noise degradations have been simulated.

From the experimental results, it has been found that the estimate nonstationary mean carries the most structured background low frequencies information and also the covariance matrix gives the higher frequency information and the stable solution of Newton-Raphson iterative method specially in the higher $\overline{(\mathrm{SNR})}_{rms}$ of image signals. It also has been known that the global adaptive MAP filter has an optimal weight over the best quality of image criterion, since overweight on the ML term solution will give rise to the ill-condition. From Fig. 7, it can be concluded that the quality of the restored image of MAP filter for nonseparable case is better than that of MAP filter for the separable case. However, the cpu time of the nonseparable case is much longer than that of the separable case. It is hope that the fast algorithm by the same concept of RWMA method can be developed in the future.

Nevertheless, this report has built the solid framework for image restoration of blurred images with the Poisson noise model. It has been learned that the overlap-save sectioning MAP filter with Newton-Raphson iterative technique can be used for solving larger dimensionality and nonlinearity MAP estimate equation, and that the nonstationary mean and variance cam be accurately estimated from the observation data-photon counts.

References

1. Harry L. Van Trees, Detection, Estimation, and Modulation Theory, Wiley 1958.

2. John B. Thomas, An Introduction to Statistical Communication Theory, Wiley 1959.

14. F. Naderi and A.A. Sawchuk, "Estimation of Image Degraded by Film- Grain Noise," Applied Optics, April 15, 1978.

15. J.W. Goodman and J.F. Belsher, "Fundamental Limitations in Linear Invariant Restoration of Atmospherical Degraded Images," Proc. SPIE 75, 1976.

16. John Amoss and Frederic Davidson, "Detection of Weak Optical Images with Photon Counting Techniques," Applied Optics, August 1972.

17. C.M. Lo and A.A. Sawchuk, "Estimation of Image Signals with Poisson Noise-I," USCIPI Report 840, Sept. 1978.

18. B. Carnahan, H.A. Luther, and J.O. Wilkes, Applied Numerical Methods, John Wiley and Sons, 1969.

19. T.G. Stockham, Jr., T.M. Cannon, and P.V. Ingebretsen, "Blind Deconvolution Through Digital Signal Processing," Proc. IEEE, Vol.63, April 1974.

20. J.B. Morton, "An Investigation Into an A Posteriori Method of Image Restoration," Ph.D. Dissertation, University of Southern California, Los Angeles,Ca.

21. L.R. Rabiner and B. Gold, Theory and Application of Digital Signal Processing, Prentice Hall, 1975.

3.4 The Quality of an MAP Restoration Filter for Poisson Noise

Chun Moo Lo and A. A. Sawchuk

Introduction

We have developed the MAP estimate for image restoration with a Poisson noise model in previous reports [15]. In this report we try to investigate the quality of this MAP estimate. The quality of estimate depends on the performance criterion chosen. There are two types of performance criterion. One is that criterion used specifies the estimator structure and the other is the performance itself. The MAP estimate and MLE estimate belong to the former one. Since the MAP estimate is the mode of the a posteriori probability density and the MLE estimate is the mode of the a priori density probability. The Bayes estimate belongs to the latter one because it minimizes the risk of the estimate. Of course, the MMSE (minimize mean square error) estimate is a special case of Bayes estimate when the cost function is proportional to mean square error of the estimate. However, it is customary to choose the conditional or unconditional expected square error of estimate as an universal measure of "quality" of all estimates. Unfortunately, the expectation operation leading this measure is, in general, very complicated owing to the complexity of various estimate. However, it is possible to derive an expression for a lower bound on the variance in terms of only the statistical properties of the observed signal and estimate bias. This quality measure of any estimate without having any knowledge of the estimate itself except that it is unbiased estimate. This lower bound for the estimate error variance is well known as Cramer-Rao lower bound (CRLB).

In short, there are two quality measures of the estimate which are the expectation of the estimate and variance of the estimation error. In general, we try to find an unbiased estimate with small estimate error variance.

Biased Estimate and Unbiased Estimate

A conditional unbiased estimate is one whose expected value is equal to the true value of the quantity being estimated. An unconditional unbiased estimate is one whose expected value is equal

We have developed the MAP estimate for image restoration with a Poisson noise model in previous reports [15]. In this report we try to investigate the quality of this MAP estimate. The quality of estimate depends on the performance criterion chosen. There are two types of performance criterion. One is that criterion used specifies the estimator structure and the other is the performance itself. The MAP estimate and MLE estimate belong to the former one. Since the MAP estimate is the mode of the a posteriori probability density and the MLE estimate is the mode of the a priori density probability. The Bayes estimate belongs to the latter one because it minimizes the risk of the estimate. Of course, the MMSE (minimize mean square error) estimate is a special case of Bayes estimate when the cost function is proportional to mean square error of the estimate. However, it is customary to choose the conditional or unconditional expected square error of estimate as an universal measure of "quality" of all estimates. Unfortunately, the expectation operation leading this measure is, in general, very complicated owing to the complexity of various estimate. However, it is possible to derive an expression for a lower bound on the variance in terms of only the statistical properties of the observed signal and estimate bias. This quality measure of any estimate without having any knowledge of the estimate itself except that it is unbiased estimate. This lower bound for the estimate error variance is well known as Cramer-Rao lower bound (CRLB).

In short, there are two quality measures of the estimate which are the expectation of the estimate and variance of the estimation error. In general, we try to find an unbiased estimate with small estimate error variance.

Biased Estimate and Unbiased Estimate

A conditional unbiased estimate is one whose expected value is equal to the true value of the quantity being estimated. An unconditional unbiased estimate is one whose expected value is equal to the expected value of the quantity being estimated. Here, the estimate $\hat{X}$ is taken to be a random variable being a function of the observations Y. Therefore, if $\hat{X}$ is a conditional unbiased estimate, then

$$E_Y\hat{X} = \int \hat{X}(Y)P(Y|X)dY = X \tag{1}$$

and if $\hat{X}$ is an unconditional unbiased estimate, then

$$E_Y\hat{X} = \int \hat{X}(Y)P(Y)dY = E(X) = \overline{X} \tag{2}$$

Biased estimates, on the other hand, do not possess this desirable feature; their expected values contain an additional function B(X) of the parameter to be estimated in equation. Accordingly, for biased estimate we have

$$E_Y\hat{X} = X+B(X) \tag{3}$$

or

$$E_Y\hat{X} = \overline{X}+\overline{B(X)} \tag{4}$$

for the conditional biased estimate and the unconditional biased estimate, respectively.

$\hat{\underline{f}}_{MAP}$ is an Unconditional Unbiased Estimate Vector

From a previous report [15], we have

$$\hat{\underline{f}}_{MAP} = \overline{\underline{f}} + \lambda R_f H^T(\underline{q}-\underline{1}) \tag{5}$$

where $\hat{\underline{f}}_{MAP}$ is the N^2 x 1 estimate vector
$\overline{\underline{F}}$ is the N^2 x 1 nonstationary mean vector

R_f is the covariance matrix of image

H is the $N^2 \times N^2$ discrete blurring matrix

Taking the expectation on both sides of eq.(5) we have

$$E[\hat{\underline{f}}_{MAP}] = \overline{\underline{f}} + E[\lambda R_f H^T(\underline{q}-\underline{1})] \tag{6}$$

Since

$$\underline{q}_i = \frac{d_i}{b_i} = \frac{\lambda g_i}{b_i} ; \qquad \overline{g}_i = \lambda b_i \tag{7}$$

$$E[\underline{q}_i] = Eb_i\{Eg_i|b_i(\frac{\lambda g_i}{b_i})\} \tag{8}$$

Where $Eg_i|b_i$ denotes conditional expectation over g_i for given b_i. Eb_i denotes expectation over b_i hence

$$E[g_i] = \lambda Eb_i[\frac{\lambda b_i}{b_i}] = \alpha\lambda = 1 \tag{9}$$

Thus

$$E[\underline{q}] = \underline{1} \tag{10}$$

Substituting (6.10) into (6.6), we get

to the expected value of the quantity being estimated. Here, the estimate $\hat{X}$ is taken to be a random variable being a function of the observations Y. Therefore, if $\hat{X}$ is a conditional unbiased estimate, then

$$E_Y\hat{X} = \int \hat{X}(Y)P(Y|X)dY = X \tag{1}$$

and if $\hat{X}$ is an unconditional unbiased estimate, then

$$E_Y\hat{X} = \int \hat{X}(Y)P(Y)dY = E(X) = \overline{X} \tag{2}$$

Biased estimates, on the other hand, do not possess this desirable feature; their expected values contain an additional function B(X) of the parameter to be estimated in equation. Accordingly, for biased estimate we have

$$E_Y\hat{X} = X+B(X) \tag{3}$$

or

$$E_Y\hat{X} = \overline{X}+\overline{B(X)} \tag{4}$$

for the conditional biased estimate and the unconditional biased estimate, respectively.

$\hat{\underline{f}}_{MAP}$ is an Unconditional Unbiased Estimate Vector

From a previous report [15], we have

$$\hat{\underline{f}}_{MAP} = \overline{\underline{f}} + \lambda R_f H^T(\underline{q}-\underline{1}) \tag{5}$$

where $\hat{\underline{f}}_{MAP}$ is the N^2 x 1 estimate vector
$\underline{F}$ is the N^2 x 1 nonstationary mean vector

$$E[\hat{\underline{f}}_{MAP}] = \bar{\underline{f}} \tag{11}$$

Therefore, $\hat{\underline{f}}_{MAP}$ is an unconditional unbiased estimate vector.

Cramer-Rao Lower Bound (CRLB)

For notational and mathematical simplicity, we begin to focus our attention on the non-random scalar parameter case, for which we will drive the Cramer-Rao inequality. Then, for the random vector parameters case, the CRLB can be derived by a straightforward modification.

CRLB for non random variables case

First, we assume X is a vector unknown constants to be estimated from a sequence of measurements y(1), y(2),...,y(k) as shown in Fig. 1. where $\underline{y} = [y_1, y_2, \ldots, y_k]^T$.

Assuming $\hat{X}$ is an unbiased parameter estimate we have

$$\int \hat{X} f(\hat{X}|X) d\hat{X} = X \tag{12}$$

and from Eq. (1) we have

$$\underbrace{\int \cdots \int}_{\text{k-fold integral}} T(\underline{y}) f(\underline{y}|X) d\underline{y} = X \tag{13}$$

where $\hat{X} = T(\underline{y})$.

Using eq.(12) and finding normalized correlation coeff. between $\hat{X} = T(\underline{y})$ and $\frac{\sigma}{\sigma X} \ln f(\underline{y}|X)$ and some algebraic manipulation, then we have

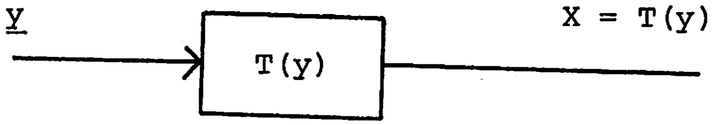


Figure 1. The block diagram of nonrandom parameter estimate

$$E[\hat{\underline{f}}_{MAP}] = \overline{\underline{f}} \tag{11}$$

Therefore, $\hat{\underline{f}}_{MAP}$ is an unconditional unbiased estimate vector.

Cramer-Rao Lower Bound (CRLB)

For notational and mathematical simplicity, we begin to focus our attention on the non-random scalar parameter case, for which we will drive the Cramer-Rao inequality. Then, for the random vector parameters case, the CRLB can be derived by a straightforward modification.

CRLB for non random variables case

First, we assume X is a vector unknown constants to be estimated from a sequence of measurements y(1), y(2),...,y(k) as shown in Fig. 1. where $\underline{y} = [y_1, y_2, \ldots, y_k]^T$.

Assuming $\hat{X}$ is an unbiased parameter estimate we have

$$\int \hat{X} f(\hat{X}|X)\,d\hat{X} = X \tag{12}$$

and from Eq. (1) we have

$$\underbrace{\int \cdots \int}_{\substack{\text{k-fold} \\ \text{integral}}} T(\underline{y}) f(\underline{y}|X)\,d\underline{y} = X \tag{13}$$

where $\hat{X} = T(\underline{y})$.

Using eq.(12) and finding normalized correlation coeff. between $\hat{X} = T(\underline{y})$ and $\frac{\sigma}{\sigma X}\ln f(\underline{y}|X)$ and some algebraic manipulation, then we have

$$\mathrm{Var}(T(y)) = \mathrm{Var}\ \hat{X} \geq \frac{1}{\mathrm{Var}\left[\dfrac{\partial \ln(\underline{y}|X)}{\partial X}\right]} \tag{14}$$

or equivalently

$$\mathrm{Var}\ \hat{X} = E[(\hat{X}-X)^2] \geq \frac{1}{\left\{\left[\dfrac{\partial \ln(\underline{y}|X)}{\partial X}\right]^2\right\}} \tag{15}$$

Eq.(15) is called Cramer-Rao inequality for the unbiased estimate. Note that CRLB is a bound on the mean-square error.

<u>CRLB for Random Variable Vector Case</u>

For the random variable vector case, the information matrix J_T now consists of two parts

$$J_T \triangleq J_D + J_P \tag{16}$$

where

$$J_P \triangleq E(\{\nabla_{\underline{X}}[\ln P(\underline{y}|\underline{X})]\}\{\nabla_{\underline{X}}[\ln P(\underline{y}|\underline{X})]\}^T) \tag{17}$$

$$J_P \triangleq E(\{\nabla_{\underline{X}}[\ln P(\underline{X})]\}\{\nabla_{\underline{X}}[\ln P(\underline{X})]\}^T) \tag{18}$$

The matrix J_D is the information matrix which represents information obtained from the data or from the a priori density $P(\underline{y}|\underline{x})$ of MAP estimate. The matrix J_P is the information matrix which represents information obtained from the a priori information. The correlation matrix of the error is

$$R_\varepsilon \overset{\Delta}{=} E(\underline{X}_\varepsilon \underline{X}_\varepsilon^T) \tag{19}$$

Where $\underline{X}_\varepsilon = (\underline{\hat{X}}-\underline{X})$. The diagonal elements represent the mean-square errors and the off diagonal elements are the cross correlations. The mean-square error of the estimate is related to the information matrix as follows

$$E[X_{\varepsilon_i}^2] \geq J_T^{ii} \tag{20}$$

The diagonal elements in the inverse of the total information matrix J_T are the lower bounds on the corresponding mean-square errors. This is the case in which we are interested.

CRLB of MAP Estimate for Poisson Noise Model

The estimate error covariance, in general, is very complicated to find due to the complexity of the posteriori density. However, for the MAP estimate it is possible to derive an expression for a lower bound on the variance since we know the a priori density $P(\underline{y}|\underline{X})$ and probability density of $\underline{X}$ $P(\underline{X})$. From Eq.(16), we have

$$J_T = J_D + J_P \tag{21}$$

where J_D and J_P are defined in eq.(17) and eq.(18) respectively. We then have

The matrix J_D is the information matrix which represents information obtained from the data or from the a priori density $P(\underline{y}|\underline{x})$ of MAP estimate. The matrix J_P is the information matrix which represents information obtained from the a priori information. The correlation matrix of the error is

$$R_\varepsilon \triangleq E(\underline{X}_\varepsilon \underline{X}_\varepsilon^T) \tag{19}$$

Where $\underline{X}_\varepsilon = (\hat{\underline{X}}-\underline{X})$. The diagonal elements represent the mean-square errors and the off diagonal elements are the cross correlations. The mean-square error of the estimate is related to the information matrix as follows

$$E[X_{\varepsilon_i}^2] \geq J_T^{ii} \tag{20}$$

The diagonal elements in the inverse of the total information matrix J_T are the lower bounds on the corresponding mean-square errors. This is the case in which we are interested.

CRLB of MAP Estimate for Poisson Noise Model

The estimate error covariance, in general, is very complicated to find due to the complexity of the posteriori density. However, for the MAP estimate it is possible to derive an expression for a lower bound on the variance since we know the a priori density $P(\underline{y}|\underline{X})$ and probability density of $\underline{X}$ $P(\underline{X})$. From Eq.(16), we have

$$J_T = J_D + J_P \tag{21}$$

where J_D and J_P are defined in eq.(17) and eq.(18) respectively. We then have

$$J_P = R_f^{-1} \tag{22}$$

$$J_D = \lambda^2 H^T R_{\underline{q}} H \tag{23}$$

where

$$R_{\underline{q}} = E\{(\underline{q}-\overline{\underline{q}})(\underline{q}-\overline{\underline{q}})^T\}$$

Hence

$$J_T \triangleq \lambda^2 H^T R_{\underline{q}} H + R_f^{-1} \tag{24}$$

This is total information matrix J_T of the MAP estimate for Poisson noise model.

When J_T^{-1} exists, from eq.(24), we have

$$E[(\hat{f}_i - f_i)^2] \geq \{J_T^{-1}\}_{ii} \tag{25}$$

where $\{J_T^{-1}\}_{ii}$ is diagonal element of J_T^{-1} and $\hat{f}_i$ is an estimate of the ith component of the restored image vector $\underline{f}_i$. Inspection of eq.(25) indicates the error bound depends on four quantities: rate function constant λ, discrete blurring matrix H, covariance matrix $R_{\underline{q}}$, and covariance matrix of image R_f. If we assume that

$R_{\underline{q}} = \sigma_{\underline{q}}^2 I$ and $\underline{R}_f = \sigma_f^2 I$ then we get

$$J_T^{-1} = [(\lambda^2 H^T H \sigma_{\underline{q}}^2 I) + (\sigma_f^2)^{-1} I]^{-1} \tag{26}$$

$$= [(\lambda^2 \|H\|^2 \sigma_{\underline{q}}^2 I) + (\sigma_f^2)^{-1} I]^{-1}$$

then

$$E[(\hat{f}_i - f_i)^2] = \frac{\sigma_f^2}{\lambda^2 \|H\|^2 \sigma_{\underline{q}}^2 \sigma_f^2 + 1} \tag{27}$$

We now rewrite eq.(27) to obtain

$$E[(f_i - \hat{f}_i)^2] = \frac{1}{\lambda^2 \|H\|^2 \sigma_{\underline{q}}^2 + \frac{1}{\sigma_f^2}} \tag{28}$$

From eq.(28), we have observed:

(1). When σ_f^2 is larger value, then we can approximate eq.(28) as

$$E[(f_i - \hat{f}_i)^2] \approx \frac{(\overline{b}_i)^2}{(\overline{\lambda b_i})^2 \|H\|^2 \sigma_{\underline{q}}^2} \tag{29}$$

where $\overline{b}_i$ is mean blurred object intensity of ith pixel and $(\lambda \overline{b}_i)^{\frac{1}{2}} = (\overline{SNR})_{r.m.s.}$. That means that λ , H, and σ_q^2 play more important

$$J_P = R_f^{-1} \tag{22}$$

$$J_D = \lambda^2 H^T R_{\underline{q}} H \tag{23}$$

where

$$R_{\underline{q}} = E\{(\underline{q}-\overline{\underline{q}})(\underline{q}-\overline{\underline{q}})^T\}$$

Hence

$$J_T \triangleq \lambda^2 H^T R_{\underline{q}} H + R_f^{-1} \tag{24}$$

This is total information matrix J_T of the MAP estimate for Poisson noise model.

When J_T^{-1} exists, from eq.(24), we have

$$E[(\hat{f}_i - f_i)^2] \geq \{J_T^{-1}\}_{ii} \tag{25}$$

where $\{J_T^{-1}\}_{ii}$ is diagonal element of J_T^{-1} and $\hat{f}_i$ is an estimate of the ith component of the restored image vector $\underline{f}_i$. Inspection of eq.(25) indicates the error bound depends on four quantities: rate function constant λ, discrete blurring matrix H, covariance matrix $R\underline{q}$, and covariance matrix of image R_f. If we assume that

$R_{\underline{q}} = \sigma_{\underline{q}}^2 I$ and $\underline{R}_f = \sigma_f^2 I$ then we get roles in the error bound of the Poisson noise model than the variance of the object σ_f^2.

(2). The error bound is inversely proportional to the square of the ensemble mean rate function $\lambda\bar{b}_i$. From Fig. 2, we can see that the Poisson noise degradation decreases rapidly as $\lambda\bar{b}_i$ increases. Consequently, the $(\overline{S/N})_{r.m.s.}$ is larger than 10 db or equal to 10 db (i.e. $\lambda\bar{b}_i \geq 100$), then

$$E[(f_i-\hat{f}_i)^2] \geq 10^{-4} \qquad (30)$$

Thus, the Poisson noise degradation effect disappears in a practical sense.

The layout of Fig. 2 is as follows:
The upper left picture is the Poisson noisy image with $\overline{\lambda b}_i=2.5$
The upper right picture is the Poisson noisy image with $\overline{\lambda b}_i=5$
The lower left picture is the Poisson noisy image with $\overline{\lambda b}_i=10$
The lower right picture is the Poisson noisy image with $\overline{\lambda b}_i=20$

(3). The error bound is inversely proportional to the properties of the point spread function H; the error due to noise is "amplified" by the point spread function. This is the result of ill-conditioning in the restoration process.

Conclusion

We can conclude that the MAP estimate for Poisson noise model is an unbiased estimate and that its estimation error variance lower bound (CRLB) is defined by eq.(35).

The use of statistical estimation is particularly desirable from the viewpoint of error analysis, since known techniques can be applied to compute the error bound. We have developed the CRLB for the Poisson noise model and also shown the behavior of the CRLB

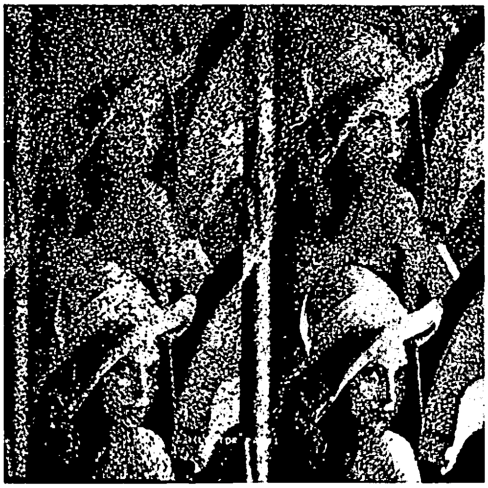


Fig. 2. Detected Image with different $\overline{(\lambda b_i)}$

approximation. From these facts, we are able to find from various algorithms which error bound is closer to the CRLB. It must be remembered that the CRLB is a lower bound and the actual restoration error will be greater. It is possible that a better suboptimal algorithm for the sectioned MAP estimate can be found to reduce the actual restoration error closer to the Cramer-Rao lower bound.

References

1. H.L. Van Trees, Detection, Estimation, and Modulation Theory, Wiley, 1968.

2. J.B. Thomas, An Introduction Statistical Communication Theory, Wiley, 1969.

3. N.E. Nahi, Estimation Theory and Application, Wiley, 1976.

4. Ralph Deutsch, Estimation Theory, Prentice-Hall, 1965.

5. D. Middleton, An Introduction to Statistical Communication Theory, McGraw-Hill, 1960.

6. A. Papoulis, Probability, Random Variables, and Stochastic Processes, McGraw-Hill, 1965.

7. W.K. Pratt, Digital Image Processing, John Wiley and Sons, 1978.

8. H.C. Andrews and B.R. Hunt, Digital Image Restoration, Prentice-Hall, 1977.

9. H.C. Andrews, Tutorial and Selected Papers in Digital Image Processing, IEEE Computer Society, 1978.

10. B.R. Hunt, "Bayesian Methods in Nonlinear Digital Image Restoration", IEEE Transactions on Computer, March 1977.

11. B.R. Hunt, "Approximate Error Bounds for Bayesian Methods of Nonlinear Digital Image Restoration", University of California, Los Alamos Scientific Laboratory.

12. H.J. Trussell and B.R. Hunt, "Sectional Method for Image Restoration", IEEE Transaction on ASSP, April, 1978.

13. H.J. Trussell and B.R. Hunt, "Note on Linear Image Restoration by Maximizing the A Posteriori Probability", IEEE Transactions on ASSP, April, 1978.

14. J.W. Goodman and J.F. Belsher, "Fundamental Limitations in Linear Invariant Restoration of Atmospherical Degraded Image", Proc. SPIE 75, 1976.

15. C.M. Lo and A.A. Sawchuk, "Estimation of Blurred Image Signals with Poisson Noise Model", University of Southern California, IPI Report 860, March 1979.

3.5 An Approach of A Posteriori Image Restoration

David D. Garber and John B. Morton

The Method

The problem of a posteriori image restoration, that is, the problem of image restoration without a priori knowledge of the nature of the blur, is a difficult problem at best. Neglecting the problem of restoring images degraded by atmospheric turbulence, research in this area has not been extensive and is fairly well contained in references [1-11].

approximation. From these facts, we are able to find from various algorithms which error bound is closer to the CRLB. It must be remembered that the CRLB is a lower bound and the actual restoration error will be greater. It is possible that a better suboptimal algorithm for the sectioned MAP estimate can be found to reduce the actual restoration error closer to the Cramer-Rao lower bound.

References

1. H.L. Van Trees, Detection, Estimation, and Modulation Theory, Wiley, 1968.

2. J.B. Thomas, An Introduction Statistical Communication Theory, Wiley, 1969.

3. N.E. Nahi, Estimation Theory and Application, Wiley, 1976.

4. Ralph Deutsch, Estimation Theory, Prentice-Hall, 1965.

5. D. Middleton, An Introduction to Statistical Communication Theory, McGraw-Hill, 1960.

6. A. Papoulis, Probability, Random Variables, and Stochastic Processes, McGraw-Hill, 1965.

7. W.K. Pratt, Digital Image Processing, John Wiley and Sons, 1978.

8. H.C. Andrews and B.R. Hunt, Digital Image Restoration, Prentice-Hall, 1977.

9. H.C. Andrews, Tutorial and Selected Papers in Digital Image Processing, IEEE Computer Society, 1978.

10. B.R. Hunt, "Bayesian Methods in Nonlinear Digital Image Restoration", IEEE Transactions on Computer, March 1977.

The approach presented herein makes two assumptions. The first assumption is that the blurred image and the unblurred image are related via a convolution integral. That is,

$$g(x,y) = \iint_{-\infty}^{\infty} h(x-\varepsilon,y-\eta)\,f(\varepsilon,\eta)\,d\varepsilon d\eta + n(x,y),$$

where $g(x,y)$ denotes the blurred image, $h(x-\xi,y-\eta)$ denotes a spatially invariant point spread function, $f(\xi,\eta)$ denotes the unblurred image, and $n(x,y)$ denotes additive noise. The second assumption is that an estimate can be made of the unblurred image of some object in the blurred image. For example, if the blurred image contains a sign and some of the letters of the blurred sign are discernible, then one can estimate the unblurred image of one of these letters.

Let $\hat{f}_1(i,j)$ denote an estimate of the unblurred image of an object contained in the blurred image. Note that $\hat{f}_1(i,j)$ now denotes a digital image. Let $g_1(i,j)$ denote the subimage of the blurred image corresponding to $\hat{f}_1(i,j)$.

We will now find a convolutional restoring filter which relates $g_1(i,j)$ to $\hat{f}_1(i,j)$ in some optimal sense. Once the filter has been defined and since the blur is assumed to be a spatially invariant blur, the restoring filter can be applied to the entire blurred image to estimate the entire unblurred image.

Mathematically, the problem is to find the restoring coefficients $a(k,\ell)$ such that

$$\sum_{k=-K}^{K} \sum_{\ell=-L}^{L} a(k,\ell) g(i-k, j-\ell)$$

estimates f(i,j) in some optimal sense. The optimal sense that we have considered is minimum squared relative error. This criterion is more highly correlated with one's visual perception than the more common least squares criterion [12].

Thus, the problem is to minimize over a(k,1),

$$\sum_{i} \sum_{j} \left[\frac{\sum_{k=-K}^{K} \sum_{\ell=-L}^{L} a(k,\ell) g_1(i-k,j-\ell) - \hat{f}_1(i,j)}{\hat{f}_1(i,j)} \right]^2 \tag{1}$$

for $\hat{f}_1(i,j) > 0$.

Taking partial derivatives with respect to a(m,n) and setting the results to zero,

$$\sum_{i} \sum_{j} \frac{2g_1(i-m,j-n)}{\hat{f}_1(i,j)} \left[\frac{\sum_{k} \sum_{\ell} a(k,\ell) g_1(i-k,j-\ell) - \hat{f}_1(i,j)}{\hat{f}_1(i,j)} \right] = 0$$

$$m=-K,-K+1,\ldots,K \qquad \text{and} \qquad n=-L,-L+1,\ldots,L.$$

Rearranging,

The approach presented herein makes two assumptions. The first assumption is that the blurred image and the unblurred image are related via a convolution integral. That is,

$$g(x,y) = \iint\limits_{-\infty}^{\infty} h(x-\varepsilon,y-\eta)f(\varepsilon,\eta)d\varepsilon d\eta+n(x,y),$$

where $g(x,y)$ denotes the blurred image, $h(x-\xi,y-\eta)$ denotes a spatially invariant point spread function, $f(\xi,\eta)$ denotes the unblurred image, and $n(x,y)$ denotes additive noise. The second assumption is that an estimate can be made of the unblurred image of some object in the blurred image. For example, if the blurred image contains a sign and some of the letters of the blurred sign are discernible, then one can estimate the unblurred image of one of these letters.

Let $\hat{f}_1(i,j)$ denote an estimate of the unblurred image of an object contained in the blurred image. Note that $\hat{f}_1(i,j)$ now denotes a digital image. Let $g_1(i,j)$ denote the subimage of the blurred image corresponding to $\hat{f}_1(i,j)$.

We will now find a convolutional restoring filter which relates $g_1(i,j)$ to $\hat{f}_1(i,j)$ in some optimal sense. Once the filter has been defined and since the blur is assumed to be a spatially invariant blur, the restoring filter can be applied to the entire blurred image to estimate the entire unblurred image.

Mathematically, the problem is to find the restoring coefficients $a(k,\ell)$ such that

$$\sum_{i}\sum_{j}\sum_{k}\sum_{\ell} \frac{g_1(i-m,j-\ell)a(k,\ell)g_1(i-k,j-\ell)}{\hat{f}_1^{\,2}(i,j)} = \sum_{i}\sum_{j} \frac{g_1(i-m,j-n)}{\hat{f}_1(i,j)} \quad ; \tag{2}$$

$$\sum_{k}\sum_{\ell} a(k,\ell) \sum_{i}\sum_{j} \frac{g_1(i-m,j-n)g_1(i-k,j-\ell)}{\hat{f}_1^{\,2}(i,j)} = \sum_{i}\sum_{j} \frac{g_1(i-m,j-n)}{\hat{f}_1(i,j)} \quad .$$

Note that equation (2) is a linear algebraic system in the (2K+1)(2L+1) unknowns $a(k,\ell)$ and can be solved in a straightforward manner. The restoring filter is then applied to the entire blurred image.

Experimental Results

To determine the effectiveness of the above ideas under the most ideal conditions, two simulations were performed. The simulations assumed perfect knowledge of the unblurred image of an object in the blurred images. In each of the two simulations the image of Figure 1 was blurred within the computer, and a 64x64 pixel subimage centered about the face of the man in Figure 1 was assumed known. This subimage would represent an ideal $\hat{f}_1(i,j)$ in equation (2). After solving equation (2) for the coefficients of the restoring convolutional filters, the restoring filters were applied to the entire blurred images to estimate the unblurred images.

The first simulation assumed the point spread function of Figure 2, a triangularly shaped point spread function extending over an area of 7x7 pixels. Displayed in Figure 3a are the results of blurring

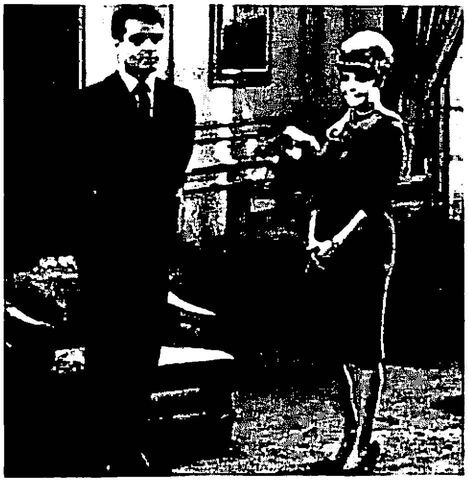


Figure 1

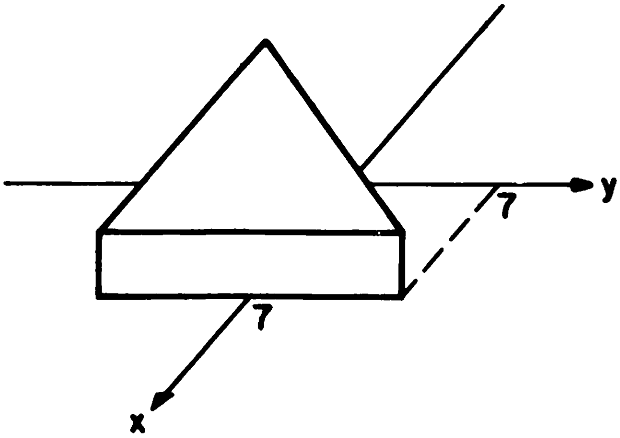


Figure 2

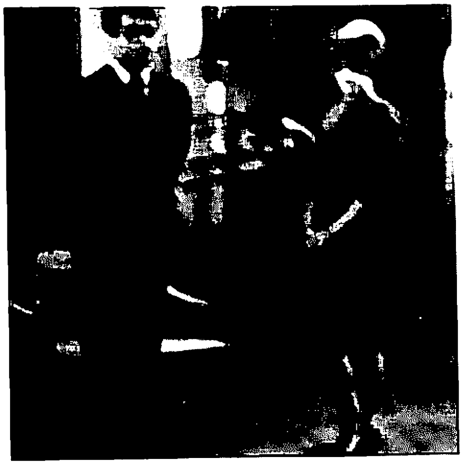


Figure 3a

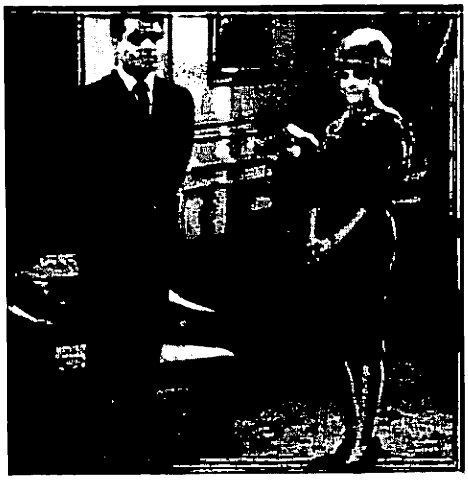


Figure 3b

Figure 1 with the point spread function of Figure 2; Figure 3b is the corresponding restoration using a 15x15 pixel restoring filter.

The second simulation assumed the Gaussian shaped point spread function illustrated in Figure 4. In Figure 5a is the resulting blurred image and in Figure 5b, the associated restored image. Again, the restoring filter was of an extent of 15x15 pixels.

Figure 6a contains a camera-induced blurred image. This is not a simulation. In this case, the exact nature of the blur and the exact nature of the unblurred image is unknown. Extensive experience with this particular image indicated that the degradation is considerable, and that the blur appears to be a combination of a diagonal motion blur and defocus. Using a restoring Wiener filter corresponding to various combinations of a diagonal motion blur and/or defocus, produced restorations less than satisfactory. In contrast, constructing an image of an "S" estimated to be ideal (unblurred) and using the image of the "S" in equation (2) as $\hat{f}_1(i,j)$, a restoring filter was calculated via equation (2). Applying the restoring filter to the image in Figure 6a, the restoration in Figure 6b resulted.

Conclusion

A method of a posteriori image restoration has been presented. The method makes two assumptions: 1) the blur is a spatially invariant blur, and 2) the unblurred image of some object in the blurred image can be estimated. The simulations demonstrated that, given an ideal estimate of the unblurred image of some object in the blurred image, the method provides quite good restorations. For the case of a camera-induced blur, the results will be contingent upon the degree to which the unblurred image of some object in the blurred image can be estimated. The results of Figure 6a and 6b illustrated that satisfactory results can be achieved via this method.

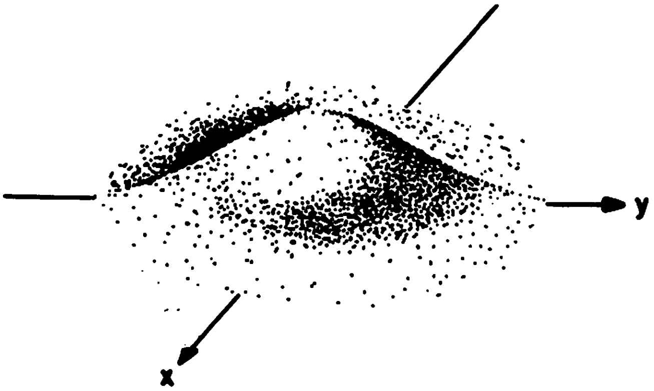


Figure 4

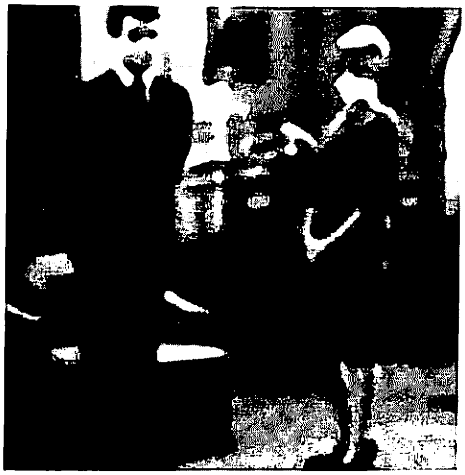


Figure 5a

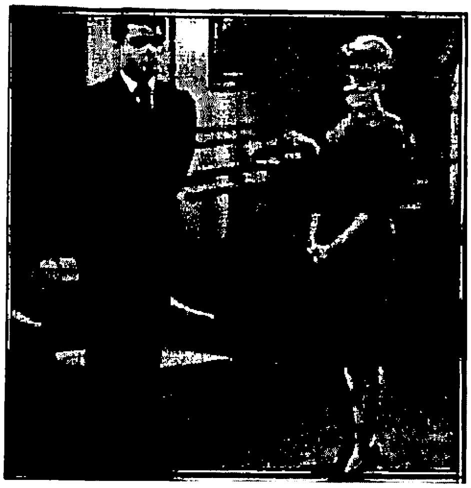


Figure 5b

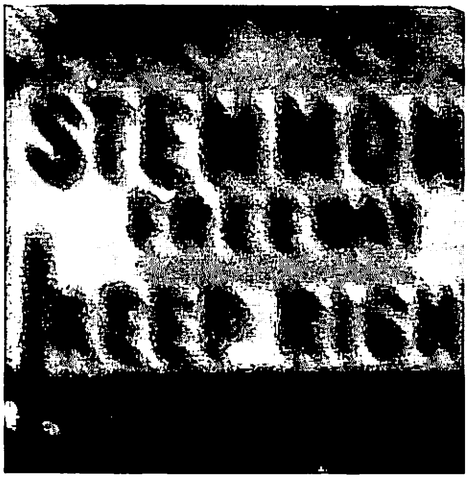


Figure 6a

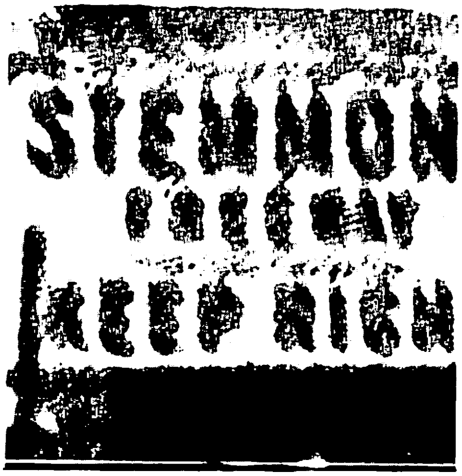


Figure 6b

References

1. Tatian, B., "Method for Obtaining the Transfer Function from the Edge Response Function," J. Opt. Soc. Amer., Vol. 55, No. 8, August 1965, pp. 1014-1019.

2. Gennery, D.B., "Determination of Optical Transfer Function by Inspection of Frequency-Domain Plot," J. Opt. Soc. Am., Vol. 63, No. 12, Dec. 1973, pp. 1571-1577.

3. Honda,T., Y. Kukushima and J. Tsujiuchi, "An Estimation of the Amount of Motion in a Linear Motion Blurred Picture by Spectrum Analysis," Optica Acta, Vol. 23, No. 10, Oct. 1976, pp. 799-811.

4. Maitre, H., "Defect Recognition in Numerical Images by Spectrum Zero Detection." Computer Graphics and Image Processing, Vol. 5, No. 2, June 1976, pp. 238-244.

5. Filip, A., "Estimating the Impulse Response of Linear, Shift-Invariant Image Degrading Systems," Ph.D. Thesis, M.I.T., October 1972.

6. Cole, E.R., "The Removal of Unknown Image Blurs by Homomorphic Filtering," Dept. of Computer Science, University of Utah, ARPA Technical Report UTEC- CSC-74-029, June 1973.

7. Cannon, T.M., "Digital Image Deblurring by Nonlinear Homomorphic Filtering," Dept. of Computer Science, University of Utah, ARPA Technical Report UTEC-CSC-74-091, August 1974.

8. Stockham, T.G., T.M. Cannon and R.B. Ingebretsen, "Blind Deconvolution Through Digital Signal Processing," Proc. IEEE, Vol. 63, No. 4, April 1975, pp. 678-692.

9. Cannon, T.M., "Blind Deconvolution of Spatially Invariant Image Blurs with Phase," IEEE Trans. ASSP, Vol. ASSP-24, No. 1, Feb.

1976, pp. 58-63.

10. Morton, J.B., and H.C. Andrews, "An A Posteriori Method of Image Restoration," JOSA (to appear 1979).

11. Hyman, J.M., and J.G. Sanderson, "A Local Method for Digitally Restoring Motion-Blurred Images," IEEE Trans. ASSP, Vol. ASSP-26, No. 3, June 1978, pp. 256-263.

12. Pratt, W.K., Digital Image Processing, Wiley-Interscience, New York, New York, 1978.

3.6 Errors in Polar Coordinate Sampling

Yeh-Hua Peter Chuan

Introduction

In many of the new imaging systems that are propping up, more and more of these seem to collect their data samples with a polar coordinate format. Most of these systems involve obtaining projections of the object and reconstructing the image from these projections. Since only a finite number of samples can be read from each projection, the polar coordinate sampling format is built into the system. As an example, these include radio astronomy, electron microscopy, x-ray tomography, optical imaging, radar imaging, and so on.

The first attempt to estimate the sampling requirements in polar coordinates appeared in 1967 [3] in which the maximum linear distance between any two adjacent samples in the Fourier transform domain was chosen so that its inverse was greater than the maximum diameter of

References

1. Tatian, B., "Method for Obtaining the Transfer Function from the Edge Response Function," J. Opt. Soc. Amer., Vol. 55, No. 8, August 1965, pp. 1014-1019.

2. Gennery, D.B., "Determination of Optical Transfer Function by Inspection of Frequency-Domain Plot," J. Opt. Soc. Am., Vol. 63, No. 12, Dec. 1973, pp. 1571-1577.

3. Honda,T., Y. Kukushima and J. Tsujiuchi, "An Estimation of the Amount of Motion in a Linear Motion Blurred Picture by Spectrum Analysis," Optica Acta, Vol. 23, No. 10, Oct. 1976, pp. 799-811.

4. Maitre, H., "Defect Recognition in Numerical Images by Spectrum Zero Detection." Computer Graphics and Image Processing, Vol. 5, No. 2, June 1976, pp. 238-244.

5. Filip, A., "Estimating the Impulse Response of Linear, Shift-Invariant Image Degrading Systems," Ph.D. Thesis, M.I.T., October 1972.

6. Cole, E.R., "The Removal of Unknown Image Blurs by Homomorphic Filtering," Dept. of Computer Science, University of Utah, ARPA Technical Report UTEC- CSC-74-029, June 1973.

7. Cannon, T.M., "Digital Image Deblurring by Nonlinear Homomorphic Filtering," Dept. of Computer Science, University of Utah, ARPA Technical Report UTEC-CSC-74-091, August 1974.

8. Stockham, T.G., T.M. Cannon and R.B. Ingebretsen, "Blind Deconvolution Through Digital Signal Processing," Proc. IEEE, Vol. 63, No. 4, April 1975, pp. 678-692.

9. Cannon, T.M., "Blind Deconvolution of Spatially Invariant Image Blurs with Phase," IEEE Trans. ASSP, Vol. ASSP-24, No. 1, Feb.

the object. This intuitively obtained result was very accurate. Smith et al. [4] in 1973 computed the Fourier transform of a Gaussian blob sampled in polar coordinates. It was found that besides another Gaussian blob that was obtained after the Fourier transformation, a series of "clutter" terms associated with the blob also appeared.

The objective of this study is to obtain an analytic expression for the errors or "clutters" associated with sampling in polar coordinate format and therefore try to determine exactly the necessary and sufficient sampling rate in both azimuth and radial dimensions. Our approach is as follows: We will sample a disc and an annula ring pupil in polar coordinate sampling and compute its Fourier transform which will be called "discrete" point spread function. Since these functions are isotropic, we will mention the transformation as Fourier Bessel Transform. We will apply Poisson's summation formula to compute the discrete point spread function and get an expression for the difference between the discrete transform and continuous transform. This difference is the error associated with the sampling.

A very significant immediate application of this study is on the estimation of the azimuth and radial sampling interval for x-ray tomographic systems which use trial and error to find an "optimum" number of azimuth samples or projections. Since each projection exposes the patient with an extra dose of radiation, it is extremely important to know the minimum number of projections that is needed to get a reconstruction that is free of sampling errors.

II. Poisson's Summation Formula

$$\sum_{m=0}^{N-1} f_m = \frac{1}{\Delta}\int_0^{N\Delta} f(x)\,dx + \frac{2}{\Delta}\sum_{n=1}^{\infty}\int_0^{N\Delta} f(x)\cos\left(\frac{2\pi}{\Delta}nx\right)dx + \frac{1}{2}[f_0 - f_N] \qquad (1)$$

where

$f_m = f(m\Delta)$

Δ = some arbitrary increment of x

N = total number of increments over which the integration is carried out.

This is Poisson's Summation Formula. It says that if we approximate an integral $\int_0^{N\Delta} f(x)dx$ by a linear sum of samples, the error incurred in the approximation will be an infinite sum of error terms, the n^{th} order of which being

$$e_n = \frac{2}{\Delta}\int_0^{N\Delta} f(x)\cos(\frac{2\pi}{\Delta}nx)\,dx$$

A slightly modified version that will be handy is as follows.

$$\sum_{m=-N+1}^{N-1} f_m = \frac{1}{\Delta}\int_{-(N-1)\Delta}^{(N-1)\Delta} f(x)\,dx + \sum_{n=1}^{\infty}\frac{2}{\Delta}\int_{-(N-1)\Delta}^{(N-1)\Delta} f(x)\cos(\frac{2\pi}{\Delta}nx)\,dx + \frac{1}{2}\left[f_{-N+1}+f_{N-1}\right] \tag{2}$$

These two formulae will be very powerful in the following sections.

III. Azimuth Sampling

Suppose we sample a unit circle with N samples as shown in figure 1. The Fourier Bessel Transform of the unit circle is the object. This intuitively obtained result was very accurate. Smith et al. [4] in 1973 computed the Fourier transform of a Gaussian blob sampled in polar coordinates. It was found that besides another Gaussian blob that was obtained after the Fourier transformation, a series of "clutter" terms associated with the blob also appeared.

The objective of this study is to obtain an analytic expression for the errors or "clutters" associated with sampling in polar coordinate format and therefore try to determine exactly the necessary and sufficient sampling rate in both azimuth and radial dimensions. Our approach is as follows: We will sample a disc and an annula ring pupil in polar coordinate sampling and compute its Fourier transform which will be called "discrete" point spread function. Since these functions are isotropic, we will mention the transformation as Fourier Bessel Transform. We will apply Poisson's summation formula to compute the discrete point spread function and get an expression for the difference between the discrete transform and continuous transform. This difference is the error associated with the sampling.

A very significant immediate application of this study is on the estimation of the azimuth and radial sampling interval for x-ray tomographic systems which use trial and error to find an "optimum" number of azimuth samples or projections. Since each projection exposes the patient with an extra dose of radiation, it is extremely important to know the minimum number of projections that is needed to get a reconstruction that is free of sampling errors.

II. Poisson's Summation Formula

$$\sum_{m=0}^{N-1} f_m = \frac{1}{\Delta}\int_0^{N\Delta} f(x)\,dx + \frac{2}{\Delta}\sum_{n=1}^{\infty}\int_0^{N\Delta} f(x)\cos(\frac{2\pi}{\Delta}nx)\,dx + \frac{1}{2}[f_0-f_N] \tag{1}$$

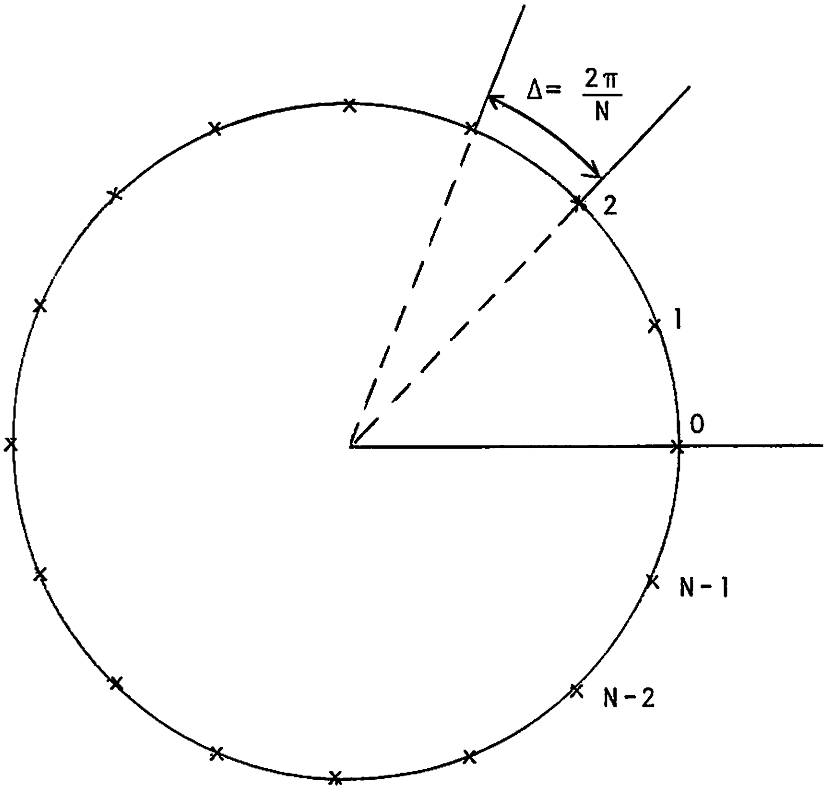


Figure 1. Angular sampling on a unit circle.

$$\frac{1}{2\pi}\int_0^{2\pi} 1 \cdot e^{j2\pi\rho\cos\theta} d_\theta = J_0(2\pi\rho)$$

and by Poisson's summation formula, the discrete version is

$$\frac{1}{N}\sum_{i=0}^{N-1} e^{j2\pi\rho\cos(\frac{i}{N}2\pi)} = J_0(2\pi\rho)+\sum_{n=1}^{\infty} e_n(\rho) \qquad (3)$$

where

$$e_n(\rho) = \frac{1}{\pi}\int_0^{2\pi} e^{j2\pi\rho\cos}\rho\cos(nN\theta)d\theta \qquad (4)$$

hence

$$e_n(\rho) = 2(-1)^{nN'}J_{nN}(2\pi\rho) \qquad (5)$$

We have assumed that N=2N '. $e_n(\rho)$ in (5) represents the azimuth sampling error.

Since N is usually large, we need only to look at the properties of Bessel function of large orders. For large orders N, $J_{nN}(z)$ is negligibly small compared to its first peak for $0 \leq z<nN$. The first peak is the most dominant one. Over (0,nN), $J_{nN}(z)$ is monotonic increasing. From [5], the first peak of $J_{nN}(z)$ occurs at

$$\rho = P_n = \frac{N}{2\pi}n + \left(\frac{0.8086N^{\frac{1}{3}}}{2\pi}\right)n^{\frac{1}{3}} + 0\left((nN)^{-\frac{1}{3}}\right) \qquad (6)$$

$$\frac{1}{2\pi}\int_{0}^{2\pi} 1 \cdot e^{j2\pi\rho\cos\theta} d_{\theta} = J_0(2\pi\rho)$$

and by Poisson's summation formula, the discrete version is

$$\frac{1}{N}\sum_{i=0}^{N-1} e^{j2\pi\rho\cos(\frac{i}{N}2\pi)} = J_0(2\pi\rho)+\sum_{n=1}^{\infty} e_n(\rho) \qquad (3)$$

where

$$e_n(\rho) = \frac{1}{\pi}\int_{0}^{2\pi} e^{j2\pi\rho\cos}\rho\cos(nN\theta)\,d\theta \qquad (4)$$

hence

$$e_n(\rho) = 2(-1)^{nN'}J_{nN}(2\pi\rho) \qquad (5)$$

We have assumed that N=2N '. $e_n(\rho)$ in (5) represents the azimuth sampling error.

Since N is usually large, we need only to look at the properties of Bessel function of large orders. For large orders N, $J_{nN}(z)$ is negligibly small compared to its first peak for $0 \leq z<nN$. The first peak is the most dominant one. Over (0,nN), $J_{nN}(z)$ is monotonic increasing. From [5], the first peak of $J_{nN}(z)$ occurs at

$$\rho = P_n = \frac{N}{2\pi}n + \left(\frac{0.8086N^{\frac{1}{3}}}{2\pi}\right)n^{\frac{1}{3}}+ 0\left((nN)^{-\frac{1}{3}}\right) \qquad (6)$$

Table 1 shows values of P_n computed for various values of n for N=256 using the linear approximation and the two term approximation of (6). Values of $J_{nN}(P_n)$ are also computed. This shows that the azimuth sampling error can become very significant compared to the main lobe peak of $J_0(0)=1$ if N is not chosen large enough. Also, even without linear approximation, P_n is still very closely equal to nP with $P \approx 41$. Figure 2 shows a plot of the exact transform $J_0(2\pi\rho)$ and 1st, 2nd, 3rd, 4th order azimuth sampling errors.

I. Radial Sampling

Figure 3 shows the disc pupil function discretely sampled in the radial dimension. It is

$$G(f) = \Delta f \sum_{k=0}^{K-1} \delta(f-k\Delta f) \tag{7}$$

where $\Delta f = f_{BW}/K-1$. The exact Fourier Bessel transform of a disc pupil of radius f_{BW} is

$$B\left\{\mathrm{circ}(f/f_{BW})\right\} = f_{BW}\,\frac{J_1(2\pi f_{BW}\rho)}{\rho} \tag{8}$$

Substituting (7) into the Fourier Bessel Transform equation, the discrete point spread function is

$$\pi\Delta f \sum_{k=-K+1}^{K-1} |k|\, J_0\!\left(\frac{2\pi}{K-1} f_{BW}\rho k\right)$$

Table 1

$N=256,\ J_0(0) = 1$

n	$P_n = \left(\frac{N}{2\pi}\right)n+\left(\frac{0.8086N^{\frac{1}{3}}}{2\pi}\right)n^{\frac{1}{3}}$	Linear approx. $\frac{nN}{2\pi}$	Percentage error	Clutter $2J_{nN}(nN)$	Value of 1st peak $J_{nN}(P_n)$
1	41.561	40.74	2.0%	0.140	0.211
2	82.517	81.49	1.3%	0.112	0.168
3	123.410	122.23	1.0%	0.098	0.147
4	164.272	162.97	0.8%	0.088	0.134
5	205.116	203.72	0.7%	0.082	0.124

Table 1 shows values of P_n computed for various values of n for N=256 using the linear approximation and the two term approximation of (6). Values of $J_{nN}(P_n)$ are also computed. This shows that the azimuth sampling error can become very significant compared to the main lobe peak of $J_0(0)=1$ if N is not chosen large enough. Also, even without linear approximation, P_n is still very closely equal to nP with $P \approx 41$. Figure 2 shows a plot of the exact transform $J_0(2\pi\rho)$ and 1st, 2nd, 3rd, 4th order azimuth sampling errors.

I. Radial Sampling

Figure 3 shows the disc pupil function discretely sampled in the radial dimension. It is

$$G(f) = \Delta f \sum_{k=0}^{K-1} \delta(f-k\Delta f) \tag{7}$$

where $\Delta f = f_{BW}/K-1$. The exact Fourier Bessel transform of a disc pupil of radius f_{BW} is

$$B\left\{\text{circ}(f/f_{BW})\right\} = f_{BW} \frac{J_1(2\pi f_{BW}\rho)}{\rho} \tag{8}$$

Substituting (7) into the Fourier Bessel Transform equation, the discrete point spread function is

$$\pi\Delta f \sum_{k=-K+1}^{K-1} |k| J_0\left(\frac{2\pi}{K-1} f_{BW}\rho k\right)$$

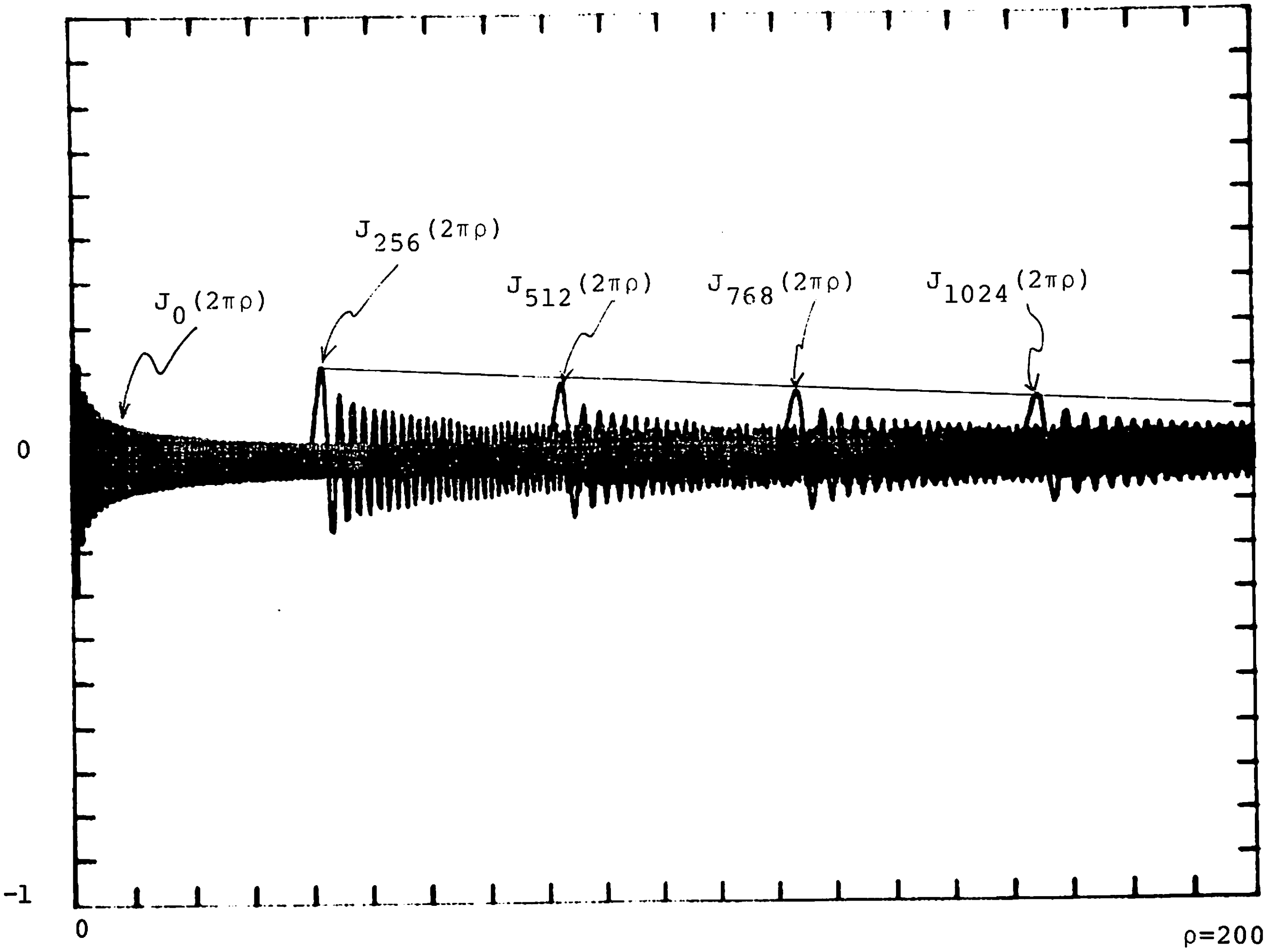
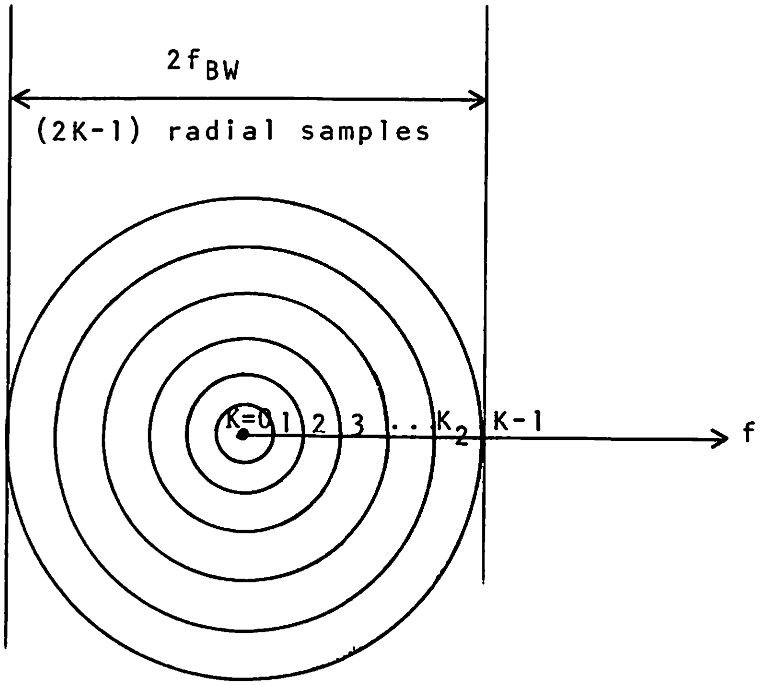
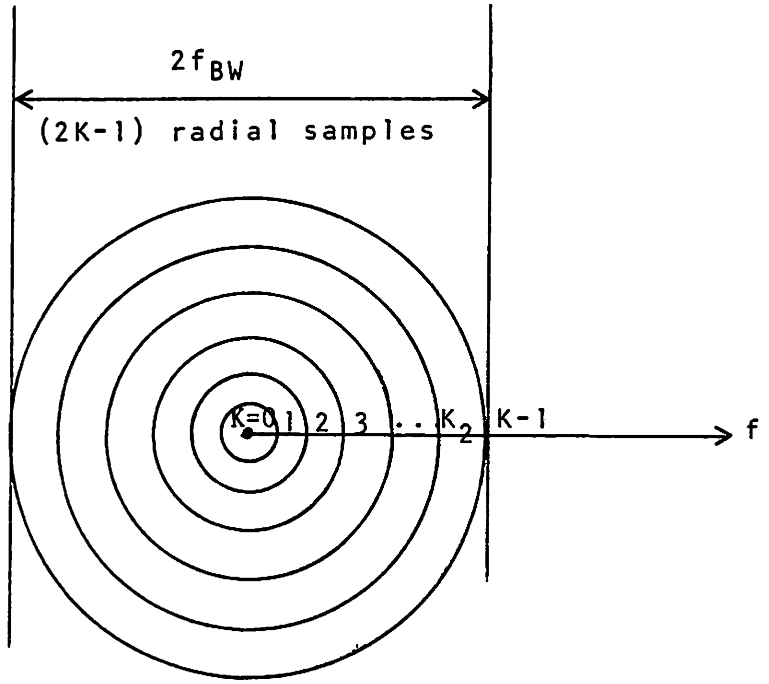


Figure 2. The exact Fourier Bessel Transform $J_0(2\pi\rho)$ and the independent error terms.



Radial Sampling

Figure 3. Full disc pupil function, discrete in radial dimension but continuous in angular dimension.



Radial Sampling

Figure 3. Full disc pupil function, discrete in radial dimension but continuous in angular dimension.

Using Poisson's summation formula (2), and carrying out the simplification, the discrete point spread function is

$$\pi \Delta f \sum_{k=-K+1}^{K-1} |k| J_0\left(\frac{2\pi}{K-1} f_{BW} \rho k\right)$$

$$= f_{BW} \frac{J_1(2\pi f_{BW}\rho)}{\rho} + \sum_{n=1}^{\infty} e_n(\rho) + 2\pi f_{BW} J_0(2\pi f_{BW}\rho) \tag{9}$$

where the second and third terms are the errors due to radial sampling;

$$e_n(\rho) = \hat{e}_n(\rho,x) * w_1(x) \Big|_{x=\frac{n(K-1)}{f_{BW}}} \tag{10}$$

$$\hat{e}_n(\rho,x) = \begin{cases} \frac{1}{2\pi^2}\left[x^2-\rho^2\right]^{-3/2} & x > \rho > 0 \qquad (11) \\ 0 & \rho > x \geqq 0 \qquad (12) \end{cases}$$

and $w_1(x) = 2\pi f_{BW} \mathrm{sinc}(2f_{BW}x)$.

B. <u>Annula Ring Pupil</u>

A similar derivation as in the full disc case will give a point spread function

$$=\left[f_{max}\frac{J_1(2\pi f_{max}\rho)}{\rho} - f_{min}\frac{J_1(2\pi f_{min}\rho)}{\rho}\right] + \sum_{n=1}^{\infty} e_n(\rho) + 2\pi f_{BW}J_0(2\pi f_{max}\rho) \qquad (13)$$

where

$$e_n(\rho) = \hat{e}_n(\rho,x) * w_2(x)\Big|_{x = \frac{n(K-1)}{f_{BW}}} \qquad (14)$$

$\hat{e}_n(\rho,x)$ is as defined in (11)and

$$w_2(x) = 2\pi f_{BW}\text{Sinc}(f_{BW}x)\cos(2\pi f_0 x) \qquad (15)$$

In both cases, $w_1(x)$ and $w_2(x)$ peak at x=0 and $\hat{e}_n(\rho,x)$ blows up at $x=\rho$. Since ρ is a moving parameter in the convolution of (10) and (14), $e_n(\rho)$ will peak in the vicinity of $\rho = n(K-1)/f_{BW}$. The radial sampling error becomes significant therefore at approximately

$$\rho = \frac{n(K-1)}{f_{BW}} \;.$$

Figure 4 shows a plot of the discrete point spread functions for the full disc case and figure 5 shows a plot for the annula ring case.

V. Simultaneous Radial and Angular Sampling

Using Poisson's summation formula (2), and carrying out the simplification, the discrete point spread function is

$$\begin{aligned}&\pi\Delta f \sum_{k=-K+1}^{K-1} |k| J_0\left(\frac{2\pi}{K-1} f_{BW}\rho k\right)\\ &= f_{BW}\frac{J_1(2\pi f_{BW}\rho)}{\rho} + \sum_{n=1}^{\infty} e_n(\rho) + 2\pi f_{BW} J_0(2\pi f_{BW}\rho)\end{aligned} \tag{9}$$

where the second and third terms are the errors due to radial sampling;

$$e_n(\rho) = \hat{e}_n(\rho,x) * w_1(x)\Big|_{x=\frac{n(K-1)}{f_{BW}}} . \tag{10}$$

$$\hat{e}_n(\rho,x) = \begin{cases} \frac{1}{2\pi^2}\left[x^2-\rho^2\right]^{-3/2} & x > \rho > 0 \qquad (11) \\ 0 & \rho > x \geqq 0 \qquad (12) \end{cases}$$

and $w_1(x) = 2\pi f_{BW}\text{sinc}(2f_{BW}x)$.

B. <u>Annula Ring Pupil</u>

A similar derivation as in the full disc case will give a point spread function

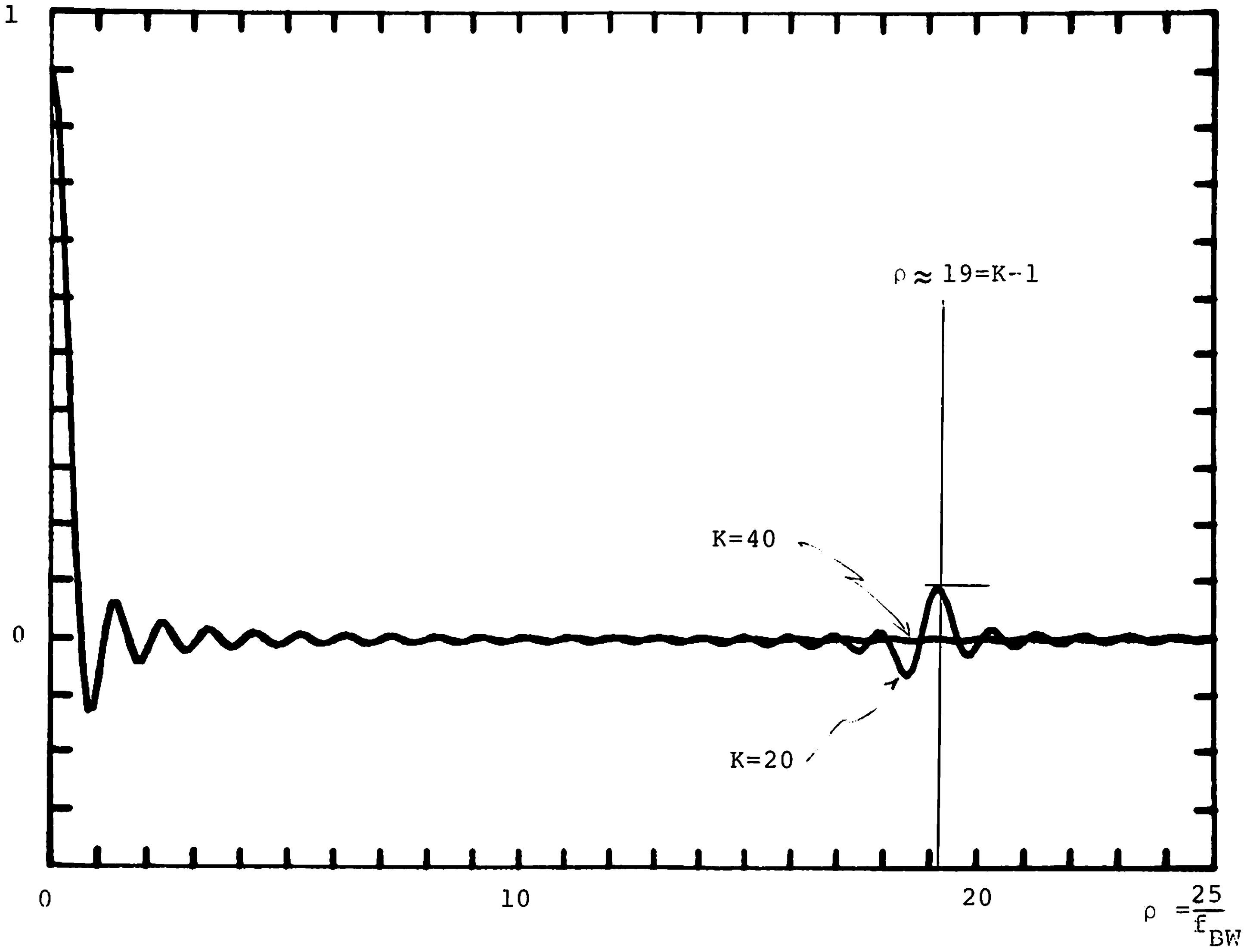


Figure 4. Discrete Fourier Bessel Transform of a radially sampled disc pupil of radius $f_{BW}=1$, with K=20 and K=40. Over this range of ρ, the curve for K=40 is taken at the exact Fourier Bessel Transform.

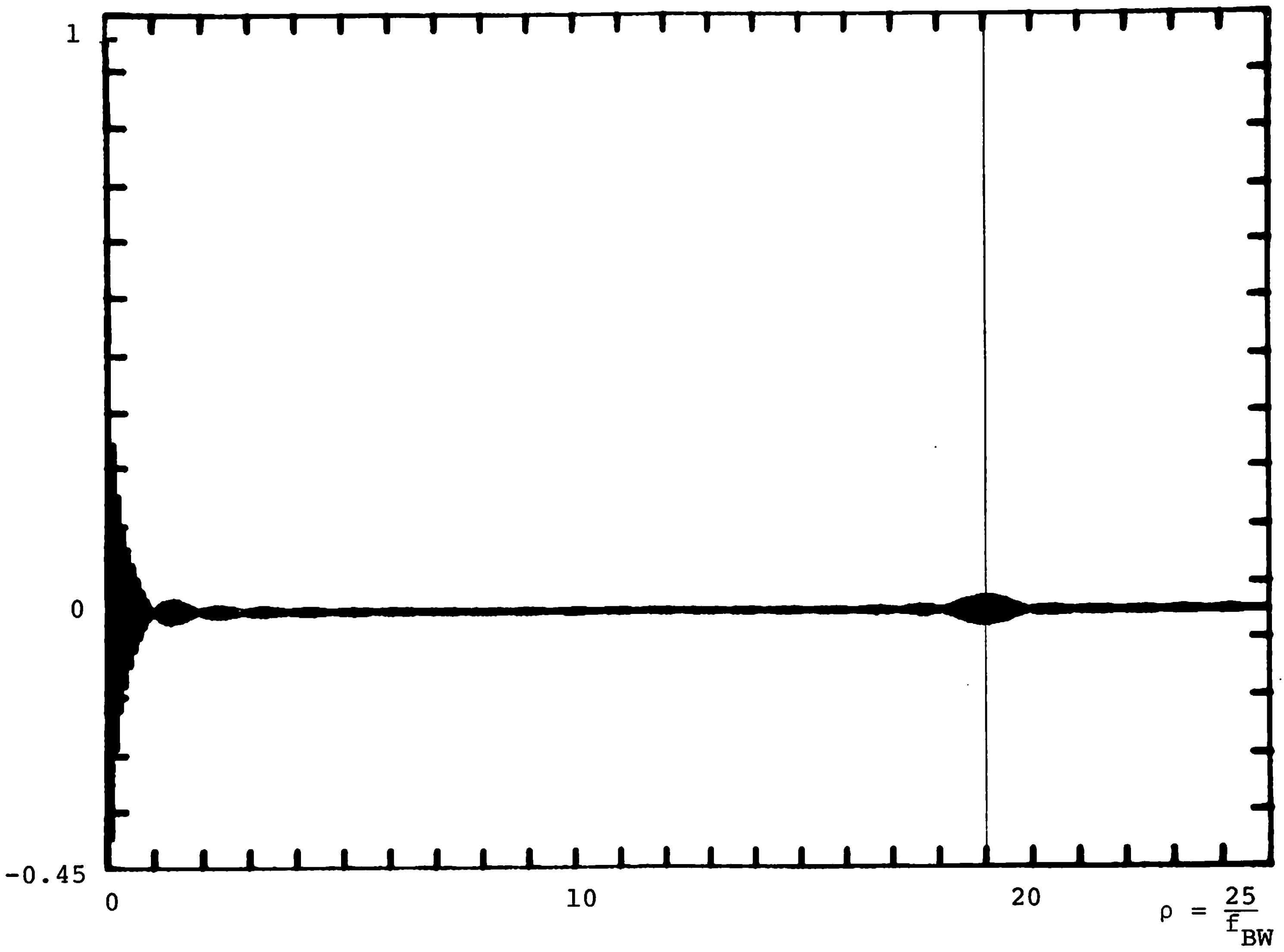


Figure 5. Fourier Bessel Transform of the annula ring with K=20. Note that the 1st order clutter occurs at around $\rho \approx K-1=19$.

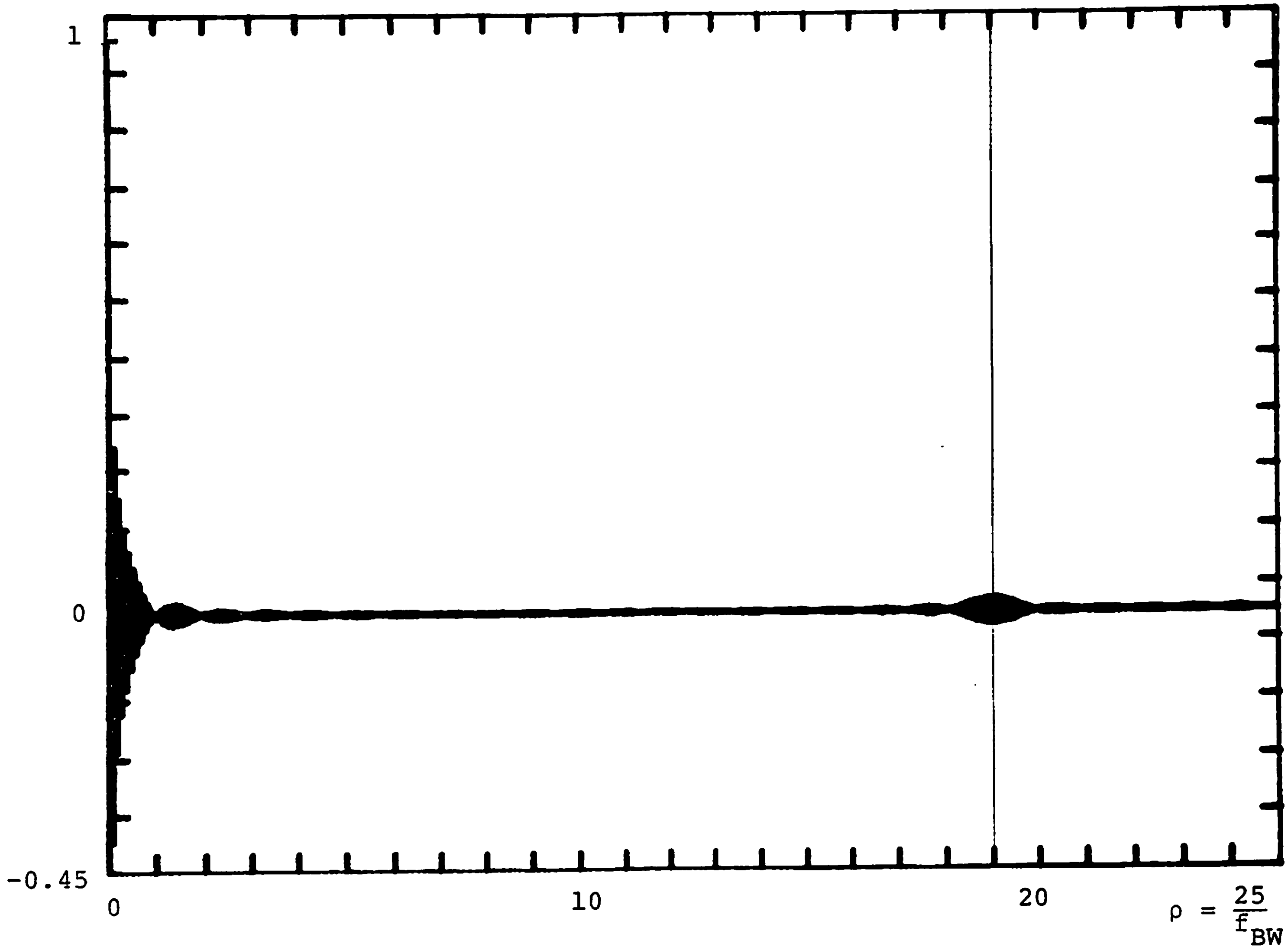


Figure 5. Fourier Bessel Transform of the annula ring with K=20. Note that the lst order clutter occurs at around $\rho \approx$ K-1=19.

Using equation (3) and summing over circles of increasing radius, the discrete point spread function can be obtained by reusing Poisson's summation formula. In general, the discrete point spread function is found to be composed of 5 terms.

1. The exact Fourier transform of the continuous pupil.
2. Radial sampling clutter - the sampling error due to radial sampling alone.
3. Angular sampling clutter - the sampling error due to azimuth sampling alone.
4. Joint sampling clutter - the sampling error due to both radial and azimuth sampling.
5. Residual sampling error - error terms that do not clutter up the point spread function.

Table 2 lists a summary of these five terms for both the disc pupil and the annula ring pupil.

The angular sampling clutter and the radial sampling clutter behave independently of each other and hence N and K can be chosen independently. But in order not to oversample in either azimuth or radial dimension, they can be chosen such that the independent clutters (2 & 3) superimpose, and this occurs when

$$N = 2\pi(K-1) \tag{16a}$$

for disc pupil

$$N = 2\pi(K-1)(f_0 + \tfrac{1}{2}f_{BW})/f_{BW} \tag{16b}$$

for annula ring pupil.

The joint sampling clutter becomes significant always beyond (in radial distance) either the angular sampling clutter or the radial sampling clutter, therefore this component is relatively unimportant.

Table 2. The five components of the Fourier Bessel Transform of the discrete pupil function.

	Disc Pupil	Annula Ring Pupil
Exact Transform	$f_{BW}\dfrac{J_1(2\pi f_{BW}\rho)}{\rho}$	$f_{max}\dfrac{J_1(2\pi f_{max}\rho)}{\rho} - f_{min}\dfrac{J_1(2\pi f_{min}\rho)}{\rho}$
m^{th} order $(m \neq 0)$ Radial Sampling Clutter	$e_m(\rho) = \hat{e}_m(\rho,x) * w_1(x)\big\|_{x = m(K-1)/f_{BW}}$	$e_m(\ ) = \hat{e}_m(\rho,x) \quad w_2(x)\big\|_{x = m(K-1)/f_{BW}}$
$w_i(x)$	$w_1(x) = 2\pi f_{BW}\text{sinc}(2f_{BW}x)$	$w_2(x) = 2\pi f_{BW}\text{sinc}(f_{BW}x)\cos(2\pi f_0 x)$
$\hat{e}_m(\rho,x)$	$\hat{e}_m(\rho,x) = \begin{cases} \dfrac{1}{2\pi^2}\left[x^2-\rho^2\right]^{-3/2} & x > \rho > 0 \\ 0 & \rho > x \geqq 0 \end{cases}$	
$e_m(\rho)$ is significant at	$\rho \approx (K-1)f_{BW}^{-1}$	$\rho \approx (K-1)f_{BW}^{-1}$
n^{th} order $(n \neq 0)$ Angular Sampling Clutter	$A_n(\rho) = (-1)^{nN'}\int_0^{f_{BW}} fJ_{nN}(2\pi\rho f)\,df$	$A_n(\rho) = (-1)^{nN'}\int_{f_{min}}^{f_{max}} fJ_{nN}(2\pi\rho f)\,df$
$A_n(\rho)$ is significant at	$\rho \approx \dfrac{N}{2\pi}\, f_{BW}^{-1}$	$\rho \approx \dfrac{N}{2\pi}\,(f_0 + \tfrac{1}{2}f_{BW})^{-1}$
Joint Sampling Clutter	$c_{nm}(\rho) = \hat{c}_{nm}(\rho,x) * w_1(x)\big\|_{x=m(K-1)/f_{BW}}$	$c_{nm}(\rho) = \hat{c}_{nm}(\rho,x) * w_2(x)\big\|_{x=m(K-1)/f_{BW}}$
$\begin{pmatrix}\text{nth order radial} \\ \text{mth order angular}\end{pmatrix}$ $(n \neq 0,\ m \neq 0)$	$\hat{c}_{nm}(\rho,x) = \begin{cases} \dfrac{4}{(2\pi x)^{nN+1}} \cdot \dfrac{1}{\varepsilon^2[1+\varepsilon]^{nN}} \cdot \left[\dfrac{1}{\varepsilon} + nN\right] & \text{for } x > \rho > 0 \\ 0 & \text{for } \rho > x \geqq 0 \end{cases}$ $\varepsilon = \sqrt{x^2-\rho^2}$	
Residual Sampling error	$\dfrac{\pi}{K-1}\, f_{BW}^2 J_0(2\pi f_{BW}\rho)$	$\dfrac{\pi}{K-1}\left[f_{max}^2 J_0(2\pi f_{max}\rho) - f_{min}^2 J_0(2\pi f_{min}\rho)\right]$

Using equation (3) and summing over circles of increasing radius, the discrete point spread function can be obtained by reusing Poisson's summation formula. In general, the discrete point spread function is found to be composed of 5 terms.

1. The exact Fourier transform of the continuous pupil.
2. Radial sampling clutter - the sampling error due to radial sampling alone.
3. Angular sampling clutter - the sampling error due to azimuth sampling alone.
4. Joint sampling clutter - the sampling error due to both radial and azimuth sampling.
5. Residual sampling error - error terms that do not clutter up the point spread function.

Table 2 lists a summary of these five terms for both the disc pupil and the annula ring pupil.

The angular sampling clutter and the radial sampling clutter behave independently of each other and hence N and K can be chosen independently. But in order not to oversample in either azimuth or radial dimension, they can be chosen such that the independent clutters (2 & 3) superimpose, and this occurs when

$$N = 2\pi(K-1) \tag{16a}$$

for disc pupil

$$N = 2\pi(K-1)(f_0 + \tfrac{1}{2}f_{BW})/f_{BW} \tag{16b}$$

for annula ring pupil.

The joint sampling clutter becomes significant always beyond (in radial distance) either the angular sampling clutter or the radial sampling clutter, therefore this component is relatively unimportant.

The residual sampling error does not clutter the point spread function but only slightly modifies the exact point spread function. For large K, this term is negligibly small.

Now suppose $E_{NK}(\rho)$ is the summation of the angular sampling clutter, radial sampling clutter, and the joint sampling clutter. Then given the maximum clutter level ε that can be tolerated and the maximum radial extent a of the object to be reconstructed from the sampled data, we can choose N and K such that

$$E_{NK}(2a) \leq \varepsilon \tag{17}$$

where $E_{NK}(2a)$ can be computed exactly using the analytic expressions in table 2.

$$E_{NK}(\rho) = e_1(\rho) + A_1(\rho) + c_{11}(\rho)$$

$e_1(\rho)$ = 1st order radial sampling clutter
$A_1(\rho)$ = 1st order angular sampling clutter
$c_{11}(\rho)$ = (1,1) order joint sampling clutter.

VI. Conclusion

We have found exact analytic expressions for the artifacts that are generated by Fourier Bessel transforming a polar coordinate sampled disc and annula ring pupil functions. The resulting point spread functions contained sampling errors which we have called clutters. The clutter terms can in general be put into three categories viz.

1. Radial Sampling clutter
2. Angular sampling clutter
3. Joint sampling clutter.

It turned out that the simultaneous or joint sampling clutter does not become significant before either the radial sampling clutter or the angular sampling clutter does. A special case arises when $N = 2\ (K-1)$ for disc pupil or $N = 2\pi(K-1)\ (f_0 + \frac{1}{2}f_{BW})/f_{BW}$ for annula ring pupil in which case all three components become significant simultaneously and henceforth their sum (joint clutter) must be considered. Then N and K can be chosen exactly by using the analytic expressions for the clutters, given the level of clutter tolerance, and maximum radial extent of the object.

Finally, it is expected that a sampling theorem for polar coordinate sampling can be derived by extending the analysis to a general two dimensional bandlimited object. It should also be noted that if the first and last radial samples were weighted by 0.5 in the (discrete) summations with respect to index k, the residual sampling error term will disappear.

References

1. Course Notes from "Finite Field Transforms," by I.S. Reed, Electrical Engineering Dept., University of Southern California, Los Angeles,Ca.

2. R.M. Messereau and A.V. Oppenheim, "Digital Reconstruction of Multidimensional Signals from their Properties," Proc. IEEE, Vol. 62, No. 10, Oct. 1974.

3. R.N. Bracewell and A.C. Riddle, "Inversion of Fan-Beam Scans in Radio Astronomy," The Astrophys. Journal, Vol. 150, Nov. 1967.

4. P.R. Smith, T.M. Peters, RHT Bates, "Image Reconstruction from Finite Number of Projections," J. Phys. A: Math., Nucl. Gen., Vol.6, March 1973.

5. G.E. Watson, Theory of Bessel Functions,Cambridge University

It turned out that the simultaneous or joint sampling clutter does not become significant before either the radial sampling clutter or the angular sampling clutter does. A special case arises when $N = 2\ (K-1)$ for disc pupil or $N = 2\pi(K-1)\ (f_0 + \frac{1}{2}f_{BW})/f_{BW}$ for annula ring pupil in which case all three components become significant simultaneously and henceforth their sum (joint clutter) must be considered. Then N and K can be chosen exactly by using the analytic expressions for the clutters, given the level of clutter tolerance, and maximum radial extent of the object.

Finally, it is expected that a sampling theorem for polar coordinate sampling can be derived by extending the analysis to a general two dimensional bandlimited object. It should also be noted that if the first and last radial samples were weighted by 0.5 in the (discrete) summations with respect to index k, the residual sampling error term will disappear.

References

1. Course Notes from "Finite Field Transforms," by I.S. Reed, Electrical Engineering Dept., University of Southern California, Los Angeles,Ca.

2. R.M. Messereau and A.V. Oppenheim, "Digital Reconstruction of Multidimensional Signals from their Properties," Proc. IEEE, Vol. 62, No. 10, Oct. 1974.

3. R.N. Bracewell and A.C. Riddle, "Inversion of Fan-Beam Scans in Radio Astronomy," The Astrophys. Journal, Vol. 150, Nov. 1967.

4. P.R. Smith, T.M. Peters, RHT Bates, "Image Reconstruction from Finite Number of Projections," J. Phys. A: Math., Nucl. Gen., Vol.6, March 1973.

5. G.E. Watson, Theory of Bessel Functions,Cambridge University

Press, 1952.

6. A. Abramowitz, L.A. Stegun, Handbook of Mathematical Functions, 1972 Edition.

3.7 Reconstruction of Rotating Targets

Yeh-Hua Peter Chuan

This report is concerned with the reconstruction aspect of the turntable radar imaging system that was described by C.C. Chen and H.C. Andrews in [1]. It was found then that the complex data $D(\theta,\hat{f})$ collected represented the two dimensional Fourier Transform $Z(\theta,\hat{f})$ of the target "reflectivity function" $\sigma(x,y)$ assuming that

1. $\sigma(x,y)$ did not change with aspect angle θ.
2. no shadowing problem existed.

Our reconstruction algorithms will be based on the above assumptions and result. Deviations in practical solid targets from the assumptions will be considered as perturbations which degrade the image. The shadowing problem has been addressed to in [2]. Some further assumptions are that the target rotation rate and the position of the center of rotation are known exactly. Corrections concerning deviation from these assumptions belong to the realm of motion compensation. Also, the collected data is narrowband in nature in that the frequency bandwidth $\hat{f}_{BW}$ is much much less than the mean frequency $\hat{f}_0$, i.e., $\hat{f}_{BW} << \hat{f}_0$.

Reconstruction algorithms based on the above assumptions have been proposed but none seem to make use of the full potential of the system to get the best limiting resolution digitally. The optical implementation of the high resolution reconstruction problem was achieved by J.L. Walker [7] by recording the data on a film in what is called the polar format. Here, we are concerned with digital reconstruction and the problems associated with it. The basic problem of reconstructing rotating targets is that of implementing the discrete version of the inverse transform relationship.

$$\sigma(x,y) = \int_0^{2\pi} \int_{\hat{f}_{min}}^{\hat{f}_{max}} \hat{f} D(\theta,\hat{f}) e^{-j2\pi\hat{f}(x\cos\theta + y\mathrm{Sin}\theta)} d\hat{f} d\theta . \qquad (1)$$

Here we have implicitly assumed that 360° of data are available. In practice, this is not always true and therefore its effect must be considered. In either case, the amount of data involved is usually so horrendously large that it poses a formidable computable and storage problem.

Also, since in practice the data is discretely sampled in polar coordinates, some form of interpolation will be needed somewhere in the reconstruction algorithm; for complex data, this becomes a difficult problem.

Here we will describe three different methods to implement (1). Throughout this report we will use spatial frequency as our basic unit in frequency and shall be denoted by a hat on the letter.

Coherent Processing

The reconstruction algorithm that makes use of the full doppler frequency range available in the data is called coherent processing.

Reconstruction algorithms based on the above assumptions have been proposed but none seem to make use of the full potential of the system to get the best limiting resolution digitally. The optical implementation of the high resolution reconstruction problem was achieved by J.L. Walker [7] by recording the data on a film in what is called the polar format. Here, we are concerned with digital reconstruction and the problems associated with it. The basic problem of reconstructing rotating targets is that of implementing the discrete version of the inverse transform relationship.

$$\sigma(x,y) = \int_0^{2\pi} \int_{\hat{f}_{min}}^{\hat{f}_{max}} \hat{f} D(\theta,\hat{f}) e^{-j2\pi\hat{f}(x\cos\theta+y\mathrm{Sin}\theta)} d\hat{f} d\theta. \qquad (1)$$

Here we have implicitly assumed that 360° of data are available. In practice, this is not always true and therefore its effect must be considered. In either case, the amount of data involved is usually so horrendously large that it poses a formidable computable and storage problem.

Also, since in practice the data is discretely sampled in polar coordinates, some form of interpolation will be needed somewhere in the reconstruction algorithm; for complex data, this becomes a difficult problem.

Here we will describe three different methods to implement (1). Throughout this report we will use spatial frequency as our basic unit in frequency and shall be denoted by a hat on the letter.

Coherent Processing

The reconstruction algorithm that makes use of the full doppler frequency range available in the data is called coherent processing.

Equation (1) can be implemented directly by integrating over $\hat{f}$ first. This step is called Range Compression and the resulting compressed data is given by

$$g_{RC}(\theta,\ell) = \int_{-\frac{1}{2}f_{BW}}^{\frac{1}{2}\hat{f}_{BW}} D(\theta,\hat{f}-\hat{f}_0)e^{j2\pi\hat{f}\ell}d\hat{f} \tag{2}$$

where

$$\hat{f}_{BW} = \hat{f}_{max}-\hat{f}_{min} \qquad \text{and} \qquad \hat{f}_0 = \frac{1}{2}(\hat{f}_{max}+\hat{f}_{min}).$$

By using the narrow assumption on (1), substituting (2) into (1) after a change in variable, we can get the following:

$$\sigma(x,y) = \hat{f}_0\int_0^{2\pi} g_{RC}(\theta,\ell_\theta)e^{-j2\pi\hat{f}_0\ell_\theta}d\theta \tag{3}$$

where

$$\ell_\theta = x\cos\theta+y\mathrm{Sin}\theta.$$

Equation (3) represents the azimuth compression step and the idea of coherent processing is obvious from (3) since the range compressed data are combined after compensating for the phase of the echo from the point (x,y) at the corresponding aspect angle θ.

The physical interpretation of range compression is well known in radar and signal processing. The physical interpretation of azimuth compression can be obtained by noticing that the doppler frequency (with respect to a change in aspect angle) of the echo corresponding to the point (x,y) is given by

$$f_D(\theta) = \frac{d}{d\theta}\Phi(\theta)$$

where

$$\Phi(\theta) = \text{doppler phase} = 2\pi\hat{f}_0\ell_\theta .$$

Hence, the azimuth compression represented in (3) corresponds to a matched filter whose frequency response is given by $f_D(\theta)$.

The point spread function of the imaging system corresponding to this reconstruction technique is the isotropic function

$$\mathrm{PSF}(\rho) = \hat{f}_{max}\frac{J_1(2\pi\rho\hat{f}_{max})}{\rho} - \hat{f}_{min}\frac{J_1(2\pi\rho\hat{f}_{min})}{\rho} \; . \qquad (4)$$

The resolution of this system is approximately 1/f .

Incoherent Processing

In contrast to coherent processing, a reconstruction which does not make use of the doppler phase of the echo is called incoherent processing. For this system, it does not matter how the range compressed data was obtained. The azimuth compression step of this processing technique is achieved by incoherently integrating the magnitude squared range compressed data as follows:

$$\sigma_{incoherent}(x,y) = \int_0^{2\pi} |g_{RC}(\theta,\ell_\theta)|^2 d\theta . \qquad (5)$$

In conventional radar systems, the doppler phase is usually lost at the output end. But if the magnitude of each range profile $|g_{RC}(\theta,\ell_\theta)|$ is recorded for a continuum of aspect angles, equation (5) can be used to reconstruct the target. Equation (5) therefore allows us to do radar imaging with conventional radar systems without the

$$f_D(\theta) = \frac{d}{d\theta}\Phi(\theta)$$

where

$$\Phi(\theta) = \text{doppler phase} = 2\pi\hat{f}_0\ell_\theta.$$

Hence, the azimuth compression represented in (3) corresponds to a matched filter whose frequency response is given by $f_D(\theta)$.

The point spread function of the imaging system corresponding to this reconstruction technique is the isotropic function

$$\mathrm{PSF}(\rho) = \hat{f}_{max}\frac{J_1(2\pi\rho\hat{f}_{max})}{\rho} - \hat{f}_{min}\frac{J_1(2\pi\rho\hat{f}_{min})}{\rho} \; . \tag{4}$$

The resolution of this system is approximately 1/f .

Incoherent Processing

In contrast to coherent processing, a reconstruction which does not make use of the doppler phase of the echo is called incoherent processing. For this system, it does not matter how the range compressed data was obtained. The azimuth compression step of this processing technique is achieved by incoherently integrating the magnitude squared range compressed data as follows:

$$\sigma_{incoherent}(x,y) = \int_0^{2\pi} |g_{RC}(\theta,\ell_\theta)|^2 d\theta . \tag{5}$$

In conventional radar systems, the doppler phase is usually lost at the output end. But if the magnitude of each range profile $|g_{RC}(\theta,\ell_\theta)|$ is recorded for a continuum of aspect angles, equation (5) can be used to reconstruct the target. Equation (5) therefore allows us to do radar imaging with conventional radar systems without the necessity to make costly modifications.

The point spread function of this imaging system can be found to be an isotropic function

$$\mathrm{PSF}(\rho) = \frac{\hat{f}_{BW}}{3}\,\frac{J_1(2\pi\hat{f}_{BW}\rho)}{\rho} + \frac{1}{4}\hat{f}_{BW}\sum_{k=1}^{\infty} c_k \frac{J_{2k+1}(2\pi\hat{f}_{BW}\rho)}{\rho} \qquad (6)$$

where

$$c_k = \frac{2k+1}{(k+\frac{3}{2})(k+\frac{1}{2})(k-\frac{1}{2})}$$

Table 1. Table of coefficients c_k for the Bessel function series in (6).

k	c_k
1	0.400
2	0.095
5	0.044
4	0.026
5	0.017

The point spread function of this system is very close to that of disc pupil function and the summation term in (6) represents deviation of the system PSF from that of a disc pupil.

The resolution of the incoherent system is essentially determined by $1/2\hat{f}_{BW}$ which is much worse than the resolution $1/\hat{f}_0$ theoretically achievable by the coherent processing technique.

Mixed Processing

Another way to implement (1) is to process the data $D(\theta,\hat{f})$ batch by batch. The image from each batch is called a frame image. These frame images can be rotated, interpolated, and superimposed to get a composite image. The batch process can be carried out coherently and the superposition can be done incoherently. Because of this, this processing method is called mixed processing, and was proposed by Chen [1].

Mixed procesing involves the following steps:

1. Segment the data azimuthwise into N segments so that equation (1) becomes

$$\sigma(\xi,\eta) = \sum_{n=0}^{N-1} \iint_{S_n} \hat{f} D(\theta,\hat{f}) e^{j2\pi\hat{f}(\xi\cos\theta+\eta\mathrm{Sin}\theta)} d\theta d\hat{f} \qquad (7)$$

where

$$S_n = \{(\theta,\hat{f})\,|\,\theta\varepsilon[\theta_n - \tfrac{1}{2}\theta_W, \theta_n + \tfrac{1}{2}\theta_W], \quad \hat{f}\varepsilon[\hat{f}_{min},\hat{f}_{max}]\},$$

$\theta_n = \eta\theta_W$ = center azimuth angle of the nth segment,

$\theta_W = 2\pi/N$ = azimuth segment width.

2. Using the narrowband approximation on (7), change of variable, rotation of axes, and Taylor's expansion on the exponential argument of the kernel in (7), we will arrive at the expression:

$$\sigma(\xi,\eta) \cong \sum_{n=0}^{N-1} \tilde{I}_n(\xi,\eta) e^{j2\pi\hat{f}_0[\xi\cos\theta_n+\eta\mathrm{Sin}\theta_n]} \qquad (8)$$

where

necessity to make costly modifications.

The point spread function of this imaging system can be found to be an isotropic function

$$\mathrm{PSF}(\rho) = \frac{\hat{f}_{BW}}{3}\,\frac{J_1(2\pi \hat{f}_{BW}\rho)}{\rho} + \frac{1}{4}\hat{f}_{BW}\sum_{k=1}^{\infty} c_k \frac{J_{2k+1}(2\pi \hat{f}_{BW}\rho)}{\rho} \qquad (6)$$

where

$$c_k = \frac{2k+1}{(k+\frac{3}{2})\,(k+\frac{1}{2})\,(k-\frac{1}{2})}$$

Table 1. Table of coefficients c_k for the Bessel function series in (6).

k	c_k
1	0.400
2	0.095
5	0.044
4	0.026
5	0.017

The point spread function of this system is very close to that of disc pupil function and the summation term in (6) represents deviation of the system PSF from that of a disc pupil.

The resolution of the incoherent system is essentially determined by $1/2\hat{f}_{BW}$ which is much worse than the resolution $1/\hat{f}_0$ theoretically achievable by the coherent processing technique.

Mixed Processing

$$\tilde{I}_n(\xi,\eta) = \text{nth rotated image frame}$$

$$= \iint_{S_n} D_n(\hat{f}_x,\hat{f}_y)\, e^{j2\pi(\hat{f}_x x+\hat{f}_x y)}\, d\hat{f}_x d\hat{f}_y \qquad (9)$$

$$x = \xi\cos\theta_n + \eta\mathrm{Sin}\theta_n$$

$$y = -\xi\mathrm{Sin}\theta_n + \cos\theta_n$$

$$\hat{f}_x = \hat{f} - \hat{f}_0$$

$$\hat{f}_y = (\theta-\theta_n)\hat{f} \quad \text{for } \theta \varepsilon \text{nth segment}$$

$$D_n(\hat{f}_x,\hat{f}_y) = D(\theta,\hat{f}).$$

Equation (8) is an approximation to the coherent processing technique. The batch processing in (9) can be implemented by FFT techniques. The processing suggested by equation (8) represents batch by batch coherent processing. Instead of (8), incoherent batch by batch processing can be done.

$$\sigma_{MP}(\xi,\eta) = \sum_{n=0}^{N-1} |\tilde{I}_n(\xi,\eta)|^2 \qquad (10)$$

The point spread function of each image frame is closely approximated by

$$\theta_W\hat{f}_0\hat{f}_{BW}\mathrm{Sinc}(\hat{f}_{BW}x)\,\mathrm{Sinc}\,(\theta_W\hat{f}_0 y)$$

where x is the down-range dimension (along the line-of-sight) and y is the cross-range dimension. The point spread function for the mixed processing technique is therefore a superposition of the point spread functions of N such image frames. The resulting point spread function is essentially given by

$$\mathrm{PSF}(\rho) = \theta\hat{f}_O\hat{f}_{BW}\mathrm{Sinc}\,(\hat{f}_m\rho)$$

where $\hat{f}_m = \min(\hat{f}_{BW}, \theta_W\hat{f}_0)$.

Computationally, this is the fastest, but the resolution is limited to the maximum azimuth segment width θ_W. θ_W is chosen small enough so that "range walking" and "variable range rate" aberrations are negligible. In particular,

$$\theta_W \leqq \frac{1}{a\hat{f}_{BW}} \qquad \text{for negligible "range walk"}$$

$$\theta_W \leqq \sqrt{\frac{2}{a\hat{f}_0}} \qquad \text{for negligible "variable range-rate"}$$

where a=maximum radial extent of the object (target). For example with

$$\hat{f}_0 = 200a^{-1} \quad , \quad \hat{f}_{BW} = 10a^{-1},$$

$\theta_W \overset{<}{=} 5.73$ for both negligible range walk and variable range-rate aberrations. The resolution of this system is limited to approximately $1/\hat{f}_0\theta_W = \frac{1}{20}a$.

Figure 1 shows the point spread functions of the imaging system using the three processing techniques.

Digital Reconstruction

Coherent Processing

A discrete version of equation (2) must be used for digital processing. Let

$$g_{RC}(i,n) = \frac{1}{K}\sum_{k=1}^{K-1} D(i,k)e^{j\frac{2\pi}{K}nk} \qquad (11)$$

$$\begin{aligned}\tilde{I}_n(\xi,\eta) &= \text{nth rotated image frame}\\ &= \iint_{S_n} D_n(\hat{f}_x,\hat{f}_y)\,e^{j2\pi(\hat{f}_x x+\hat{f}_x y)}\,d\hat{f}_x d\hat{f}_y \qquad (9)\end{aligned}$$

$x = \xi\cos\theta_n + \eta\mathrm{Sin}\theta_n$

$y = -\xi\mathrm{Sin}\theta_n + \cos\theta_n$

$\hat{f}_x = \hat{f} - \hat{f}_0$

$\hat{f}_y = (\theta-\theta_n)\hat{f}$ for $\theta\varepsilon$nth segment

$D_n(\hat{f}_x,\hat{f}_y) = D(\theta,\hat{f})$.

Equation (8) is an approximation to the coherent processing technique. The batch processing in (9) can be implemented by FFT techniques. The processing suggested by equation (8) represents batch by batch coherent processing. Instead of (8), incoherent batch by batch processing can be done.

$$\sigma_{MP}(\xi,\eta) = \sum_{n=0}^{N-1} |\tilde{I}_n(\xi,\eta)|^2 \qquad (10)$$

The point spread function of each image frame is closely approximated by

$$\theta_W \hat{f}_0 \hat{f}_{BW} \mathrm{Sinc}(\hat{f}_{BW} x)\,\mathrm{Sinc}\,(\theta_W \hat{f}_0 y)$$

where x is the down-range dimension (along the line-of-sight) and y is the cross-range dimension. The point spread function for the mixed processing technique is therefore a superposition of the point spread functions of N such image frames. The resulting point spread function is essentially given by

$$\mathrm{PSF}(\rho) = \theta \hat{f}_O \hat{f}_{BW} \mathrm{Sinc}\,(\hat{f}_m \rho)$$

where $\hat{f}_m = \min(\hat{f}_{BW}, \theta_W \hat{f}_0)$.

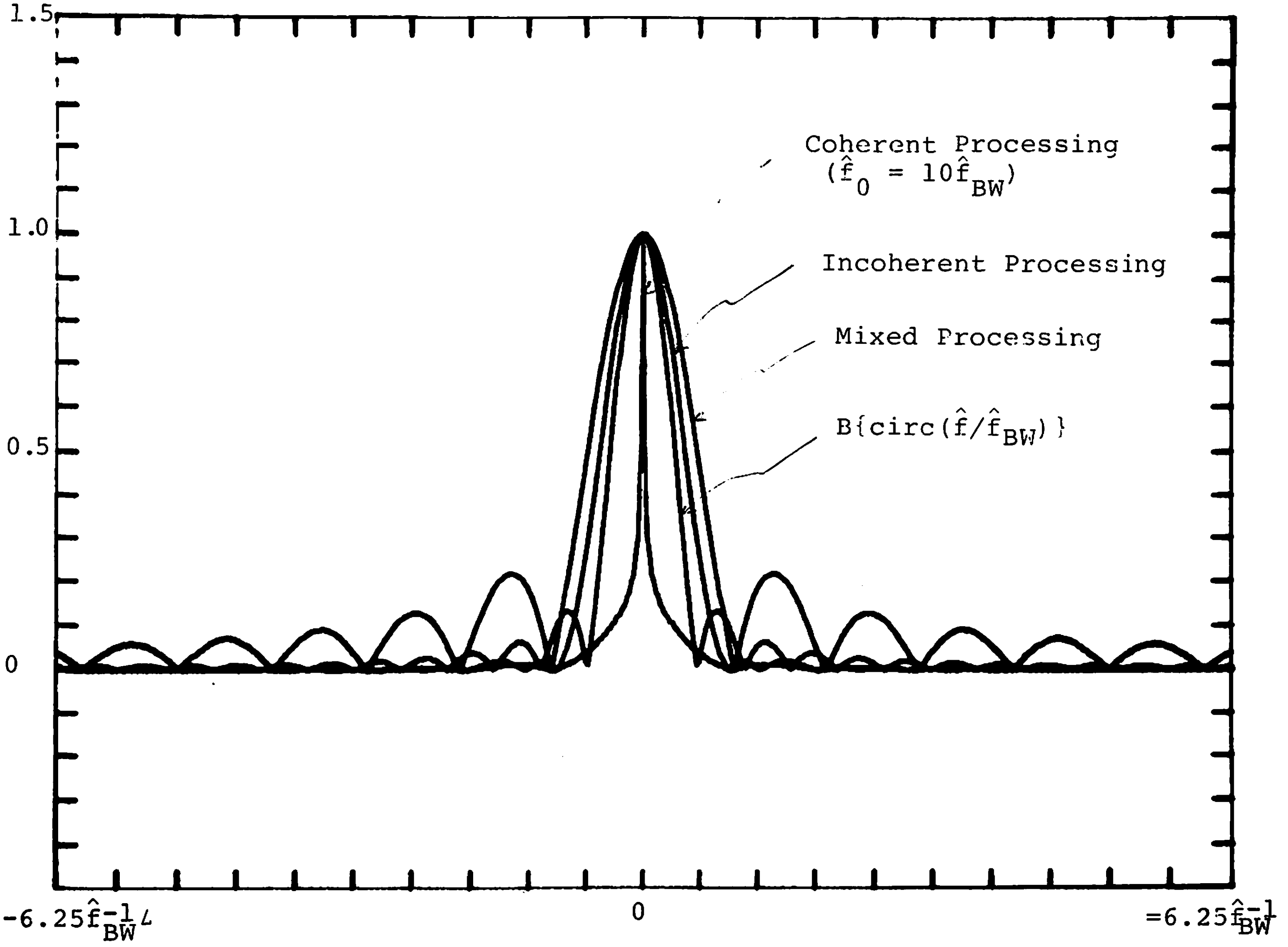


Figure 1. Plots of magnitudes of point spread functions related to the three processing techniques. $B\{circ(\hat{f}/\hat{f}_{BW})\}$ is plotted as a comparison to the incoherent processing case. The point spread functions plotted are:

$f_{max}\frac{J_1(2\pi f_{max}\rho)}{\rho} - f_{min}\frac{J_1(2\pi f_{min}\rho)}{\rho}$	Coherent Processing
$\frac{f_{BW}}{3}\frac{J_1(2\pi f_{BW}\rho)}{\rho} + \frac{f_{BW}}{4}\sum_{k=1}^{\infty} c_k \frac{J_{2k+1}(2\pi f_{BW}\rho)}{\rho}$	Incoherent Processing
$f_{BW}\mathrm{Sinc}(f_{BW}\,\rho)$ (Approximately)	Mixed Processing

where $\quad i = 0,1,2,\ldots,N-1, \quad k,n = 0,1,2,\ldots K-1$

$$g_{RC}(i,n) = g_{RC}\left(\frac{i}{N}2\pi,\ \frac{n}{\hat{f}_{BW}}\right)$$

and

$$D(i,k) = D\left(\frac{i}{N}2\pi,\ \hat{f}_{min} + \frac{k}{K}\hat{f}_{BW}\right).$$

To compute $\sigma(x,y)$, we use a discrete version of (3), namely

$$\sigma(x,y) \cong \frac{\hat{f}_0}{N}\sum_{i=0}^{N-1} g_{RC}(i,[\ell_\theta \hat{f}_{BW}])e^{j2\pi f_0 \ell_\theta} \tag{12}$$

where
$$\ell_\theta = x\cos\left(\frac{i}{N}2\pi\right) + y\sin\left(\frac{i}{N}2\pi\right)$$

$[z]$ = the interger nearest to z.

Expression (12) uses a nearest neighbor interpolation scheme, which is crude compared to those using sinc function weighting factors over several discrete samples of $g_{RC}(i,n)$. But it turns out that this method works quite well. [Note that complex interpolation is not necessary because the blur function of the range compressed data is a real sinc function.] Among the radar community, expression (12) is called doppler phase compensation.

From the section on "Errors in Polar Coordinate Sampling" in this issue, we can determine the azimuth sample rate $(\frac{2\pi}{N})$ and the range sampling rate $(\hat{f}_{BW}/K-1)$. If the azimuth sampling rate is not high enough one must modify (12) to a form such that for each point (x,y) where $$i = 0,1,2,\ldots,N-1, \quad k,n = 0,1,2,\ldots K-1$$

$$g_{RC}(i,n) = g_{RC}\left(\frac{i}{N}2\pi, \frac{n}{\hat{f}_{BW}}\right)$$

and

$$D(i,k) = D\left(\frac{i}{N}2\pi, \hat{f}_{min}+ \frac{k}{K}\hat{f}_{BW}\right).$$

To compute $\sigma(x,y)$, we use a discrete version of (3), namely

$$\sigma(x,y) \cong \frac{\hat{f}_0}{N}\sum_{i=0}^{N-1} g_{RC}(i,[\ell_\theta \hat{f}_{BW}])e^{j2\pi f_0 \ell_\theta} \tag{12}$$

where $$\ell_\theta = x\cos\left(\frac{i}{N}2\pi\right)+y\sin\left(\frac{i}{N}2\pi\right)$$

[z] = the interger nearest to z.

Expression (12) uses a nearest neighbor interpolation scheme, which is crude compared to those using sinc function weighting factors over several discrete samples of $g_{RC}(i,n)$. But it turns out that this method works quite well. [Note that complex interpolation is not necessary because the blur function of the range compressed data is a real sinc function.] Among the radar community, expression (12) is called doppler phase compensation.

From the section on "Errors in Polar Coordinate Sampling" in this issue, we can determine the azimuth sample rate $(\frac{2\pi}{N})$ and the range sampling rate $(\hat{f}_{BW}/K-1)$. If the azimuth sampling rate is not high enough one must modify (12) to a form such that for each point (x,y) being reconstructed, only those aspect angles that lie within θ_C of Ø will be coherently integrated. Ø = arctan $(\pm y/x)$; θ_C is the azimuth coherence interval given by

$$\theta_c \leq \sin^{-1}\left(\frac{1}{2\Delta\theta a\hat{f}_0}\right) \tag{13}$$

where

$\Delta\theta$ = azimuth sampling interval.

Hence

$$\sigma(x,y) \cong \frac{\hat{f}_0}{N}\sum_{i=0}^{N-1}\left\{g_{RC}\left(i,[\ell_\theta \hat{f}_{BW}]\right)e^{j2\pi\hat{f}_0\ell_\theta}\right\}\mathrm{rect}\left(\frac{\frac{2\pi i}{N}-\phi}{2\theta_c}\right) \tag{14}$$

If the range sampling rate is not high enough, we can also use (14) but with

$$\theta_c = \cos^{-1}\left(\frac{K}{2a\hat{f}_{BW}}\right).$$

Incoherent Processing

Since $|g_{RC}(\theta,\ell_\theta)|^2$ represents approximately the real valued projection of the target reflectivity function $\sigma(x,y)$, we can use tomographic schemes to implement equation (5). $\sigma_{incoherent}(x,y)$ is actually called the layergram and does not replicate (x,y) very well. Tomographic schemes that process in the spatial domain (e.g. convolution-back-projection [3],[4]) or in the Fourier domain ([3], [5]) can be used but with computational speed as the criterion, the former scheme may be preferred. For a review on tomographic reconstruction, see [6], [8].

Mixed Processing

Two dimensional FFT techniques can be applied to each azimuth segment. Since only the magnitude of the resulting image frames are used, standard interpolation schemes can be used to rotate the image frames.

Experimental Results

Table 2 shows the parameters of the data for a model F102A plane. Two sets of data were used.

Set 1. The F102A model plane is sitting right side up on the supporting pillars above the turntable. The data was collected over 180° of azimuth angle only.

Set 2. The same plane is sitting vertically (roll angle=90°) and data was collected over 360° azimuth angle.

In both sets of data, the nominal azimuth sampling interval was 0.2°. The maximum radial extent of the object was taken to be approximately 10 ft. This corresponds to a coherence angle interval of $\theta_C = 47^\circ$. Hence, the data is undersampled. For set 2, the sampling interval varies from 0.1° to 0.3°. This corresponds to a coherence interval of 29.4° which is even worse. Sampling in range was oversampled.

Figures 2(a), 3(a) are the coherent reconstructions of the model plane from data sets 1 and 2 respectively, using the modified reconstruction algorithm given by equation (14).

For incoherent processing a much lower sampling rate is needed. From the section "Errors in Polar Coordinate Sampling" in this report, the sampling interval necessary is $\Delta\theta \leqq 1/a\ \hat{f}_{BW} = 2.7^\circ$. Figures 2(b), reconstruction, see [6], [8].

Mixed Processing

Two dimensional FFT techniques can be applied to each azimuth segment. Since only the magnitude of the resulting image frames are used, standard interpolation schemes can be used to rotate the image frames.

Experimental Results

Table 2 shows the parameters of the data for a model F102A plane. Two sets of data were used.

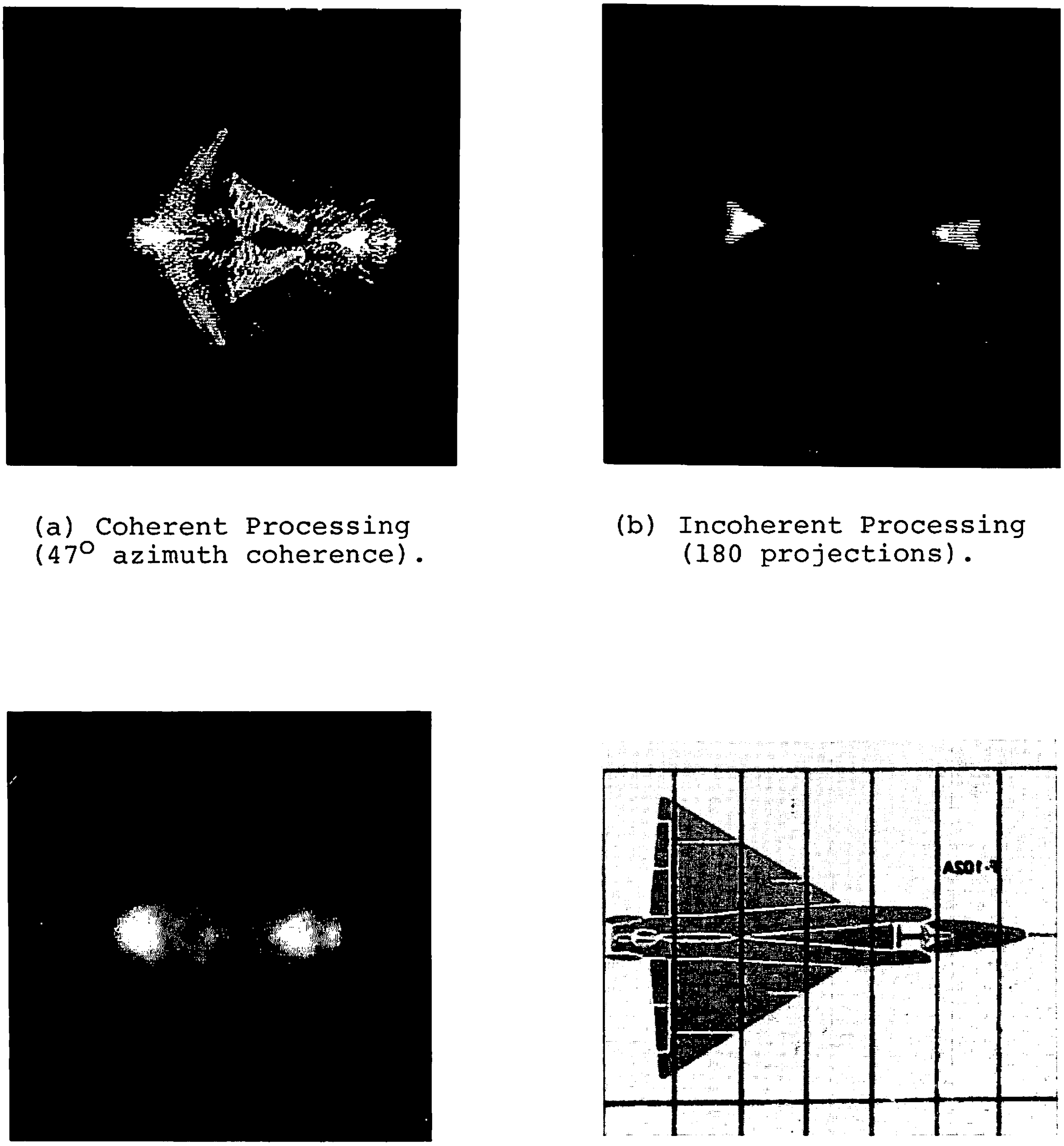
Set 1. The F102A model plane is sitting right side up on the supporting pillars above the turntable. The data was collected over 180° of azimuth angle only.

Set 2. The same plane is sitting vertically (roll angle=90°) and data was collected over 360° azimuth angle.

In both sets of data, the nominal azimuth sampling interval was 0.2°. The maximum radial extent of the object was taken to be approximately 10 ft. This corresponds to a coherence angle interval of $\theta_C = 47^{\circ}$. Hence, the data is undersampled. For set 2, the sampling interval varies from 0.1° to 0.3°. This corresponds to a coherence interval of 29.4° which is even worse. Sampling in range was oversampled.

Figures 2(a), 3(a) are the coherent reconstructions of the model plane from data sets 1 and 2 respectively, using the modified reconstruction algorithm given by equation (14).

For incoherent processing a much lower sampling rate is needed. From the section "Errors in Polar Coordinate Sampling" in this report, the sampling interval necessary is $\Delta\theta \leqq 1/a\ \hat{f}_{BW}=2.7^{\circ}$. Figures 2(b),



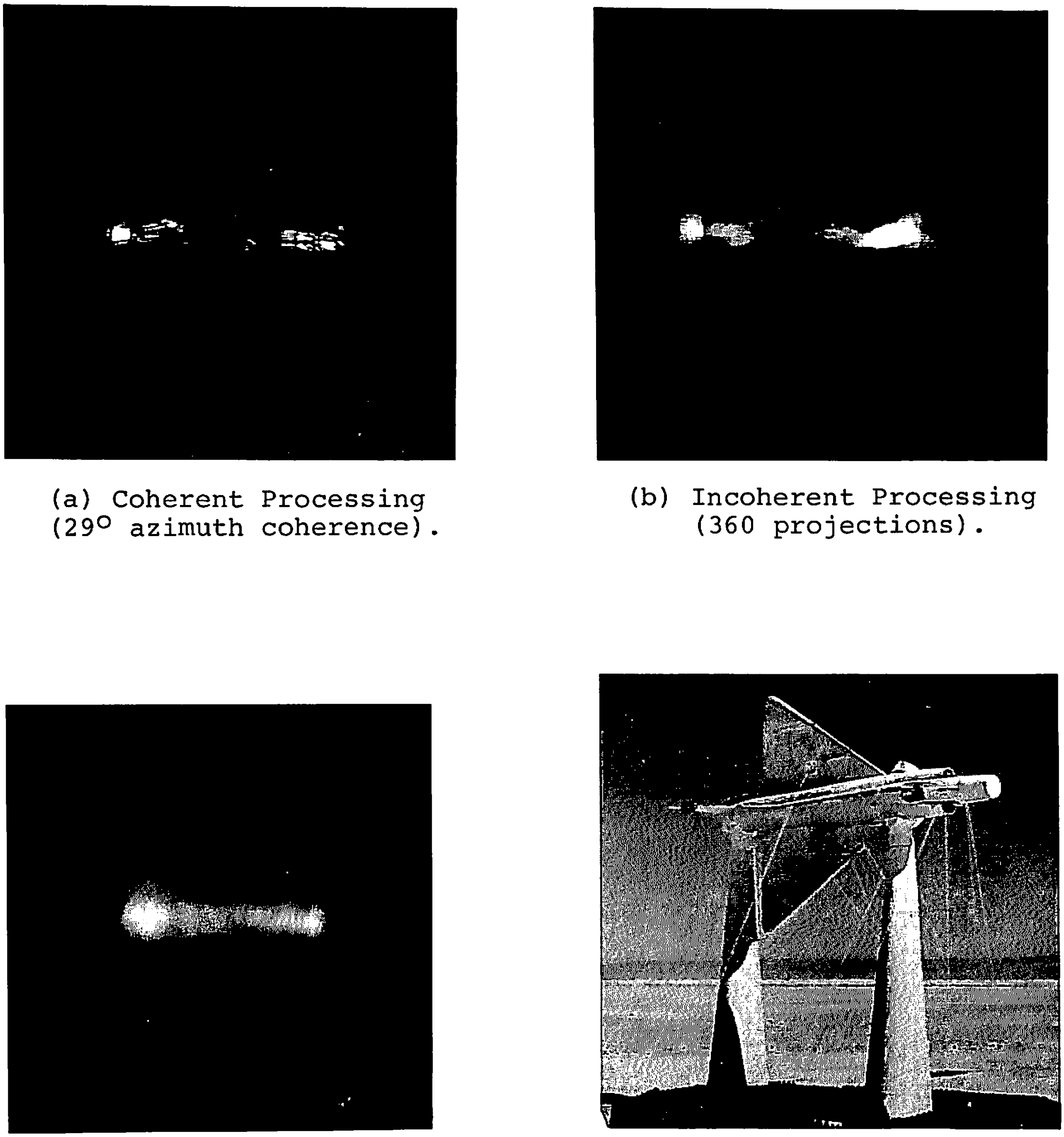
(a) Coherent Processing
(47° azimuth coherence).

(b) Incoherent Processing
(180 projections).

(c) Mixed Processing
(6.4° azimuth segments).

(d) A sketch diagram of
F102A.

Figure 2. Reconstructions from data set 1 using the three reconstruction techniques. A sketch of the original plane is also shown in (d).



(a) Coherent Processing (29° azimuth coherence).

(b) Incoherent Processing (360 projections).

(c) Mixed Processing (6.4° azimuth segments).

(d) Actual model F102A on a turntable.

Figure 3. Reconstructions from data set 2 using the three reconstruction techniques. Data set 2 was collected with the model F102A placed vertically on a turntable as shown in (d).

3(b) show the incoherent reconstructions from data sets 1 and 2 respectively with sampling interval of 1^o and using the tomographic reconstruction scheme of Shepp and Logan [5].

Table 2. Parameters for Model F102A-Plane Data

Model: 0.29 x actual plane dimensions

Plane dimension: 68 ft. (Nose-Tail), 38 ft. (Wing-Span)

Model dimension: 20 ft. (Nose-Tail), 11 ft. (Wing-Span)

Actual physical data	Normalized data (against a=10ft.)
f_{min} = 9.130 GHz	$\hat{f}_{min}$ = 185.947 (a^{-1})
f_{max} = 9.997 GHz	$\hat{f}_{max}$ = 203.36 (a^{-1})
f_0 = 9.5637 GHz	$\hat{f}_0$ = 194.478 (a^{-1})
f_{BW} = 0.83 GHz	$\hat{f}_{BW}$ = 17.719 (a^{-1})
Δf = 3.4 MHz	$\Delta\hat{f}$ = 6.92 x 10 (a^{-1})
$\Delta\theta$ = 0.2^o	$\Delta\theta$ = 0.00349 radians
K = 256	K = 256
N = 1800	N = 1800

Conversion factor

$$\underset{(sec^{-1})}{f_{temporal}} \xrightarrow{\times \frac{2}{c}} \underset{(ft^{-1})}{\hat{f}_{spatial}}$$

Figures 2(c) and 3(c) show composite images of the model plane from data sets 1 and 2 respectively, using mixed processing technique.

Each image frame is obtained by taking the DFT of a 6.4^{0} segment of the data. Figures 2(d) and 3(d) show the actual model plane on the turntable.

The results show that the coherent processing technique does improve the resolution of the image to a significant extent as it was predicted by the point spread functions. The blunt nose of the model plane, the air intake, the cockpit position, the vertical diameter of the fuselage, the wing angle and the vertical tail length can all be distinctly identified. Because coherent integration was carried out over aspect angles that are $\pm 29^{0}$ from a target point angle $\emptyset$, a target point will be reconstructed only if it can be "seen" (by the radar) within $\pm 29^{0}$ from the line of sight. Therefore, the front slope of the vertical tail fin is not obvious in figure 3(a).

The incoherent processing technique also shows a significant amount of "intelligibility" for data set 1 but not so for data set 2. The mixed processing technique gives about the same resolution as the incoherent processing technique, but the experimental results do not show conclusively that one is "better" than the other.

Conclusion

We have proposed two processing techniques called coherent processing and incoherent processing. The coherent processing technique is computationally much more demanding than the other two techniques but it does show a remarkable improvement in resolution and "intelligibility." The incoherent processing requires the least amount of computational effort and storage memory but it is restricted by the necessity of having 360^{0} of the data being available. A more concrete comparative study is needed on the sensitivity to the estimation error on the center of rotation, the error on the rotation rate, doppler phase errors and so on.

Each image frame is obtained by taking the DFT of a 6.4° segment of the data. Figures 2(d) and 3(d) show the actual model plane on the turntable.

The results show that the coherent processing technique does improve the resolution of the image to a significant extent as it was predicted by the point spread functions. The blunt nose of the model plane, the air intake, the cockpit position, the vertical diameter of the fuselage, the wing angle and the vertical tail length can all be distinctly identified. Because coherent integration was carried out over aspect angles that are $\pm 29^{\circ}$ from a target point angle $\emptyset$, a target point will be reconstructed only if it can be "seen" (by the radar) within $\pm 29^{\circ}$ from the line of sight. Therefore, the front slope of the vertical tail fin is not obvious in figure 3(a).

The incoherent processing technique also shows a significant amount of "intelligibility" for data set 1 but not so for data set 2. The mixed processing technique gives about the same resolution as the incoherent processing technique, but the experimental results do not show conclusively that one is "better" than the other.

Conclusion

We have proposed two processing techniques called coherent processing and incoherent processing. The coherent processing technique is computationally much more demanding than the other two techniques but it does show a remarkable improvement in resolution and "intelligibility." The incoherent processing requires the least amount of computational effort and storage memory but it is restricted by the necessity of having 360° of the data being available. A more concrete comparative study is needed on the sensitivity to the estimation error on the center of rotation, the error on the rotation rate, doppler phase errors and so on.

Another major significance our approach to the reconstruction problem is the complete negligence of the temporal (which is practical) aspect of the system. By using spatial units (e.g. spatial frequency, spatial distance) and angular units, understanding of the system becomes considerably simplified and insight is generated. Even then, intuition into the practical design parameters (e.g. time sampling requirement, pulse repetition period, etc.) are not lost because of the simplicity of the conversion factors from spatial and angular units to temporal units.

References

1. C.C. Chen and H.C. Andrews,"Turntable Radar Imaging," USCIPI Report 800, March 31, 1978.

2. Y.H. Chuan,"Splotlight S.A.R. Imaging Using Rat-Scat Data," USCIPI Report 840, Sept. 30, 1978.

3. R.N. Bracewell and A.C. Riddle, "Inversion of Fan-Beam Scans in Radio Astronomy," Astrophys. J. Vol. 150, Nov. 1967.

4. G.N. Ramachandran and A.V. Lakshminarayanan, "Three Dimensional Reconstruction from Radiographs and Electron Micrographs: Application of Convolution Instead of Fourier Transforms," Proc. Nat. Acad. Sci., Vol. 68, 1971.

5. L.A. Shepp and B.F. Logan, "The Fourier Reconstruction of a Head Section," IEEE Trans. on Nuclear Science, Vol. NS-21, June 1974.

6. B.K.P. Horn, "Density Reconstruction Using Arbitrary Ray-Sampling Schemes," IEEE Proc. Vol. 66, No. 5, May 1978.

7. J.L. Walker, "Range Doppler Imaging of Rotating Objects," Dissertation at U. of Michigan, 1974.

8. H.J. Scudder, "Introduction to Computer Aided Tomography," IEEE Proc., Vol. 66, No. 6, June 1978.

4. Smart Sensor Projects

4.1 Implementation of Advanced Real-Time Image Understanding Algorithms

G.R. Nudd, P.A. Nygaard*, S.D. Fouse and T.A. Nussmeier

Introduction

During the past period, we have continued our work to develop custom designed integrated circuits for real-time implementation of image understanding algorithms. The work has centered on three areas; the detailed design and layout of a third test chip, TCIII, the development of new concepts for more advanced (higher level) processing operations (including a texture chip) [1], and the design and construction of the necessary circuits such as clock drivers to operate the processors.

In the previous phase of this program, we developed concepts and test circuits for "real-time" (equivalent to television data rates) processing of "low-level" or preprocessing algorithms, including edge detection, unsharp-masking, local averaging, adaptive stretch and binarization. Most of these circuits (apart from the binarizers) performed the calculations as analog operations. We were able to demonstrate an accuracy, or intensity resolution, equivalent to approximately 6 bits using these techniques. (During this period of the contract, we reported this work at the Image Understanding Systems and Industrial Applications session of the annual meeting of the Society of Photo-Optical Instrumentation Engineers in San Diego, in August 1978, and at the 1978 International Charge Coupled Devices Symposium, San Diego, October 1978. These papers are included as appendices.) The rationale for this approach has been that the modern sensors, in many cases, are themselves analog charge coupled devices (CCDs), and thus by structuring the preprocessing to be in the form of two-dimensional CCD transversal filters, we can simultaneously obtain

*Mr. Nygaard is with the Hughes Carlsbad Research Center.

a number of parallel real-time processed outputs directly on the imaging chip itself, as shown in Figure 1.

We have shown that less than 0.1% extra silicon area and circuit complexity is required to provide these processing operations. Also, since the sensor data itself consists of analog charge packages representing the picture intensities, and each of the processing operations is performed directly in the charge domain [2], we can maintain the optimum accuracy, linearity and dynamic range, avoiding the errors and nonlinearities inherent in the charge to voltage transducers, etc. The processing techniques rely on a non-destructive sensing approach, using floating gate arrays [3], hence the original picture elements are also always available. For example, approximately 20 additional CCD stages are required to perform the five operations described above, which would represent an additional signal degradation of approximately 0.2% (assuming a charge transfer efficiency of 0.9999 for a high quality images). This approach is particularly appropriate for operations such as image enhancement, feature extraction and data compression, etc., where a number of operations have to be performed directly on each of the pixels in the sensed image. At this stage in the processing, an accuracy of six bits is probably sufficient for almost all applications, but the number of operations is very large, typically several hundred million per second. Analog processing is ideal for such tasks, and the essentially pipelined operation of the 2-D transversal filters can comfortably handle the computation rates. However, further down the processing chain, the data rates are typically reduced, while the required accuracy and dynamic range increases. Hence our approach is to employ fast analog preprocessors integrated at or close to the sensor itself and then follow this by custom built programmable digital processing using highly regular LSI or VLSI designs. For example, a full processor might appear as shown in Figure 2.

We report on three separate tasks in this report. Section II describes our continuing work on Test Chip III, to develop high speed a number of parallel real-time processed outputs directly on the imaging chip itself, as shown in Figure 1.

We have shown that less than 0.1% extra silicon area and circuit complexity is required to provide these processing operations. Also, since the sensor data itself consists of analog charge packages representing the picture intensities, and each of the processing operations is performed directly in the charge domain [2], we can maintain the optimum accuracy, linearity and dynamic range, avoiding the errors and nonlinearities inherent in the charge to voltage transducers, etc. The processing techniques rely on a non-destructive sensing approach, using floating gate arrays [3], hence the original picture elements are also always available. For example, approximately 20 additional CCD stages are required to perform the five operations described above, which would represent an additional signal degradation of approximately 0.2% (assuming a charge transfer efficiency of 0.9999 for a high quality images). This approach is particularly appropriate for operations such as image enhancement, feature extraction and data compression, etc., where a number of operations have to be performed directly on each of the pixels in the sensed image. At this stage in the processing, an accuracy of six bits is probably sufficient for almost all applications, but the number of operations is very large, typically several hundred million per second. Analog processing is ideal for such tasks, and the essentially pipelined operation of the 2-D transversal filters can comfortably handle the computation rates. However, further down the processing chain, the data rates are typically reduced, while the required accuracy and dynamic range increases. Hence our approach is to employ fast analog preprocessors integrated at or close to the sensor itself and then follow this by custom built programmable digital processing using highly regular LSI or VLSI designs. For example, a full processor might appear as shown in Figure 2.

We report on three separate tasks in this report. Section II describes our continuing work on Test Chip III, to develop high speed

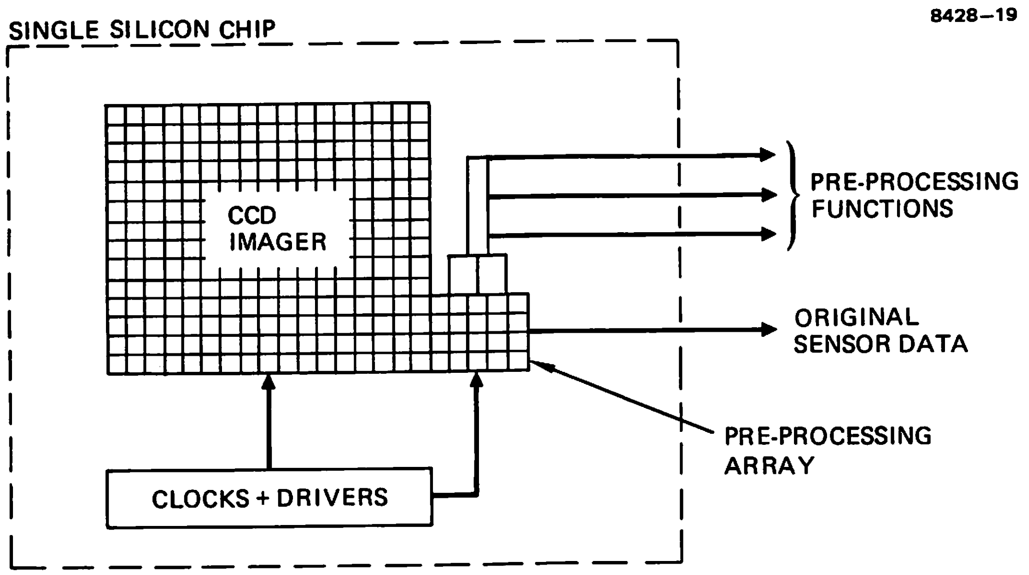


Figure 1. Schematic of integrated analog pre-processing and imaging chip.

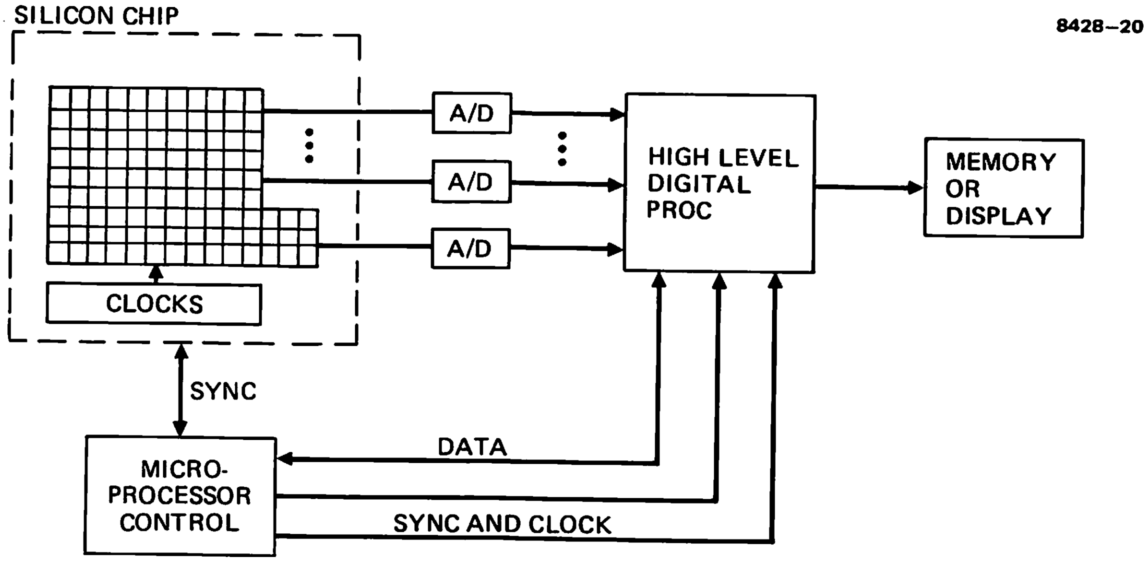


Figure 2. Schematic of image understanding system.

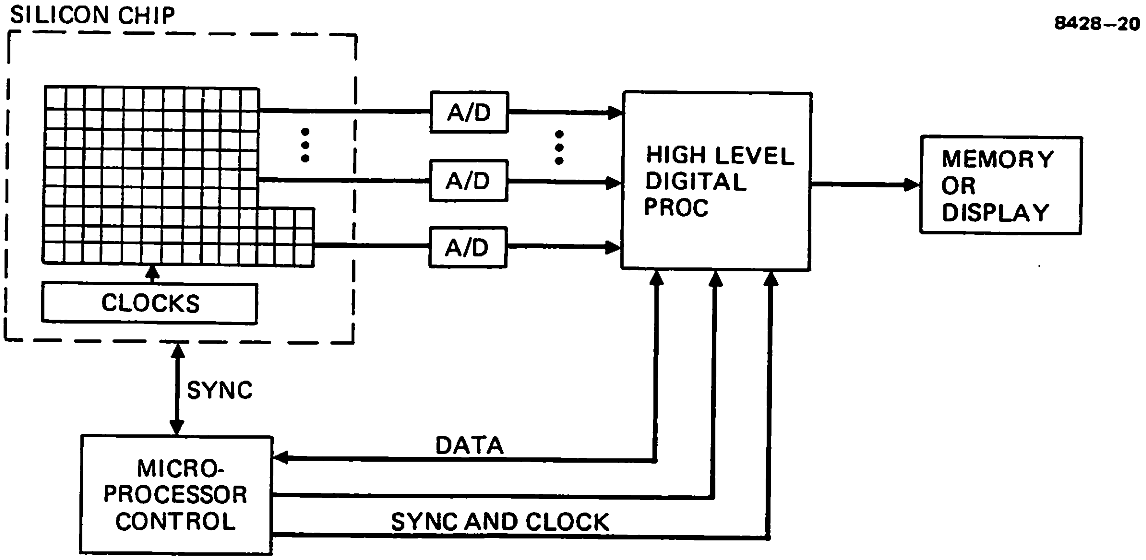


Figure 2. Schematic of image understanding system.

preprocessing functions. Section III discusses the work we have undertaken on our image processing facility to enable us to operate our custom built integrated circuits in real-time, and in Section IV, we describe an approach to higher level processing such as texture and segmentation.

II. Design and Fabrication of Test Chip III

We have investigated five circuits for inclusion in our third test chip. These include a 3x3 Laplacian operator, a 7x7 kernel, which we are currently implementing as an edge detector but can be mask programmed to perform other operations such as the binary checker-boards or unsharp masking, a 5x5 programmable filter, which we intend to integrate with a commercial microprocessor, a 5x5 'cross-shaped' median and a large bipolar convolutional array for 26x26 pixel convolutions.

For each of these, we have developed circuit concepts which will enable the data to be processed at real-time data rates. Circuit simulations which evaluate the accuracy, speed and hence dynamic-range have been completed for each circuit. The detailed designs and layouts of most of these operators have now been completed and we estimate that the drawings will be sent to the mask maker by mid-March. If we are able to obtain prompt delivery from the mask-makers (probably Micro-Fab, Inc.), we anticipate having processed parts by July, which should allow their evaluation to be undertaken prior to the end of this phase of the contract in September. The program schedule is given in Figure 3.

The technology used to implement the algorithms is n-channel, two-phase metal-oxide-semiconductor (MOS) and CCD. The full chip size is approximately 225 milsx225 mils and conventional photolithography is employed, resulting in a line width of approximately 5 μm. With this resolution and by using surface channel technology, a clock rate 7.5 MHz is possible. A block schematic and a brief description of

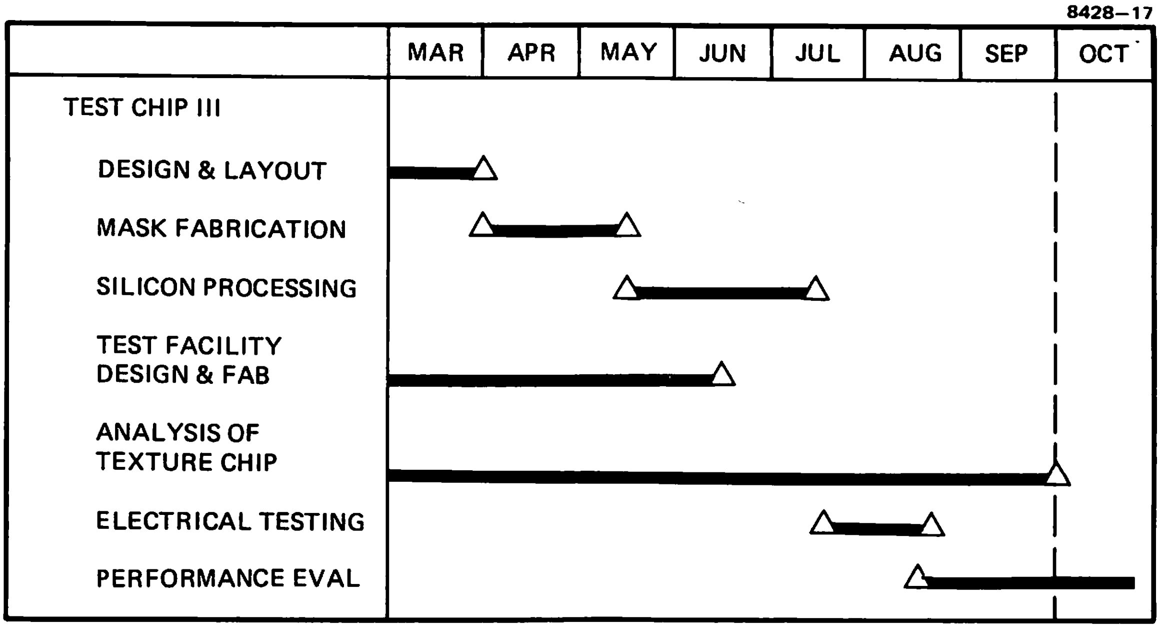


Figure 3. Schedule for test chip III.

preprocessing functions. Section III discusses the work we have undertaken on our image processing facility to enable us to operate our custom built integrated circuits in real-time, and in Section IV, we describe an approach to higher level processing such as texture and segmentation.

II. Design and Fabrication of Test Chip III

We have investigated five circuits for inclusion in our third test chip. These include a 3x3 Laplacian operator, a 7x7 kernel, which we are currently implementing as an edge detector but can be mask programmed to perform other operations such as the binary checker-boards or unsharp masking, a 5x5 programmable filter, which we intend to integrate with a commercial microprocessor, a 5x5 'cross-shaped' median and a large bipolar convolutional array for 26x26 pixel convolutions.

For each of these, we have developed circuit concepts which will enable the data to be processed at real-time data rates. Circuit simulations which evaluate the accuracy, speed and hence dynamic-range have been completed for each circuit. The detailed designs and layouts of most of these operators have now been completed and we estimate that the drawings will be sent to the mask maker by mid-March. If we are able to obtain prompt delivery from the mask-makers (probably Micro-Fab, Inc.), we anticipate having processed parts by July, which should allow their evaluation to be undertaken prior to the end of this phase of the contract in September. The program schedule is given in Figure 3.

The technology used to implement the algorithms is n-channel, two-phase metal-oxide-semiconductor (MOS) and CCD. The full chip size is approximately 225 milsx225 mils and conventional photolithography is employed, resulting in a line width of approximately 5 μm. With this resolution and by using surface channel technology, a clock rate 7.5 MHz is possible. A block schematic and a brief description of each circuit is given below.

II.A. Laplacian Operator

The Laplacian operator [4] is a bipolar weighting scheme $\underline{A}$ given in equation (1) operating on a 3x3 array of picture elements, which produces a convolution output $\underline{A} * \underline{p}$ where $\underline{p}$ is the unprocessed image array.

$$\underline{A} = \begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix} \tag{1}$$

It is used for crispening and edge sharpening. It can be implemented directly at the sensor using a two-dimensional CCD array consisting of a set of linear transversal filters. A schematic of the system is shown in Figure 4. Each filter is a two-phase n-channel device with eighteen gates. The added latency time for this device is equivalent to approximately 0.5 pixels ($\simeq$ 0.1 μsec). This is in addition to the inherent delay of the algorithm of approximately one video line ($\simeq$ 63 μsec). The input to each CCD structure consists of a Tompsett [5] structure as shown in Figure 5. We have used this input technique on each of the CCD structures to provide the optimum linearity at the voltage to charge conversion.

The circuit uses the floating gate technique to sense nine adjacent charge packets representing the array of 3x3 adjacent pixels. As the three adjacent lines of charge, representing the video data, are clocked through the array, the weighted sum of the charge or pixel magnitudes, at each clock sample, is applied to an 'on-chip' sample and hold. For example the voltage on the floating gate array, sensed by the sample and held at the nth clock period, T, is

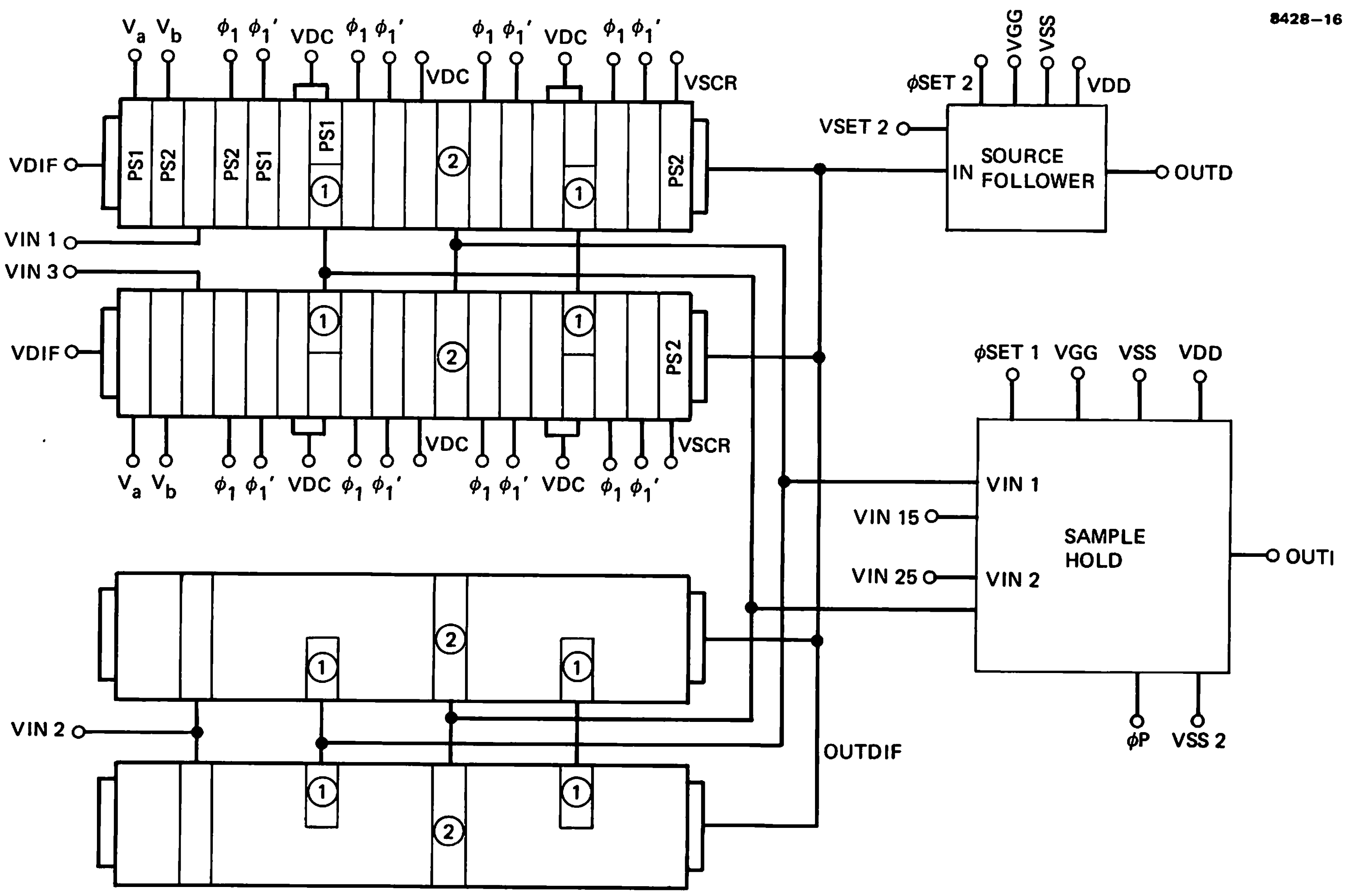


Figure 4. Schematic of CCD Laplacian operator.

each circuit is given below.

II.A. Laplacian Operator

The Laplacian operator [4] is a bipolar weighting scheme $\underline{A}$ given in equation (1) operating on a 3x3 array of picture elements, which produces a convolution output $\underline{A} * \underline{p}$ where $\underline{p}$ is the unprocessed image array.

$$\underline{A} = \begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix} \qquad (1)$$

It is used for crispening and edge sharpening. It can be implemented directly at the sensor using a two-dimensional CCD array consisting of a set of linear transversal filters. A schematic of the system is shown in Figure 4. Each filter is a two-phase n-channel device with eighteen gates. The added latency time for this device is equivalent to approximately 0.5 pixels ($\simeq$ 0.1 μsec). This is in addition to the inherent delay of the algorithm of approximately one video line ($\simeq$ 63 μsec). The input to each CCD structure consists of a Tompsett [5] structure as shown in Figure 5. We have used this input technique on each of the CCD structures to provide the optimum linearity at the voltage to charge conversion.

The circuit uses the floating gate technique to sense nine adjacent charge packets representing the array of 3x3 adjacent pixels. As the three adjacent lines of charge, representing the video data, are clocked through the array, the weighted sum of the charge or pixel magnitudes, at each clock sample, is applied to an 'on-chip' sample and hold. For example the voltage on the floating gate array, sensed by the sample and held at the nth clock period, T, is

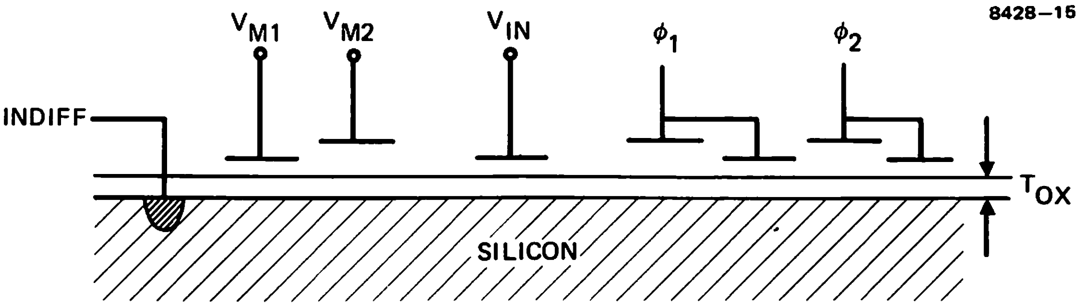


Figure 5. Schematic of Tompsett input structure.

$$V_0(nT) = \frac{\sum_{i=1}^{N} Q_i(nT)}{C_A} \qquad (2)$$

where C_A is the total capacitance of the array and connecting bus-line, $Q_i(nT)$ is the total charge under gate "i" at time nT, and N is the total number of gates in the array. Further, the charge packet in each CCD stage, q_i, is proportional to the corresponding picture intensity p_i, and the area of the individual gates a_i determines the proportion of the charge capacitively coupling to the array. Hence we can write

$$q_i(nT) = \text{const} \cdot p_i(nT) \qquad \text{(Here the constant relates to the input characteristics of the device.)} \qquad (3)$$

and

$$Q_i(nT) = a_i/a_c q_i(nT) \qquad \text{(Where } a_c \text{ is the effective area of a full CCD stage.)} \qquad (4)$$

Thus

$$V_0(nT) = \text{const} \cdot \sum_{i=1}^{N} \frac{a_i p_i(nT)}{C_A} \qquad (5)$$

which, for a two-dimensional 3x3 array, can be written in the form of the convolution

$$V_0(nT) = \frac{\sum_{i=1}^{N} Q_i(nT)}{C_A} \qquad (2)$$

where C_A is the total capacitance of the array and connecting bus-line, $Q_i(nT)$ is the total charge under gate "i" at time nT, and N is the total number of gates in the array. Further, the charge packet in each CCD stage, q_i, is proportional to the corresponding picture intensity p_i, and the area of the individual gates a_i determines the proportion of the charge capacitively coupling to the array. Hence we can write

$$q_i(nT) = \text{const} \cdot p_i(nT) \qquad \text{(Here the constant relates to the input characteristics of the device.)} \qquad (3)$$

and

$$Q_i(nT) = a_i/a_c q_i(nT) \qquad \text{(Where } a_c \text{ is the effective area of a full CCD stage.)} \qquad (4)$$

Thus

$$V_0(nT) = \text{const} \cdot \sum_{i=1}^{N} \frac{a_i p_i(nT)}{C_A} \qquad (5)$$

which, for a two-dimensional 3x3 array, can be written in the form of the convolution

$$V_0 = \text{const} \cdot \underline{A} * \underline{p} \tag{6}$$

where

$$\underline{A} = \begin{bmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \\ a_7 & a_8 & a_9 \end{bmatrix} = \begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix} \tag{7}$$

Hence the area, a_i, or length of each gate, must be proportional to the elements of $\underline{A}$. Since the length of each gate must be a positive value, the conventional approach would be to implement $\underline{A}$ as

$$|\underline{A}| = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \tag{8}$$

and connect each of the gates to either a positive or negative bus line of a differential amplifier. In practice, the differential amplifier, which itself can be implemented on-chip in n-MOS technology, becomes a significant portion of the total area of the chip, typically comparable, or perhaps even larger, than the CCD array itself. Further the differential amplifiers are themselves voltage controlled devices and the charge to voltage transition necessary can introduce nonlinearities and noise. For this reason, we employ a Hughes patented Displacement Charge Subtraction technique (DCS) [6], which implements $\underline{A}$ directly as

$$\begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix}$$

and results in a low capacitance technique which eliminates "common-mode" output. This technique has been shown to provide up to 90 dB dynamic range and 68 dB common mode rejection [6].

The processed outputs from the array are fed directly to 'on-chip' sample and hold circuits which eliminate clock feedthrough and rejects coherent noise sources. These circuits have been designed and simulated to run at 10 MHz data rate, with accuracy equivalent as six-bits. Finally, a source-follower output is provided for measuring the charge transfer efficiency and for diagnostic purposes.

II.B. 7x7 Mask Programmable Kernel

In the April 1978 Semi-Annual Report, a number of processing algorithms were discussed which use a 7x7 array both with a binary checkerboard weighting for image decomposition, and as a version of unsharp masking or deblurring. Because of the interest in this kernel size, we have built a mask programmable array which can be used to form a variety of operators. The basic concept consists of a 7x7 array of CCD stages which can be operated from seven parallel adjacent video lines. The basic structure is shown in Figure 6. Each of the seven linear arrays can be operated at 7.5 MHz with a dynamic range and intensity resolution equivalent to six bits. Again, a source follower is included for test and evaluation purposes and to measure the linearity and charge transfer efficiency. We have included two on-chip sample and holds so that two orthogonal vector operations can be performed. An MOS absolute value circuit similar to the one we used on the previous test chip is included so that two combinations of individual vector operations, such as used in edge detection, can be

$$\begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix}$$

and results in a low capacitance technique which eliminates "common-mode" output. This technique has been shown to provide up to 90 dB dynamic range and 68 dB common mode rejection [6].

The processed outputs from the array are fed directly to 'on-chip' sample and hold circuits which eliminate clock feedthrough and rejects coherent noise sources. These circuits have been designed and simulated to run at 10 MHz data rate, with accuracy equivalent as six-bits. Finally, a source-follower output is provided for measuring the charge transfer efficiency and for diagnostic purposes.

II.B. 7x7 Mask Programmable Kernel

In the April 1978 Semi-Annual Report, a number of processing algorithms were discussed which use a 7x7 array both with a binary checkerboard weighting for image decomposition, and as a version of unsharp masking or deblurring. Because of the interest in this kernel size, we have built a mask programmable array which can be used to form a variety of operators. The basic concept consists of a 7x7 array of CCD stages which can be operated from seven parallel adjacent video lines. The basic structure is shown in Figure 6. Each of the seven linear arrays can be operated at 7.5 MHz with a dynamic range and intensity resolution equivalent to six bits. Again, a source follower is included for test and evaluation purposes and to measure the linearity and charge transfer efficiency. We have included two on-chip sample and holds so that two orthogonal vector operations can be performed. An MOS absolute value circuit similar to the one we used on the previous test chip is included so that two combinations of individual vector operations, such as used in edge detection, can be

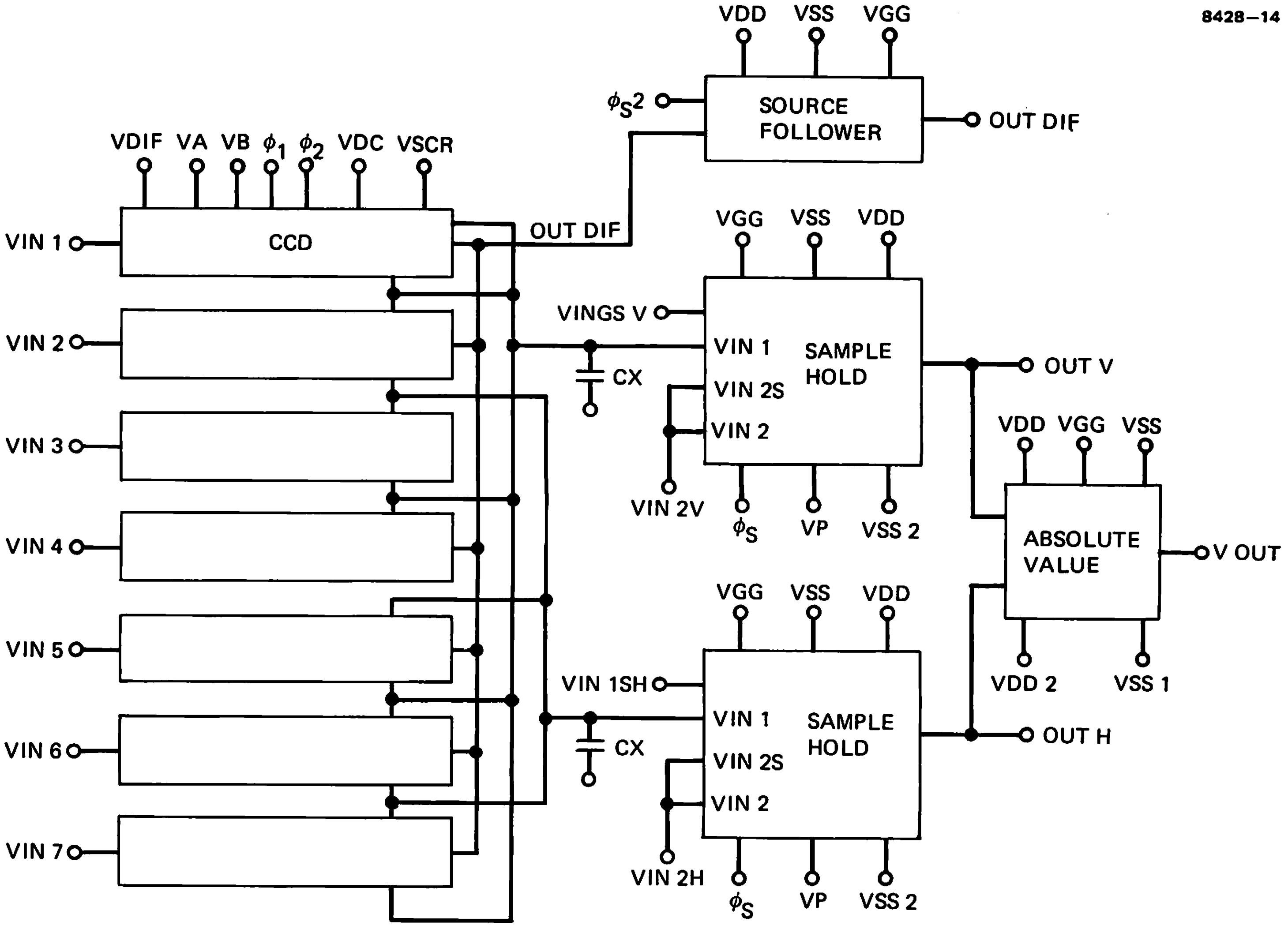


Figure 6. 7 x 7 Mask programmable array.

achieved. A schematic of this circuit which has been designed and simulated to operate at full video rates is shown in Figure 7. With this basic structure, we can use a mask change to perform a variety of different operations. Basically only those levels which determine the filter weightings and the bus line interconection need be changed to provide each of the operations discussed in the IPI Semi-Annual Report, April 1978. This technique provides a very flexible and cost effective way of performing a wide variety of 7x7 algorithms.

Initially, we have designed the mask to perform a 7x7 edge detection operation with radially symmetric weights. The weights used are given by

$$\begin{gathered}
H_{\underline{x}} = \begin{bmatrix}
.08 & .17 & .25 & .34 & .25 & .17 & .08 \\
.17 & .34 & .51 & .64 & .51 & .34 & .17 \\
.25 & .51 & .64 & 1.0 & .64 & .51 & .25 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-.25 & -.51 & -.64 & -1.0 & -.64 & -.51 & -.25 \\
-.17 & -.34 & -.51 & -.64 & -.51 & -.34 & -.17 \\
-.08 & -.17 & -.25 & -.34 & -.25 & -.17 & -.08
\end{bmatrix} \\
H_{\underline{Y}} = \begin{bmatrix}
.08 & .17 & .25 & 0 & -.25 & -.17 & -.08 \\
.17 & .34 & .51 & 0 & -.51 & -.34 & -.17 \\
.25 & .51 & .64 & 0 & -.64 & -.51 & -.25 \\
.34 & .64 & 1.0 & 0 & -1.0 & -.64 & -.34 \\
.25 & .51 & .64 & 0 & -.64 & -.51 & -.25 \\
.17 & .34 & .51 & 0 & -.51 & -.34 & -.17 \\
.08 & .17 & .25 & 0 & -.25 & -.17 & -.08
\end{bmatrix}
\end{gathered} \tag{9}$$

achieved. A schematic of this circuit which has been designed and simulated to operate at full video rates is shown in Figure 7. With this basic structure, we can use a mask change to perform a variety of different operations. Basically only those levels which determine the filter weightings and the bus line interconection need be changed to provide each of the operations discussed in the IPI Semi-Annual Report, April 1978. This technique provides a very flexible and cost effective way of performing a wide variety of 7x7 algorithms.

Initially, we have designed the mask to perform a 7x7 edge detection operation with radially symmetric weights. The weights used are given by

$$\begin{gathered}
H_{\underline{x}} = \begin{bmatrix}
.08 & .17 & .25 & .34 & .25 & .17 & .08 \\
.17 & .34 & .51 & .64 & .51 & .34 & .17 \\
.25 & .51 & .64 & 1.0 & .64 & .51 & .25 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-.25 & -.51 & -.64 & -1.0 & -.64 & -.51 & -.25 \\
-.17 & -.34 & -.51 & -.64 & -.51 & -.34 & -.17 \\
-.08 & -.17 & -.25 & -.34 & -.25 & -.17 & -.08
\end{bmatrix} \\
H_{\underline{Y}} = \begin{bmatrix}
.08 & .17 & .25 & 0 & -.25 & -.17 & -.08 \\
.17 & .34 & .51 & 0 & -.51 & -.34 & -.17 \\
.25 & .51 & .64 & 0 & -.64 & -.51 & -.25 \\
.34 & .64 & 1.0 & 0 & -1.0 & -.64 & -.34 \\
.25 & .51 & .64 & 0 & -.64 & -.51 & -.25 \\
.17 & .34 & .51 & 0 & -.51 & -.34 & -.17 \\
.08 & .17 & .25 & 0 & -.25 & -.17 & -.08
\end{bmatrix}
\end{gathered} \tag{9}$$

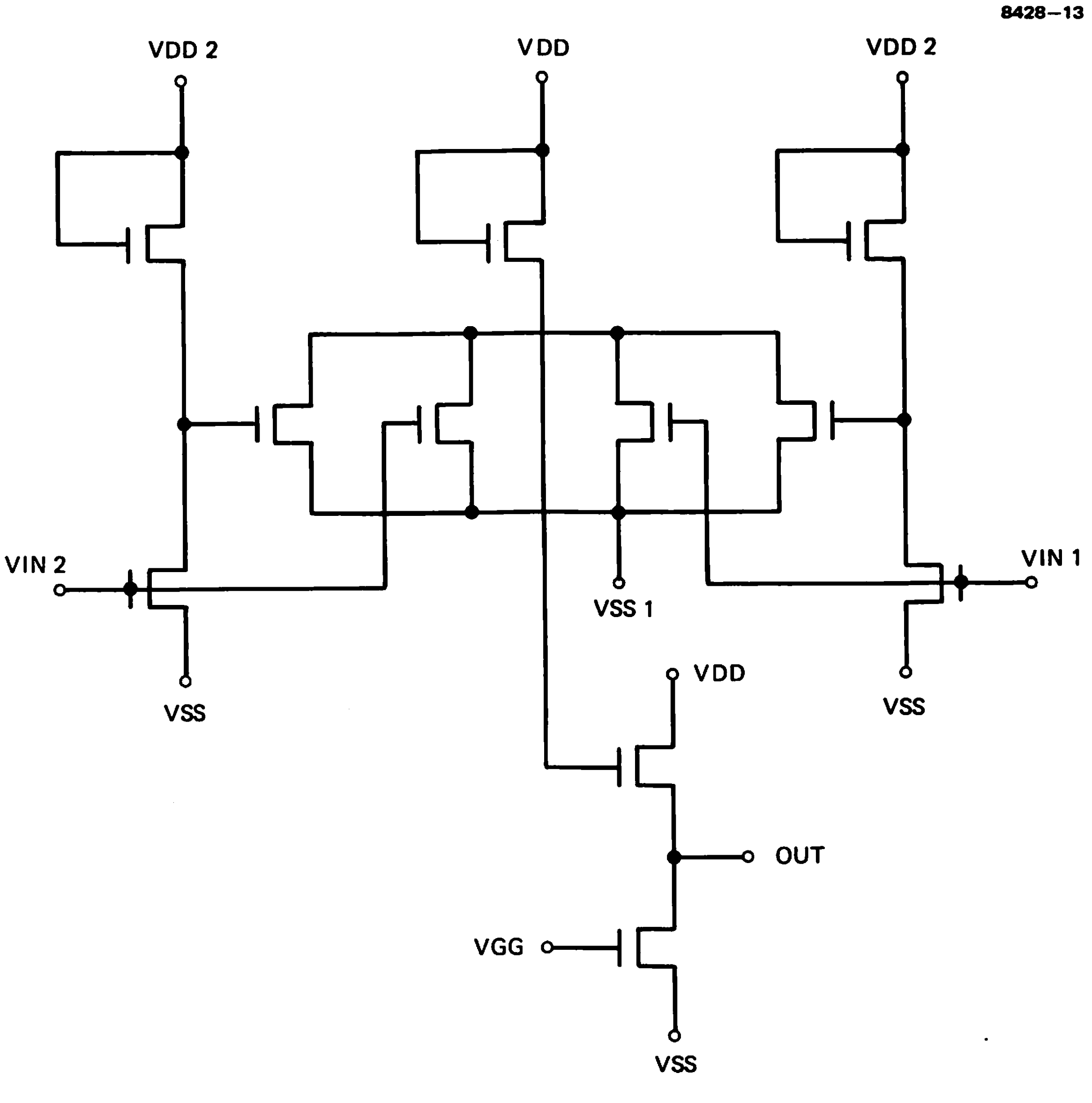


Figure 7. Schematic of "non-chip" MOS absolute value and summer circuit.

and the edge-detected output can be written as

$$S_i = |\underline{p} * H_{\underline{x}}| + |\underline{p} * H_{\underline{y}}| \qquad (10)$$

A detailed view of one linear CCD array to achieve this is shown in Figure 8. Here the weightings are arranged to be inversely proportional to their distance from the center of the array. In this way the edge value is concentrated or "focused" towards the center picture value, and the larger array size gives greater immunity to noise in the sensed image. We have performed a number of simulations using this operator and have found that on some images, it is preferable to the 3x3 Sobel. An example of this is shown in Figure 9.

We anticipate that the flexibility inherent in this approach and the options available on the circuit (sample and holds, absolute values, etc.) will allow a wide variety of useful operators including vector edge detection and checkerboard decomposition to be built very inexpensively.

II.C. 5x5 Programmable Array

One of the most interesting circuit functions available on Test Chip III is a 5x5 array which will be electrically programmable from external voltages. The programmable approach should allow many of the image understanding operations of interest, on a 5x5 array, to be performed with one circuit. The concept is shown in Figure 10. It has been designed to accept data at the standard 7.5 MHz video rate and enable the weighting functions to be changed at the frame rate of 30 Hz. Since each weighting node has been brought out directly to external pin of the 64 pin package, we can independently vary each element of the 25 point convolution, with an accuracy of approximately 2%. Further, since our aim is to drive the weights from a commercial microprocessor, we can in effect null out many of the processing inaccuracies and nonlinearities to obtain optimal performance. A schematic of the circuit is given in Figure 11. The concept of the and the edge-detected output can be written as

$$S_i = |\underline{p} * H_{\underline{x}}| + |\underline{p} * H_{\underline{y}}| \tag{10}$$

A detailed view of one linear CCD array to achieve this is shown in Figure 8. Here the weightings are arranged to be inversely proportional to their distance from the center of the array. In this way the edge value is concentrated or "focused" towards the center picture value, and the larger array size gives greater immunity to noise in the sensed image. We have performed a number of simulations using this operator and have found that on some images, it is preferable to the 3x3 Sobel. An example of this is shown in Figure 9.

We anticipate that the flexibility inherent in this approach and the options available on the circuit (sample and holds, absolute values, etc.) will allow a wide variety of useful operators including vector edge detection and checkerboard decomposition to be built very inexpensively.

II.C. 5x5 Programmable Array

One of the most interesting circuit functions available on Test Chip III is a 5x5 array which will be electrically programmable from external voltages. The programmable approach should allow many of the image understanding operations of interest, on a 5x5 array, to be performed with one circuit. The concept is shown in Figure 10. It has been designed to accept data at the standard 7.5 MHz video rate and enable the weighting functions to be changed at the frame rate of 30 Hz. Since each weighting node has been brought out directly to external pin of the 64 pin package, we can independently vary each element of the 25 point convolution, with an accuracy of approximately 2%. Further, since our aim is to drive the weights from a commercial microprocessor, we can in effect null out many of the processing inaccuracies and nonlinearities to obtain optimal performance. A schematic of the circuit is given in Figure 11. The concept of the

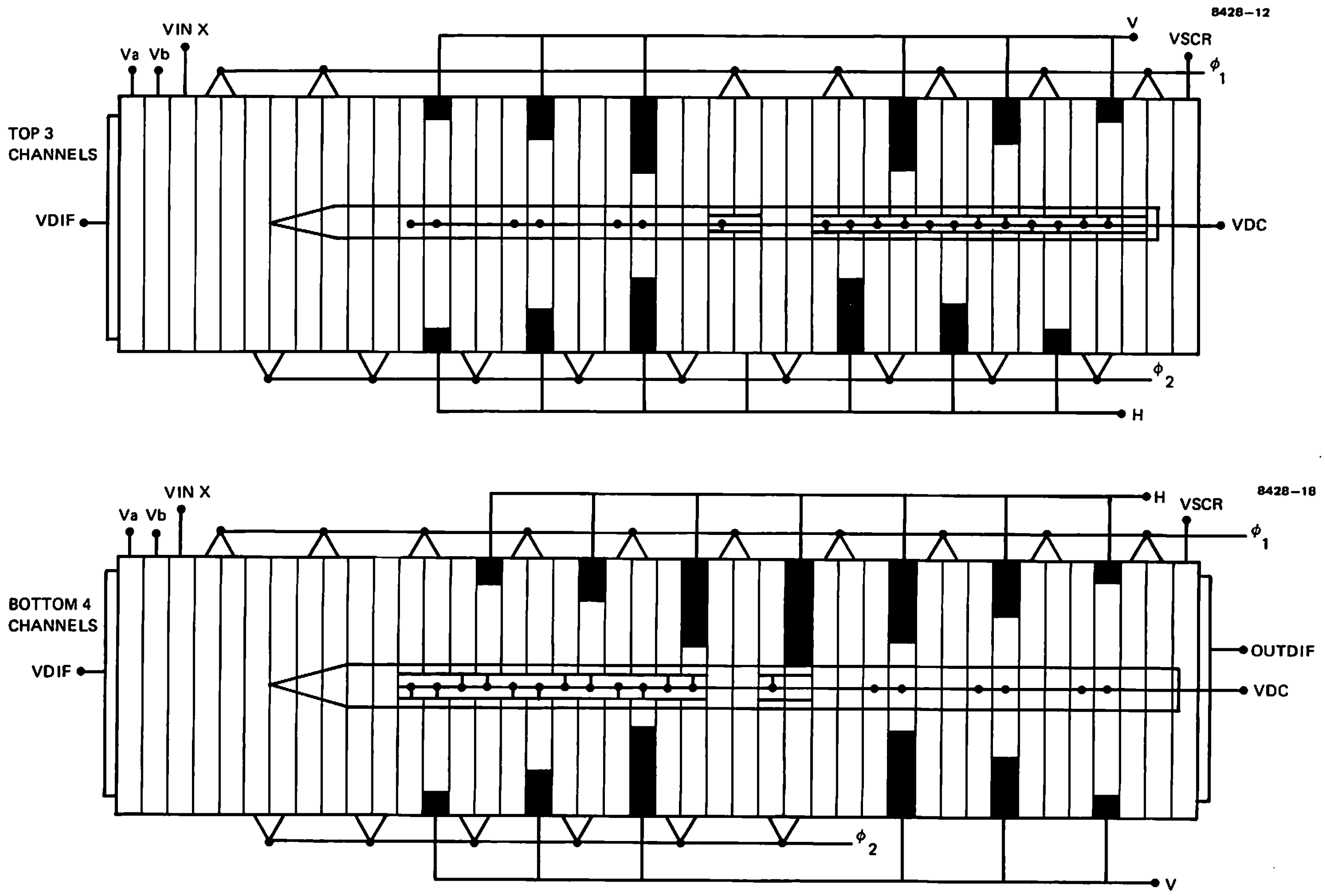
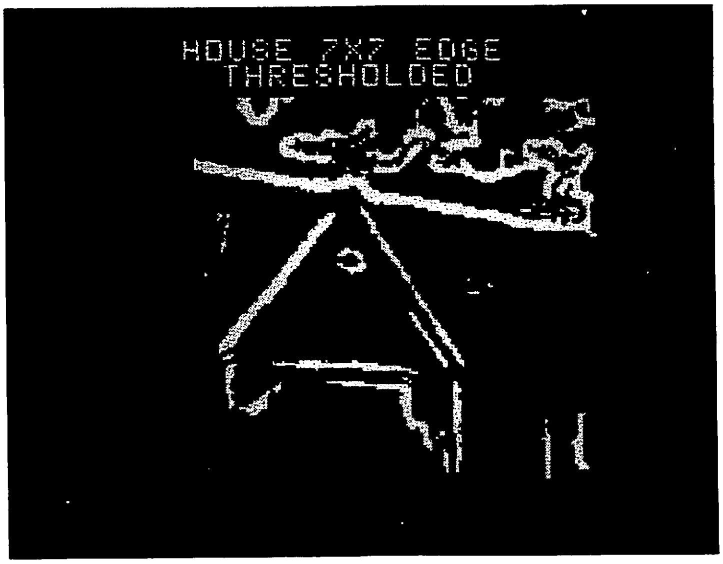


Figure 8. Mask structure programmed for 7 x 7 edge detection.

8428–11



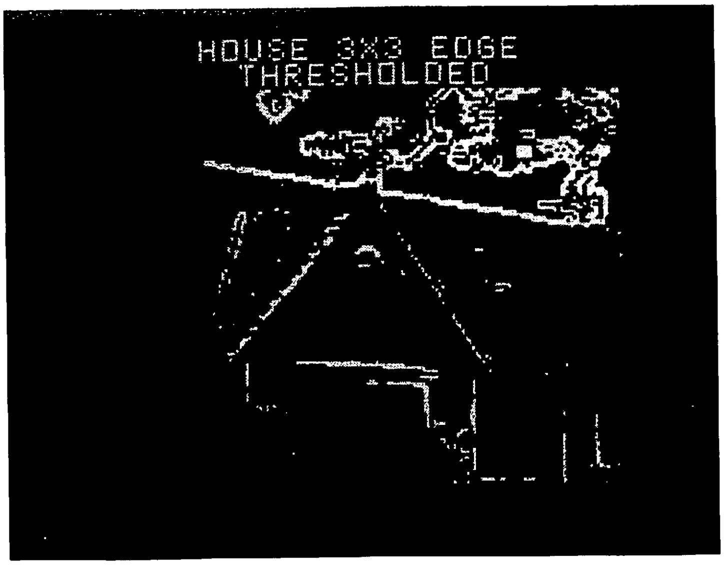


Figure 9. Comparison of 7 x 7 and 3 x 3 edge detection algorithm.

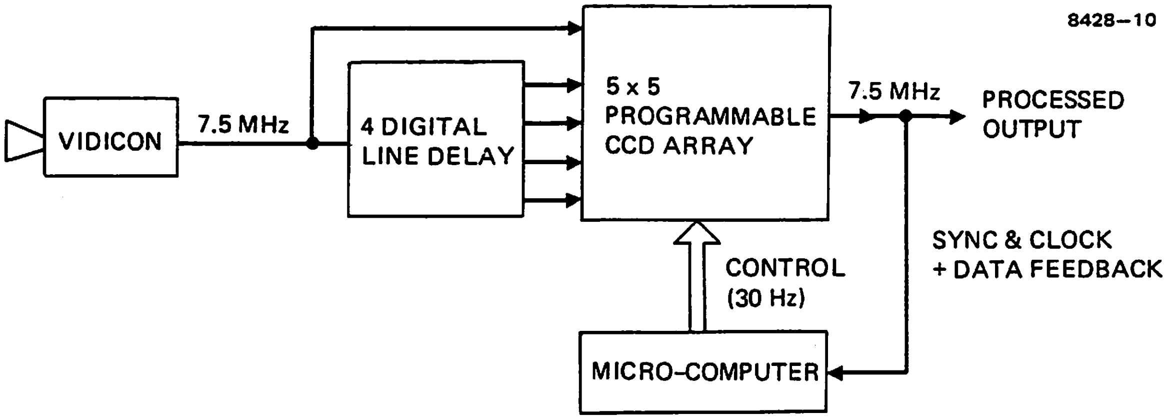


Figure 10. Schematic of test system for programmable filter incorporating micro-computer control.

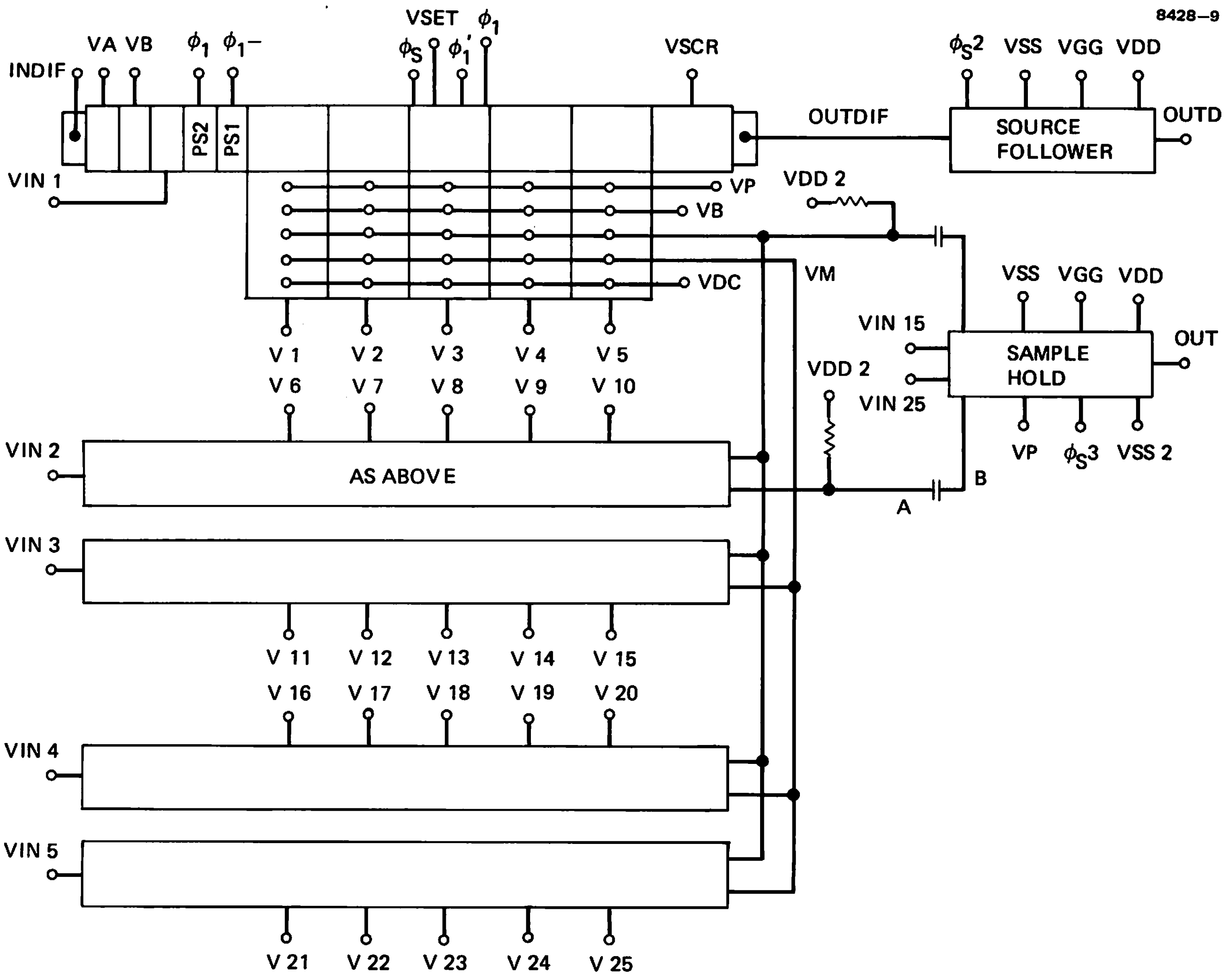


Figure 11(a). Schematic of 5 x 5 programmable filter.

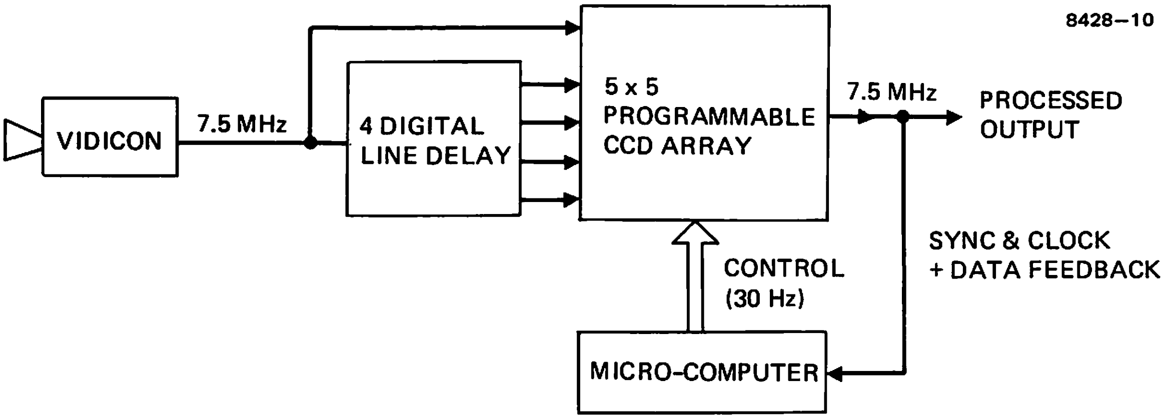


Figure 10. Schematic of test system for programmable filter incorporating micro-computer control.

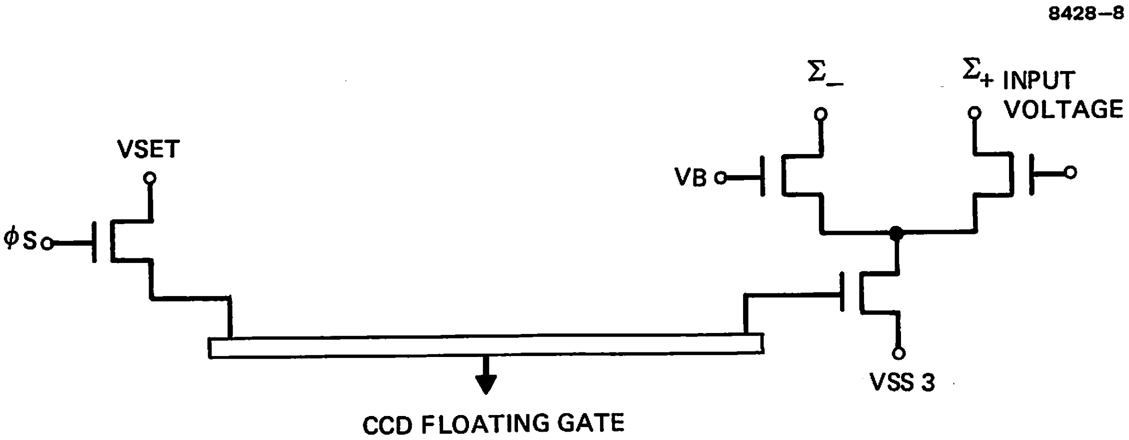


Figure 11(b). Schematic of programmable tap weight.

weighting technique relies on each floating gate being connected to the output summing bus through a MOS-FET chain. The gate voltages on these transistors in effect determine the gain of the weightings.

One of the significances of this approach is that we will be able to perform extensive experiments to determine the advantages and added flexibility achieved by incorporating both high speed (7.5 MHz) parallel analog processing and the lower speed programmable digital computations available from the microprocessor. This should put us in an excellent position to proceed with our higher level processing such as segmentation based on analog features, as discussed in Section IV, and also give us valuable information about the low level-high level interface.

II.D. 5x5 "Plus-Shaped" Median Filtering

Both USC IPI and Hughes Research Laboratories (HRL) have done extensive simulation on median filtering. The median operator is an obvious candidate for preprocessing and can be very useful for both rejection of impulsive noise and to overcome defects in the imaging system, etc. Both HRL and IPI studies show that a "plus-shaped" array with nine pixels is optimum for many of the images of interest. Perhaps the most direct approach to a median filter is to perform a sort operation and then choose the fifth element in the stack (for a 5x5 cross). To do this, n(n-1)/2 or 36 comparators are required. The conventional approach is to form the ladder network of "bubble-sort" array shown in Figure 12 for five inputs. Here each comparator module (CM) has three basic states, depending on the relative magnitude of the two inputs "a" and "b." In the configuration shown, if $a \geq b$ it acts simply as two parallel one element delays. For b>a however, it acts as a cross-bar switch reversing the two outputs. The effect after a total of six comparisons is to provide nine parallel outputs order in increasing magnitude, whereupon the center output is the median value.

weighting technique relies on each floating gate being connected to the output summing bus through a MOS-FET chain. The gate voltages on these transistors in effect determine the gain of the weightings.

One of the significances of this approach is that we will be able to perform extensive experiments to determine the advantages and added flexibility achieved by incorporating both high speed (7.5 MHz) parallel analog processing and the lower speed programmable digital computations available from the microprocessor. This should put us in an excellent position to proceed with our higher level processing such as segmentation based on analog features, as discussed in Section IV, and also give us valuable information about the low level-high level interface.

II.D. 5x5 "Plus-Shaped" Median Filtering

Both USC IPI and Hughes Research Laboratories (HRL) have done extensive simulation on median filtering. The median operator is an obvious candidate for preprocessing and can be very useful for both rejection of impulsive noise and to overcome defects in the imaging system, etc. Both HRL and IPI studies show that a "plus-shaped" array with nine pixels is optimum for many of the images of interest. Perhaps the most direct approach to a median filter is to perform a sort operation and then choose the fifth element in the stack (for a 5x5 cross). To do this, n(n-1)/2 or 36 comparators are required. The conventional approach is to form the ladder network of "bubble-sort" array shown in Figure 12 for five inputs. Here each comparator module (CM) has three basic states, depending on the relative magnitude of the two inputs "a" and "b." In the configuration shown, if $a \geq b$ it acts simply as two parallel one element delays. For $b>a$ however, it acts as a cross-bar switch reversing the two outputs. The effect after a total of six comparisons is to provide nine parallel outputs order in increasing magnitude, whereupon the center output is the median value.

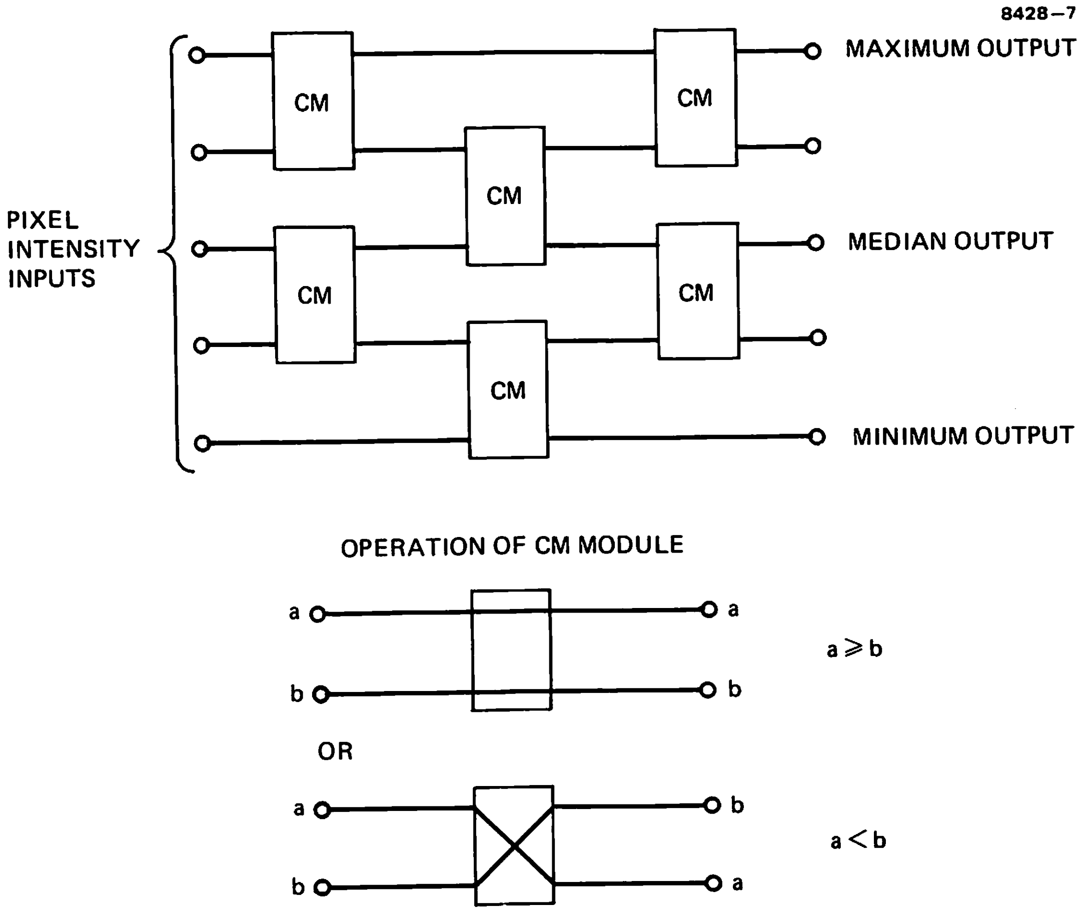


Figure 12. Schematic of real-time median filter operator for 5 pixels.

This structure can be built directly in MOS/LS using MOS-FETs to provide a result equivalent to 7 bits, and run at 7.5 MHz. It can also be built very effectively as a compact CCD structure. In either case, it can be built in to a modular design which will allow the array size to be increased by adding parallel chips. Our present design is a direct MOS implementation which uses external delays which determine the pixel array shape. In this way we can perform the operation over a variety of kernels.

II.E. 26x26 Bi-Polar Convolution Filter

We have included on this chip a processing algorithm suggested by Professor David Marr and his colleagues at MIT. It can be used as part of the preprocessing required for a human vision sytem. From a technology standpoint, it is interesting because it has a significantly larger kernel size than the arrays built to date and requires high accuracy. The full kernel is shown in Figure 13, consisting of a 26x26 element with a weighting range of approximately 1:150. In order to provide the most conservative approach and hence the highest probability of success, we have built a 26x13 element array and intend to use two chips for the full convolution. The circularly symmetric nature of the required convolution allows two identical arrays to be used with modified input structures, as shown in Figure 14. Further, we have decided to build the array with three separate outputs which will be helpful in the test and evaluation stage, and more significantly, has enabled us to scale or normalize the weighting to achieve higher accuracy. As shown in Figure 15, we have normalized all the weightings such that they lie within a range of 1 to 15. To do this, we have built all the negative weights on the outer rim starting at 0 and decreasing to -15 and going back to zero on a scale where 8 m represents one unit of weighting, i.e., with gate lengths of 0 to 134 m. These gates are each connected to a summing bus and brought to the negative input of an 'on-chip' MOS differential amplifier. The single annulus of weights ranging from zero to +15 are also built on the same scale and brought out to the

This structure can be built directly in MOS/LS using MOS-FETs to provide a result equivalent to 7 bits, and run at 7.5 MHz. It can also be built very effectively as a compact CCD structure. In either case, it can be built in to a modular design which will allow the array size to be increased by adding parallel chips. Our present design is a direct MOS implementation which uses external delays which determine the pixel array shape. In this way we can perform the operation over a variety of kernels.

II.E. 26x26 Bi-Polar Convolution Filter

We have included on this chip a processing algorithm suggested by Professor David Marr and his colleagues at MIT. It can be used as part of the preprocessing required for a human vision sytem. From a technology standpoint, it is interesting because it has a significantly larger kernel size than the arrays built to date and requires high accuracy. The full kernel is shown in Figure 13, consisting of a 26x26 element with a weighting range of approximately 1:150. In order to provide the most conservative approach and hence the highest probability of success, we have built a 26x13 element array and intend to use two chips for the full convolution. The circularly symmetric nature of the required convolution allows two identical arrays to be used with modified input structures, as shown in Figure 14. Further, we have decided to build the array with three separate outputs which will be helpful in the test and evaluation stage, and more significantly, has enabled us to scale or normalize the weighting to achieve higher accuracy. As shown in Figure 15, we have normalized all the weightings such that they lie within a range of 1 to 15. To do this, we have built all the negative weights on the outer rim starting at 0 and decreasing to -15 and going back to zero on a scale where 8 m represents one unit of weighting, i.e., with gate lengths of 0 to 134 m. These gates are each connected to a summing bus and brought to the negative input of an 'on-chip' MOS differential amplifier. The single annulus of weights ranging from zero to +15 are also built on the same scale and brought out to the

8428-6

0	0	0	0	0	0	0	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	-1	-1	-1	-1	-2	-2	-2	-2	-2	-2	-1	-1	-1	-1	0	0	0	0	0	0
0	0	0	0	-1	-1	-1	-2	-2	-3	-3	-3	-3	-3	-3	-3	-3	-2	-2	-1	-1	-1	0	0	0	0
0	0	0	-1	-1	-1	-2	-3	-3	-4	-5	-5	-6	-6	-5	-5	-4	-3	-3	-2	-1	-1	-1	0	0	0
0	0	-1	-1	-1	-2	-3	-4	-5	-7	-8	-9	-9	-9	-9	-8	-7	-5	-4	-3	-2	-1	-1	-1	0	0
0	0	-1	-1	-2	-3	-5	-6	-8	-10	-11	-12	-13	-13	-12	-11	-10	-8	-6	-5	-3	-2	-1	-1	0	0
0	-1	-1	-2	-3	-5	-7	-9	-11	-13	-14	-15	-15	-15	-15	-14	-13	-11	-9	-7	-5	-3	-2	-1	-1	0
0	-1	-2	-3	-4	-6	-9	-12	-14	-15	-15	-14	-13	-13	-14	-15	-15	-14	-12	-9	-6	-4	-3	-2	-1	0
-1	-1	-2	-3	-5	-8	-11	-14	-15	-14	-9	-4	-1	-1	-4	-9	-14	-15	-14	-11	-8	-5	-3	-2	-1	-1
-1	-1	-3	-4	-7	-10	-13	-15	-14	-7	4	17	25	25	17	4	-7	-14	-15	-13	-10	-7	-4	-3	-1	-1
-1	-2	-3	-5	-8	-11	-14	-15	-9	4	25	48	63	63	48	25	4	-9	-15	-14	-11	-8	-5	-3	-2	-1
-1	-2	-3	-5	-9	-12	-15	-14	-4	17	48	81	102	102	81	48	17	-4	-14	-15	-12	-9	-5	-3	-2	-1
-1	-2	-3	-6	-9	-13	-15	-13	-1	25	63	102	126	126	102	63	25	-1	-13	-15	-13	-9	-6	-3	-2	-1
-1	-2	-3	-6	-9	-13	-15	-13	-1	25	63	102	126	126	102	63	25	-1	-13	-15	-13	-9	-6	-3	-2	-1
-1	-2	-3	-5	-9	-12	-15	-14	-4	17	48	81	102	102	81	48	17	-4	-14	-15	-12	-9	-5	-3	-2	-1
-1	-2	-3	-5	-8	-11	-14	-15	-9	4	25	48	63	63	48	25	4	-9	-15	-14	-11	-8	-5	-3	-2	-1
-1	-1	-3	-4	-7	-10	-13	-15	-14	-7	4	17	25	25	17	4	-7	-14	-15	-13	-10	-7	-4	-3	-1	-1
-1	-1	-2	-3	-5	-8	-11	-14	-15	-14	-9	-4	-1	-1	-4	-9	-14	-15	-14	-11	-8	-5	-3	-2	-1	-1
0	-1	-2	-3	-4	-6	-9	-12	-14	-15	-15	-14	-13	-13	-14	-15	-15	-14	-12	-9	-6	-4	-3	-2	-1	0
0	-1	-1	-2	-3	-5	-7	-9	-11	-13	-14	-15	-15	-15	-15	-14	-13	-11	-9	-7	-5	-3	-2	-1	-1	0
0	0	-1	-1	-2	-3	-5	-6	-8	-10	-11	-12	-13	-13	-12	-11	-10	-8	-6	-5	-3	-2	-1	-1	0	0
0	0	-1	-1	-1	-2	-3	-4	-5	-7	-8	-9	-9	-9	-9	-8	-7	-5	-4	-3	-2	-1	-1	-1	0	0
0	0	0	-1	-1	-1	-2	-3	-3	-4	-5	-5	-6	-6	-5	-5	-4	-3	-3	-2	-1	-1	-1	0	0	0
0	0	0	0	-1	-1	-1	-2	-2	-3	-3	-3	-3	-3	-3	-3	-3	-2	-2	-1	-1	-1	0	0	0	0
0	0	0	0	0	0	-1	-1	-1	-1	-2	-2	-2	-2	-2	-2	-1	-1	-1	-1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	0	0	0	0	0	0	0

Figure 13. Weighting required for 26 x 26 element array.

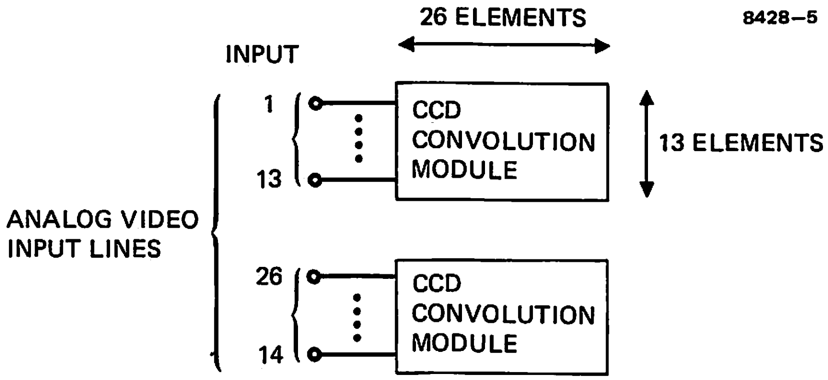


Figure 14. Input structure for 26 x 26 pixel array.

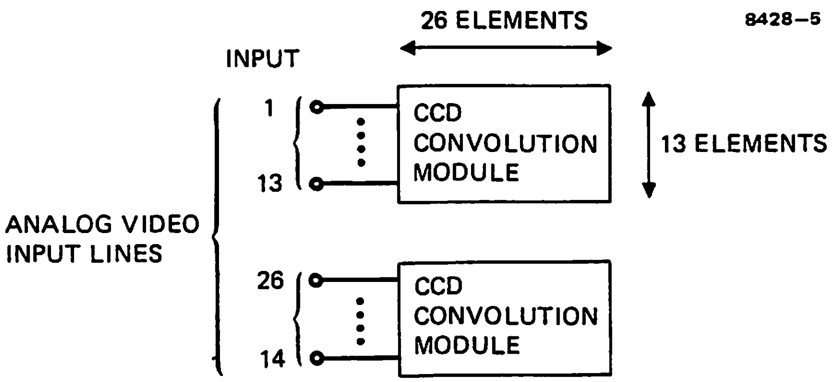


Figure 14. Input structure for 26 x 26 pixel array.

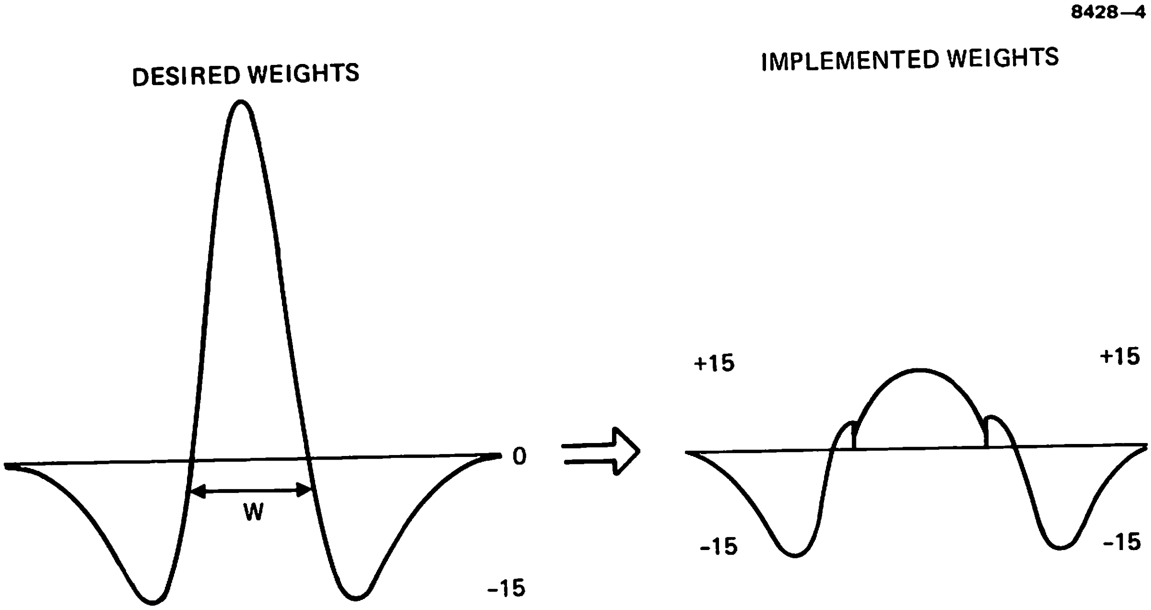


Figure 15. Modified weighting scheme used on 26 x 26 array.

opposite polarity input of the differential amplifiers as shown. In this way, a bipolar outputs representing input values -15 to +15 is available from the amplifier. The remaining weights ranging from +15 to +126 have all been normalized by division by 8 to again bring them approximately to the range of 15. These gates are therefore drawn at one-eighth scale relative to the other but an off-chip amplifier with a gain of 8 will be used to provide the required weightings. A schematic of the full circuit is shown in Figure 16.

II.F. Status of Test Chip III

Each of the above circuits have been designed, simulated and the final layout is near completion prior to sending out to the mask-maker. We anticipate the mask making will take 6-8 weeks and the silicon processing at Hughes will take approximately 6 weeks. Processed parts will therefore be available in July.

III. Test Facilities and Development of Necessary Peripheral Circuitry

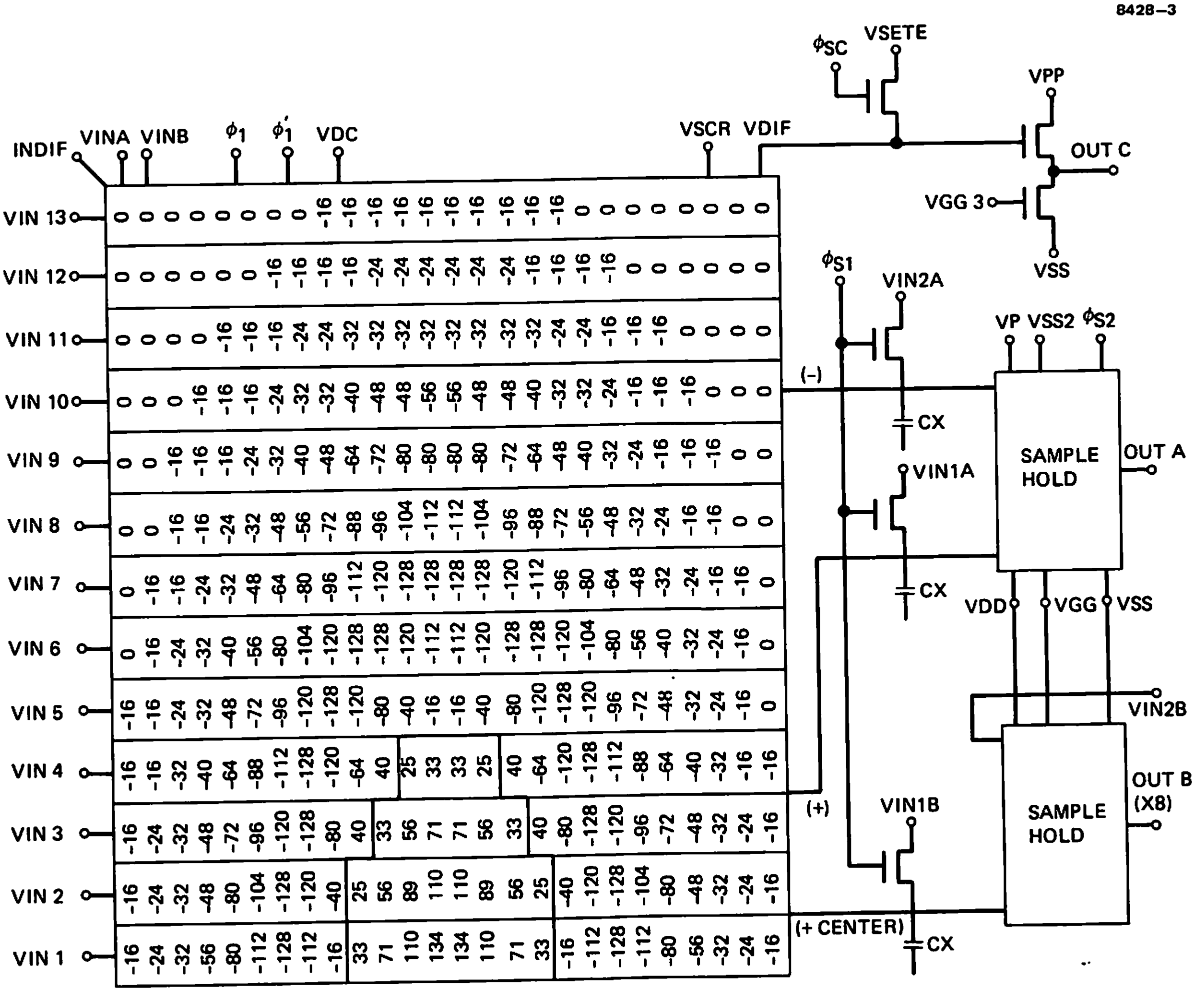
In order to both test electrically and later perform the detailed performance evaluation of each of the above circuits, it is necessary to build a number of specialized circuits. These include video line delays for signal formatting, CCD clocks, analog driver circuitry and some specialized interfaces beteen the video sensor (typically a vidicon), the microprocessor and our custom integrated circuits. Also as in our CCD structures we have not addressed the problem of video line delays or other picture formatting circuits, we have had to build external formatters. For the 3x3 arrays, used in the previous phase of the program, we employed two commercially available analog CCD line delays (CCD321) to obtain the necessary three lines of video. Since much of our future work on the program will concentrate on digital approaches, we have tried where possible to build the external control and interface circuits in digital technology. This will allow us to operate all the interface and formatting electronics, the sensor, tour opposite polarity input of the differential amplifiers as shown. In this way, a bipolar outputs representing input values -15 to +15 is available from the amplifier. The remaining weights ranging from +15 to +126 have all been normalized by division by 8 to again bring them approximately to the range of 15. These gates are therefore drawn at one-eighth scale relative to the other but an off-chip amplifier with a gain of 8 will be used to provide the required weightings. A schematic of the full circuit is shown in Figure 16.

II.F. Status of Test Chip III

Each of the above circuits have been designed, simulated and the final layout is near completion prior to sending out to the mask-maker. We anticipate the mask making will take 6-8 weeks and the silicon processing at Hughes will take approximately 6 weeks. Processed parts will therefore be available in July.

III. Test Facilities and Development of Necessary Peripheral Circuitry

In order to both test electrically and later perform the detailed performance evaluation of each of the above circuits, it is necessary to build a number of specialized circuits. These include video line delays for signal formatting, CCD clocks, analog driver circuitry and some specialized interfaces beteen the video sensor (typically a vidicon), the microprocessor and our custom integrated circuits. Also as in our CCD structures we have not addressed the problem of video line delays or other picture formatting circuits, we have had to build external formatters. For the 3x3 arrays, used in the previous phase of the program, we employed two commercially available analog CCD line delays (CCD321) to obtain the necessary three lines of video. Since much of our future work on the program will concentrate on digital approaches, we have tried where possible to build the external control and interface circuits in digital technology. This will allow us to operate all the interface and formatting electronics, the sensor, tour



VIN 13	0	0	0	0	0	0	0	0	-16	-16	-16	-16	-16	-16	-16	-16	-16	-16	0	0	0	0	0	0	0	0
VIN 12	0	0	0	0	0	0	-16	-16	-16	-16	-24	-24	-24	-24	-24	-24	-16	-16	-16	-16	0	0	0	0	0	0
VIN 11	0	0	0	0	-16	-16	-16	-24	-24	-32	-32	-32	-32	-32	-32	-32	-32	-24	-24	-16	-16	-16	0	0	0	0
VIN 10	0	0	0	-16	-16	-16	-24	-32	-32	-40	-48	-48	-56	-56	-48	-48	-40	-32	-32	-24	-16	-16	-16	0	0	0
VIN 9	0	0	-16	-16	-16	-24	-32	-40	-48	-64	-72	-80	-80	-80	-80	-72	-64	-48	-40	-32	-24	-16	-16	-16	0	0
VIN 8	0	0	-16	-16	-24	-32	-48	-56	-72	-88	-96	-104	-112	-112	-104	-96	-88	-72	-56	-48	-32	-24	-16	-16	0	0
VIN 7	0	-16	-16	-24	-32	-48	-64	-80	-96	-112	-120	-128	-128	-128	-128	-120	-112	-96	-80	-64	-48	-32	-24	-16	-16	0
VIN 6	0	-16	-24	-32	-40	-56	-80	-104	-120	-128	-128	-120	-112	-112	-120	-128	-128	-120	-104	-80	-56	-40	-32	-24	-16	0
VIN 5	-16	-16	-24	-32	-48	-72	-96	-120	-128	-120	-80	-40	-16	-16	-40	-80	-120	-128	-120	-96	-72	-48	-32	-24	-16	0
VIN 4	-16	-16	-32	-40	-64	-88	-112	-128	-120	-64	40	25	33	33	25	40	-64	-120	-128	-112	-88	-64	-40	-32	-16	-16
VIN 3	-16	-24	-32	-48	-72	-96	-120	-128	-80	40	33	56	71	71	56	33	40	-80	-128	-120	-96	-72	-48	-32	-24	-16
VIN 2	-16	-24	-32	-48	-80	-104	-128	-120	-40	25	56	89	110	110	89	56	25	-40	-120	-128	-104	-80	-48	-32	-24	-16
VIN 1	-16	-24	-32	-56	-80	-112	-128	-112	-16	33	71	110	134	134	110	71	33	-16	-112	-128	-112	-80	-56	-32	-24	-16

Figure 16. 26 x 13 pixel array implemented on test chip III.

processor and, when appropriate, the microprocesor from the same master clock. To date we have designed and are nearing the completion of a six line delay, using commercial digital circuits to provide the seven adjacent lines which will be used for our 3x3, 5x5 and 7x7 processing. Initially the larger array (26x26 pixel kernel) will be tested in a modular way prior to building the necessary 14 line box. We have also started analysis and design work on the necessary two phase clocking system and the diffusion pulses and resets, etc. This work will continue during the next six month period.

A full schematic of the new test facilites required to exercise our circuits is shown in Figure 17. We have decided to incorporate a scan converter system as shown which will allow us to vary both the spatial and temporal resolution of our processing. The architecture of this system has now been designed and we are starting some of the detailed design and fabrication which we will discuss in more detail in the next report.

IV. Concept Developtment for Higher Level Processing

As stated earlier, it is our philosophy to perform all, or much, of the low-level preprocessig functions by high-speed parallel analog operations. The essentially pipelined structure of the CCD transversal filters make this approach optimum for operations requiring a relatively low number of computations on each pixel. At the higher level where successive multiplications or high order powers of essentially analog (or six bit) intensity levels are often required, the dynamic range and accuracy requirements essentially preclude analog operation. We have spent a considerable amount of time in this phase of the program addressing this issue. As a specific operation to analyze, we have chosen the texture processor of Professor W.K. Pratt [1]. A schematic of this operation is given in Figure 16. It consists basically of a Sobel or Laplacian operation followed by moment operations of the form processor and, when appropriate, the microprocesor from the same master clock. To date we have designed and are nearing the completion of a six line delay, using commercial digital circuits to provide the seven adjacent lines which will be used for our 3x3, 5x5 and 7x7 processing. Initially the larger array (26x26 pixel kernel) will be tested in a modular way prior to building the necessary 14 line box. We have also started analysis and design work on the necessary two phase clocking system and the diffusion pulses and resets, etc. This work will continue during the next six month period.

A full schematic of the new test facilites required to exercise our circuits is shown in Figure 17. We have decided to incorporate a scan converter system as shown which will allow us to vary both the spatial and temporal resolution of our processing. The architecture of this system has now been designed and we are starting some of the detailed design and fabrication which we will discuss in more detail in the next report.

IV. Concept Developtment for Higher Level Processing

As stated earlier, it is our philosophy to perform all, or much, of the low-level preprocessig functions by high-speed parallel analog operations. The essentially pipelined structure of the CCD transversal filters make this approach optimum for operations requiring a relatively low number of computations on each pixel. At the higher level where successive multiplications or high order powers of essentially analog (or six bit) intensity levels are often required, the dynamic range and accuracy requirements essentially preclude analog operation. We have spent a considerable amount of time in this phase of the program addressing this issue. As a specific operation to analyze, we have chosen the texture processor of Professor W.K. Pratt [1]. A schematic of this operation is given in Figure 16. It consists basically of a Sobel or Laplacian operation followed by moment operations of the form

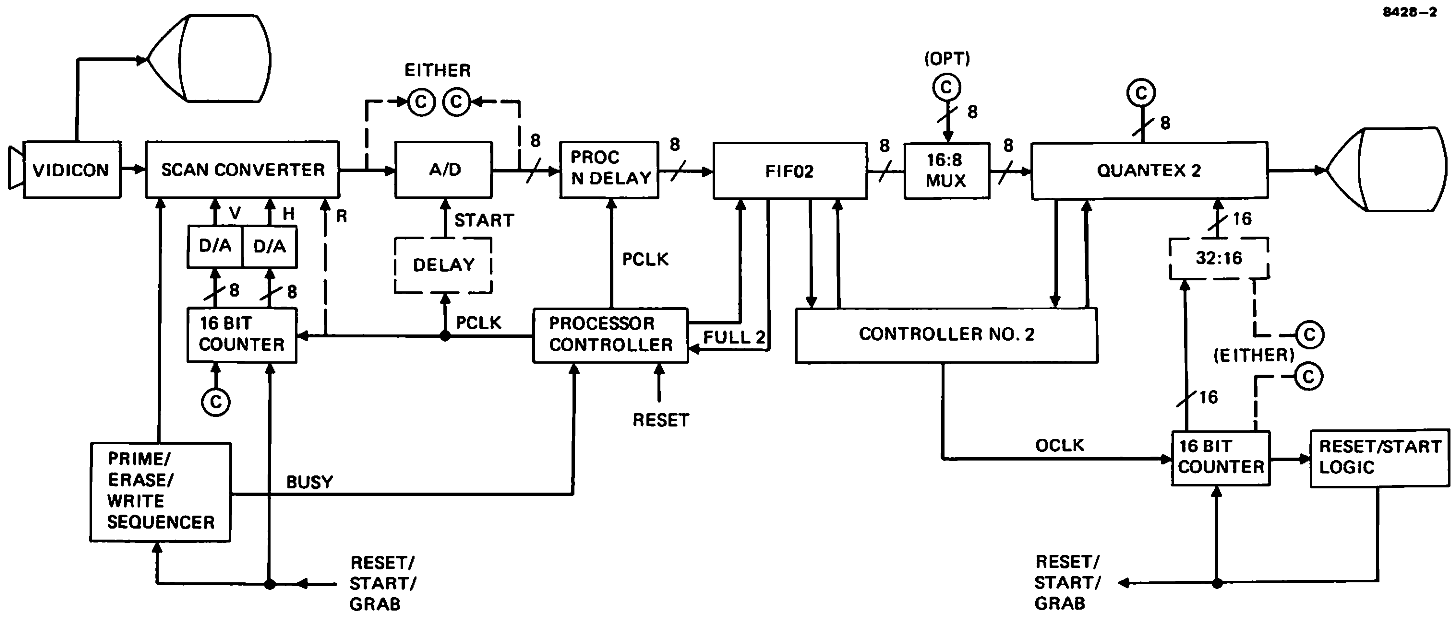


Figure 17. Schematic of test facility required for test chip III.

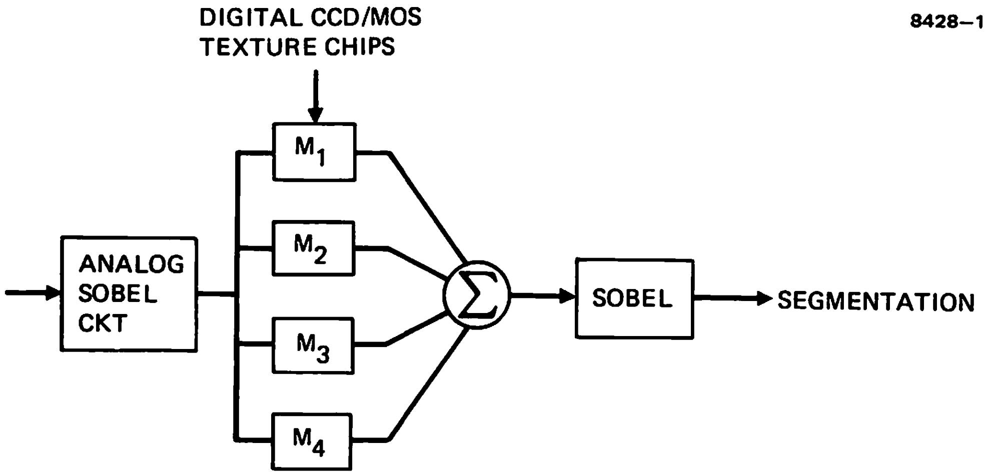


Figure 18. Concept of texture processing using moment calculations.

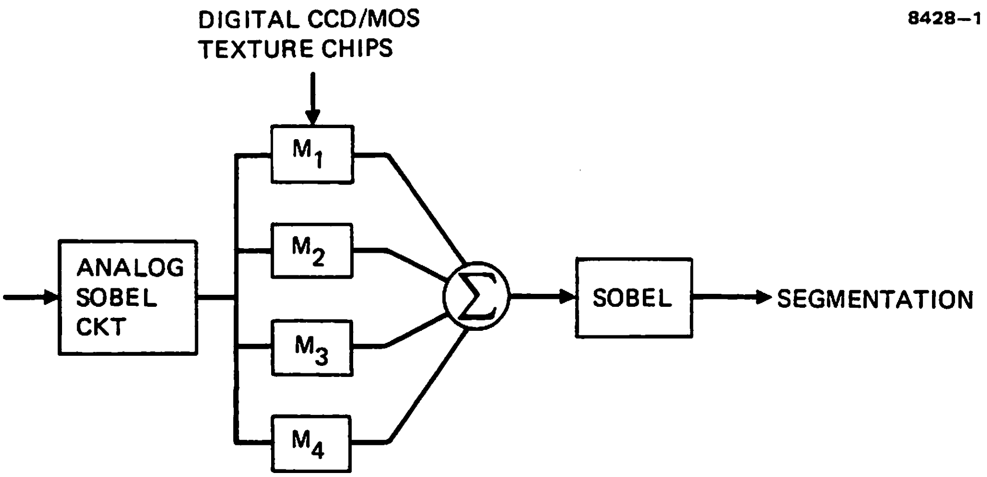


Figure 18. Concept of texture processinɡ using moment calculations.

$$M_n = \sum^{N} (p_i-\bar{p})^n \tag{11}$$

where p_i is the analog picture intensity, $\bar{p}$ is the average picture intensity over a given sized kernel and N represents the number of pixels in the kernel, n is the order of the moment, typically 1 through 4. From our previous work on this program we already have much of this hardware available. For example, the Sobel operator, the Laplacian and the local averager have all been built and demonstrated. The kernel size in our work to date (in Test Chip I and II) is limited to 3x3, but in our later circuits (Test Chip III), we will demonstrate a 26x26 pixel kernel. For effective texture analysis, a 225 element array (15x15) is apparently sufficient and hence we anticipate little difficulty in providing a sufficiently large kernel. However, if we assume the original pixel intensity can vary over 6 bits or 1 part in 64, we can, under the worst conditions, anticipate a required output dynamic range of 24 bits or 1 part in 10 [7]. This clearly is far too large for a direct analog implementation. Particularly as in subsequent processes, we might expect to calculate small differences of very large numbers (between the normalized third and fourth moment, for example). A digital approach seems to be essential on these grounds.

At first sight, it might appear that if we calculate $(p_i-\bar{p})^n$ directly, since the individual picture elements p_i are typically closely distributed about the local mean $\bar{p}$, $(p_i-\bar{p})$ will be a small number. (This will of course, be true for those images which have a low variance.) However, this approach causes considerable difficulty in the computation. For example, for each new picture element, which will occur approximately every 100 nanosec, we will be required to calculate $\Sigma(p_i-\bar{p})^n$ in its entirety, because $\bar{p}$ will also change at the pixel rate. For a 15x15 window, calculating just the first four moments, this will result in a through put of $1.3x10^{10}$ operations per second, the majority of which will be multiplications. This is clearly an inappropriate approach since a high speed multiply might take 50 nanosec or more in the fastest high speed emitter-coupled-logic (ECL) technology resulting in approximately 10 mins to process one frame. Several hundred channels would be required in a parallel architecture, requiring a very large amount of power and circuitry, to achieve real-time operation.

Clearly a preferable approach is to perform the non-centered moments

$$\begin{aligned}
M_1 &= \frac{1}{N}\Sigma(p_i) \\
M_2 &= -\frac{1}{N^2}(\Sigma p_i)^2+\frac{1}{N}\Sigma p_i^2 \\
M_3 &= -\frac{1}{N^3}(\Sigma p_i)^3-\frac{3}{N^2}\Sigma p_i^2(\Sigma p_i)+\frac{1}{N}\Sigma p_i^3 \\
M_4 &= -\frac{3}{N^4}(\Sigma p_i)^4+\frac{6}{N^3}\Sigma p_i^2(\Sigma p_i)^2-\frac{4}{N^2}\Sigma p_i^3(\Sigma p_i)+\frac{1}{N}\Sigma p_i^4
\end{aligned} \tag{12}$$

In this way, the partial products can be calculated and stored, and with each new pixel we are required only to subtract the contributions of the oldest pixel from the summation and add the newest. This can reduce the calculation rate by several orders of magnitude and enable real-time or near real-time operation.

We have spent considerable time configuring an architecture which would be built from a number of identical modules and hence be appropriate to the new generation of LS and VLSI design techniques and can be implemented in present state of the art technology. Our technology choice again would be MOS/CCD primarily because of the lower power and small size of the CCD logic elements [7] and because

$$M_n = \sum^{N} (p_i-\overline{p})^n \qquad (11)$$

where p_i is the analog picture intensity, $\overline{p}$ is the average picture intensity over a given sized kernel and N represents the number of pixels in the kernel, n is the order of the moment, typically 1 through 4. From our previous work on this program we already have much of this hardware available. For example, the Sobel operator, the Laplacian and the local averager have all been built and demonstrated. The kernel size in our work to date (in Test Chip I and II) is limited to 3x3, but in our later circuits (Test Chip III), we will demonstrate a 26x26 pixel kernel. For effective texture analysis, a 225 element array (15x15) is apparently sufficient and hence we anticipate little difficulty in providing a sufficiently large kernel. However, if we assume the original pixel intensity can vary over 6 bits or 1 part in 64, we can, under the worst conditions, anticipate a required output dynamic range of 24 bits or 1 part in 10 [7]. This clearly is far too large for a direct analog implementation. Particularly as in subsequent processes, we might expect to calculate small differences of very large numbers (between the normalized third and fourth moment, for example). A digital approach seems to be essential on these grounds.

At first sight, it might appear that if we calculate $(p_i-\overline{p})^n$ directly, since the individual picture elements p_i are typically closely distributed about the local mean $\overline{p}$, $(p_i-\overline{p})$ will be a small number. (This will of course, be true for those images which have a low variance.) However, this approach causes considerable difficulty in the computation. For example, for each new picture element, which will occur approximately every 100 nanosec, we will be required to calculate $\Sigma(p_i-\overline{p})^n$ in its entirety, because $\overline{p}$ will also change at the pixel rate. For a 15x15 window, calculating just the first four moments, this will result in a through put of $1.3\text{x}10^{10}$ operations per an integral part of the processor would consist of serial high speed memory - a capability for which CCD/MOS is ideally suited. Hughes Research Laboratories, as part of our own internal research program, has designed and demonstrated all the standard logic functions using CCD technology. We have also demonstrated high-speed, low-power binary addition using a novel CCD ripple adder technique which should allow us to perform the necessary full addition in a single clock cycle. We are currently investigating the optimum structured design for the LSI/VLSI circuits, which will enable us to demonstrate the concept in the next phase of the Image Understanding Program.

References

1. O.D. Faugeras and W.K. Pratt, "Stochastic-Based Visual Texture Feature Extraction," USC IPI Memorandum, January 1979.

2. G.R. Nudd, P.A. Nygaard, C.L. Jiang, "Application of Charge-Coupled Device Technology to Two-Dimensional Image Processing," presented at the 1978 International Charge-Coupled Devices Symposium, San Diego, October 1978.

3. C. Seguin and M.F. Tompett, Charge Transfer Devices, Advances in Electronics and Electron Physics, No. 8, Academic Press, 1975.

4. W.K. Pratt, Digital Image Processing, Wiley & Sons, New York, 1978.

5. M.F. Tompsett and E.J. Zimay, "Use of Charge Coupled Devices for Delaying Analog Signals," IEEE Journal of Solid State Circuits, SC-8.

6. P.R. Prince, T.J. Mahoney, J.A. Sekula, D.G. Maeding, N. Cuk, "Displacement Charge Subtraction CCD Transversal Filters," Proc. Int. Conf. on the Application of CCD, San Diego, October 1978.

7. T.A. Zimmerman, R.A. Allen and R.W. Jacobs, "Digital

Charge-Coupled Logics (DCCL)," IEEE J. Solid State Circuits, Vol. SC-12, No. 5, October 1977.

an integral part of the processor would consist of serial high speed memory - a capability for which CCD/MOS is ideally suited. Hughes Research Laboratories, as part of our own internal research program, has designed and demonstrated all the standard logic functions using CCD technology. We have also demonstrated high-speed, low-power binary addition using a novel CCD ripple adder technique which should allow us to perform the necessary full addition in a single clock cycle. We are currently investigating the optimum structured design for the LSI/VLSI circuits, which will enable us to demonstrate the concept in the next phase of the Image Understanding Program.

References

1. O.D. Faugeras and W.K. Pratt, "Stochastic-Based Visual Texture Feature Extraction," USC IPI Memorandum, January 1979.

2. G.R. Nudd, P.A. Nygaard, C.L. Jiang, "Application of Charge-Coupled Device Technology to Two-Dimensional Image Processing," presented at the 1978 International Charge-Coupled Devices Symposium, San Diego, October 1978.

3. C. Seguin and M.F. Tompett, Charge Transfer Devices, Advances in Electronics and Electron Physics, No. 8, Academic Press, 1975.

4. W.K. Pratt, Digital Image Processing, Wiley & Sons, New York, 1978.

5. M.F. Tompsett and E.J. Zimay, "Use of Charge Coupled Devices for Delaying Analog Signals," IEEE Journal of Solid State Circuits, SC-8.

6. P.R. Prince, T.J. Mahoney, J.A. Sekula, D.G. Maeding, N. Cuk, "Displacement Charge Subtraction CCD Transversal Filters," Proc. Int. Conf. on the Application of CCD, San Diego, October 1978.

7. T.A. Zimmerman, R.A. Allen and R.W. Jacobs, "Digital

APPENDIX A

Papers Presented this Period

APPLICATION OF CHARGE-COUPLED DEVICE TECHNOLOGY TO TWO-DIMENSIONAL IMAGE PROCESSING*

G.R. Nudd
Hughes Research
Laboratories
Malibu, CA 90265

P.A. Nygaard
Industrial Products
Division
Carlsbad, CA 92008

C.L. Jiang
Hughes Aircraft
Company
Culver City, CA 90230

Abstract

This paper describes the application of charge-coupled device (CCD) technology to two-dimensional image processing. The processing operations discussed are widely used as preprocessing functions for more complex image understanding techniques. Algorithms such as edge detection, local averaging, and unsharp masking[1,2] have been implemented directly in the charge domain using extensions of the analog transversal filtering techniques previously demonstrated. The design concepts and circuit layouts are discussed together with the performance data on test imagery and "real-time" video.

1. Introduction

In the past several years, there has been a significantly increased interest in image enhancement and image understanding both for commercial systems (such as industrial inspection) and for military sensors. The processing algorithms and techniques developed have generally been implemented on general-purpose digital computers, and, in general, the processing times required to perform even relatively simple operations, such as local edge detection, have limited their use to non-real-time applications. The Sobel edge-detection scheme[3] described here, for example, requires approximately 5 x 10^6 operations per frame and might take 5 to 10 sec on a PDP-10 machine. This is two to three orders of magnitude slower than is required for "real-time" video ($\approx$ 7.5 MHz).

Since low-level or preprocessing operations typically require the greatest computation time, one would generally want to use the preprocessor to dramatically reduce the data rate. This would allow the higher level operations (such as the so-called syntactic operations) to be performed at much lower throughput. The aim of this work is to investigate the feasibility of performing several commonly used preprocessing operations in CCD circuitry and thereby to increase the processing speed to allow real-time operation. CCDs were choosen both because they have inherently low power-delay products (which allow very high circuit densities) and because many modern sensors are themselves CCDs. In this way, the preprocessing functions might be incorporated directly into the sensor as options. This is the basis of the so-called "smart sensor" philosophy. The functions described here are edge detection, local averaging, adaptive stretch, binarization, and unsharp masking. The formulations of each of these algorithms is given in Section 2. Where appropriate, we have tried to structure the processing in the form of analog transversal filters to achieve optimal-speed and circuit density. This has required the development of two-dimensional filtering operations and novel circuit techniques to perform operations, such as absolute value determination, directly in the charge

*This work was supported in part by a subcontract from the Image Processing Institute of the University of Southern California under Contract No. F33615-76-C-1203 from the Defense Advanced Research Projects Agency.

APPLICATION OF CHARGE-COUPLED DEVICE TECHNOLOGY TO TWO-DIMENSIONAL IMAGE PROCESSING*

G.R. Nudd
Hughes Research Laboratories
Malibu, CA 90265

P.A. Nygaard
Industrial Products Division
Carlsbad, CA 92008

C.L. Jiang
Hughes Aircraft Company
Culver City, CA 90230

Abstract

This paper describes the application of charge-coupled device (CCD) technology to two-dimensional image processing. The processing operations discussed are widely used as preprocessing functions for more complex image understanding techniques. Algorithms such as edge detection, local averaging, and unsharp masking[1,2] have been implemented directly in the charge domain using extensions of the analog transversal filtering techniques previously demonstrated. The design concepts and circuit layouts are discussed together with the performance data on test imagery and "real-time" video.

1. Introduction

In the past several years, there has been a significantly increased interest in image enhancement and image understanding both for commercial systems (such as industrial inspection) and for military sensors. The processing algorithms and techniques developed have generally been implemented on general-purpose digital computers, and, in general, the processing times required to perform even relatively simple operations, such as local edge detection, have limited their use to non-real-time applications. The Sobel edge-detection scheme[3] described here, for example, requires approximately 5 x 10^6 operations per frame and might take 5 to 10 sec on a PDP-10 machine. This is two to three orders of magnitude slower than is required for "real-time" video (≈ 7.5 MHz).

Since low-level or preprocessing operations typically require the greatest computation time, one would generally want to use the preprocessor to dramatically reduce the data rate. This would allow the higher level operations (such as the so-called syntactic operations) to be performed at much lower throughput. The aim of this work is to investigate the feasibility of performing several commonly used preprocessing operations in CCD circuitry and thereby to increase the processing speed to allow real-time operation. CCDs were choosen both because they have inherently low power-delay products (which allow very high circuit densities) and because many modern sensors are themselves CCDs. In this way, the preprocessing functions might be incorporated directly into the sensor as options. This is the basis of the so-called "smart sensor" philosophy. The functions described here are edge detection, local averaging, adaptive stretch, binarization, and unsharp masking. The formulations of each of these algorithms is given in Section 2. Where appropriate, we have tried to structure the processing in the form of analog transversal filters to achieve optimal-speed and circuit density. This has required the development of two-dimensional filtering operations and novel circuit techniques to perform operations, such as absolute value determination, directly in the charge

*This work was supported in part by a subcontract from the Image Processing Institute of the University of Southern California under Contract No. F33615-76-C-1203 from the Defense Advanced Research Projects Agency.

domain. This should provide optimal dynamic range and linearity.

2. Definition of Processing Algorithms

Five preprocessing operations have been implemented. Each is based on a kernel of 3 x 3 pixels, shown in Figure 1. The first test circuit is a CCD implementation of the Sobel edge-detection algorithm.[3] This circuit was chosen because it demonstrates two operations important to image processing: (1) the possibility of achieving a two-dimensional convolution with arbitrary weightings and (2) the ability to perform nonlinear functions such as the absolute magnitude operation.

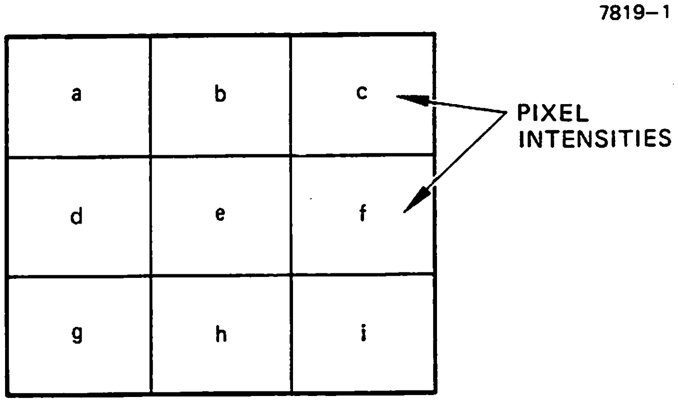


Figure 1. Schematic of the basic 3 x 3 kernel.

The Sobel algorithm operates on the full array and evaluates

$$S(e) = 1/8\ [|(a + 2b + c) - (g + 2h + i)| + |(a + 2d + g) - (c + 2f + i)|] \qquad (1)$$

for each picture element. This output is a measure of the edge components passing through the kernel and is independent of both the polarity of the edge and, to a large extent, its orientation. The other operations are

Local averaging:

$$f_m(e) = 1/9\ [a + b + c + d + e + f + g + h + i] \qquad (2)$$

Unsharp masking:

$$S_u(e) = (1 - \alpha)e - \alpha\ f_m(e) \qquad (3)$$

Adaptive binarization:

$$S_b(e) = \begin{cases} 1 \text{ for } f_m(e) \leq e \\ 0 \text{ for } f_m(e) > e \end{cases} \qquad (4)$$

Adaptive stretch:

$$S_a(e) = \begin{cases} 2 \text{ Min } \{e, r/2\} & \text{for } f_m(e) \leq r/2 \\ 2 \text{ Max } \{(e, r/2), 0\} & \text{for } f_m(e) > r/2, \end{cases} \qquad (5)$$

where r is the maximum pixel intensity.

All of the above operations can be obtained by combinations of three basic functions: the local means $f_m(e)$, the edges S(e), and the center pixel intensity. Each of these functions can be obtained directly from the CCD analog transversal operations described below.

3. Device Descriptions

The CCD implementation of two-dimensional edge detection and local mean algorithm is an important aspect of many real-time image-processing applications. Further, since the functions discussed in Section 2 can be derived from combinations of center pixel intensities, local means, and edges, only the CCD edge detection and local mean circuits will be described in detail.

The CCD Sobel circuit consists of a 3 x 3 two-dimensional transversal filter, an absolute value operator, and a summing circuit. Figure 2 is a functional block diagram for the circuit. Three lines of analog video signal are fed into the 3 x 3 CCD Sobel transversal filter. Two differential outputs are obtained and amplified before taking their absolute values and summing. The final output |a+2b+c - (g+2h+i)| + |a+2d+g - (c+2f+i)| provides edge information about the image (as is shown in Section 4). Other input and output points are also available for individual circuit tests, as indicated in Figure 2.

The CCD Sobel circuit has three parallel signal channels for the three analog video lines of the image. The inputs are of the Tompsett fill-and-spill type. The two-dimensional processing results from the appropriate inter-connection of the eight

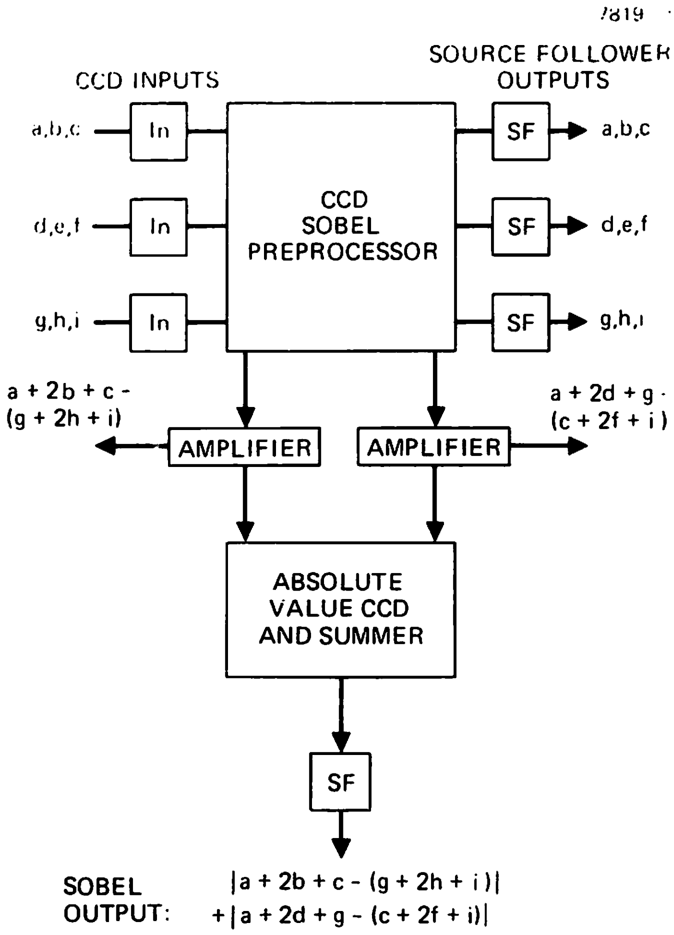

7819
CCD INPUTS
SOURCE FOLLOWER OUTPUTS
a,b,c
d,e,f
g,h,i
In
In
In
CCD SOBEL PREPROCESSOR
SF
SF
SF
a,b,c
d,e,f
g,h,i
a + 2b + c - (g + 2h + i)
a + 2d + g - (c + 2f + i)
AMPLIFIER
AMPLIFIER
ABSOLUTE VALUE CCD AND SUMMER
SF
SOBEL OUTPUT:
|a + 2b + c - (g + 2h + i)| + |a + 2d + g - (c + 2f + i)|

Figure 2. Block diagram for CCD Sobel operator.

floating gates of the three-channel split-electrode transversal filter. Figure 3 shows that the output in each of the four bus lines is proportional to the charges under the connected gates (a, b, c, etc.). The necessary weightings (1, 2, 1, etc.) are achieved by varing the floating gate area. The differences between the weighted sums are obtained through the output differential amplifiers. Each output, therefore, represents an orthogonal edge component. These components then act as inputs to the gates of the CCD absolute value circuit shown in Figure 4 to achieve the two-dimensional Sobel output.

The CCD absolute value circuit operates using a novel technique that allows a charge storage that is equivalent only to the input signal magnitude and is independent of signal polarity. During the input phase, ϕ_{INA} is pulsed low first (high surface electron potential in an n-channel CCD) and then settles high (low surface electron potential). When the signal voltage V_{SIG} is less than the reference voltage V_{REF} set by the REF gate, the electrons will fill the potential well under the gates B2 and FZ, as shown in Figure 4(a). During the output phase, ϕ_{OUTA} is pulsed high and the charge packet is transferred to the summing output. This charge is proportional to

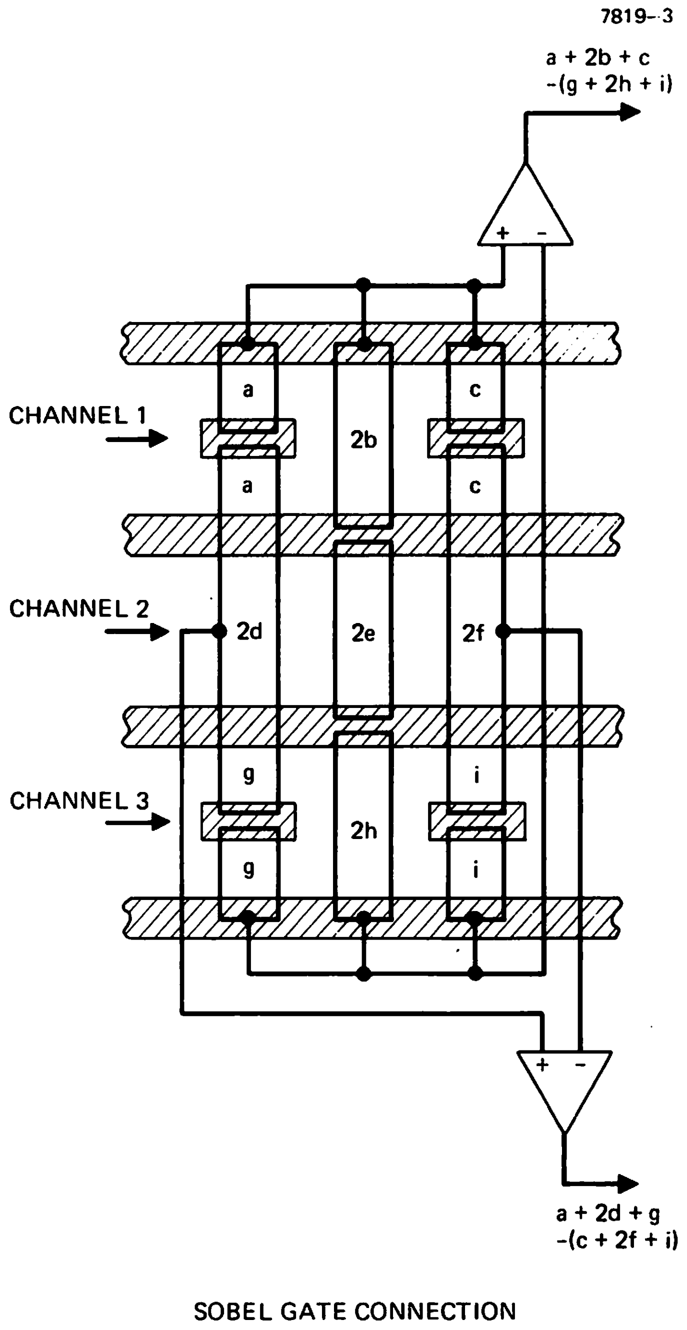


Figure 3. CCD Sobel preprocessor.

$$\left\{ \left(V_{FZ} - V_{REF}\right) \left(A_{FZ} + A_{B2}\right) + \left(V_{REF} - V_{SIG}\right) \left(A_{FZ} + A_{B2}\right) \right\} ,$$

domain. This should provide optimal dynamic range and linearity.

2. Definition of Processing Algorithms

Five preprocessing operations have been implemented. Each is based on a kernel of 3 x 3 pixels, shown in Figure 1. The first test circuit is a CCD implementation of the Sobel edge-detection algorithm.[3] This circuit was chosen because it demonstrates two operations important to image processing: (1) the possibility of achieving a two-dimensional convolution with arbitrary weightings and (2) the ability to perform nonlinear functions such as the absolute magnitude operation.

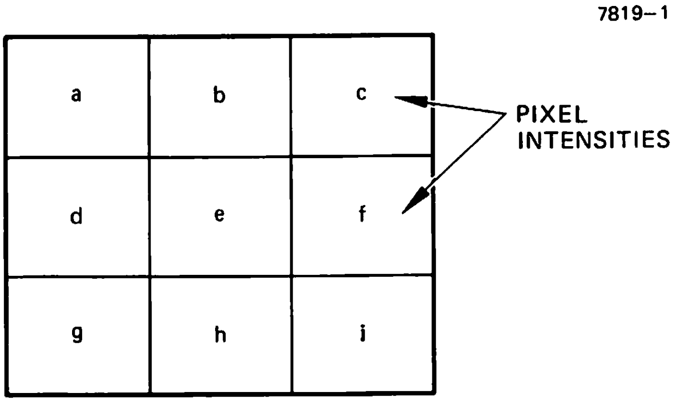


Figure 1. Schematic of the basic 3 x 3 kernel.

The Sobel algorithm operates on the full array and evaluates

$$S(e) = 1/8\ [|(a + 2b + c) - (g + 2h + i)| + |(a + 2d + g) - (c + 2f + i)|] \qquad (1)$$

for each picture element. This output is a measure of the edge components passing through the kernel and is independent of both the polarity of the edge and, to a large extent, its orientation. The other operations are

Local averaging:

$$f_m(e) = 1/9\ [a + b + c + d + e + f + g + h + i] \qquad (2)$$

Unsharp masking:

$$S_u(e) = (1 - \alpha)e - \alpha\ f_m(e) \qquad (3)$$

Adaptive binarization:

$$S_b(e) = \begin{cases} 1 \text{ for } f_m(e) \leq e \\ 0 \text{ for } f_m(e) > e \end{cases} \qquad (4)$$

Adaptive stretch:

$$S_a(e) = \begin{cases} 2 \text{ Min } \{e, r/2\} & \text{for } f_m(e) \leq r/2 \\ 2 \text{ Max } \{(e, r/2), 0\} & \text{for } f_m(e) > r/2, \end{cases} \qquad (5)$$

where r is the maximum pixel intensity.

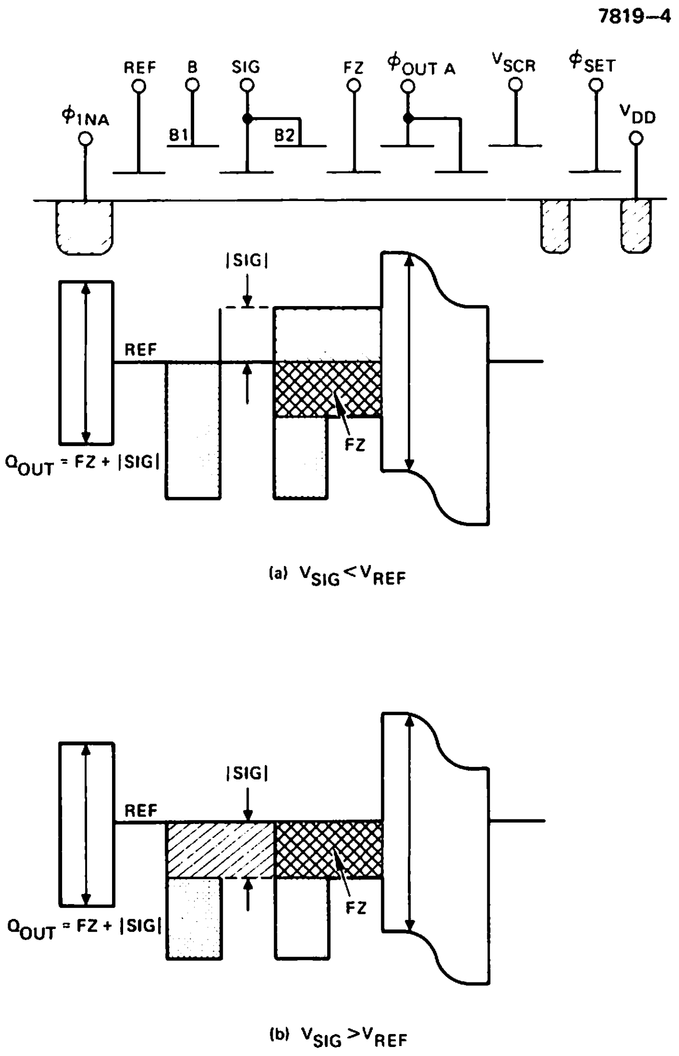
All of the above operations can be obtained by combinations of three basic functions: the local means $f_m(e)$, the edges S(e), and the center pixel intensity. Each of these functions can be obtained directly from the CCD analog transversal operations described below.

3. Device Descriptions

The CCD implementation of two-dimensional edge detection and local mean algorithm is an important aspect of many real-time image-processing applications. Further, since the functions discussed in Section 2 can be derived from combinations of center pixel intensities, local means, and edges, only the CCD edge detection and local mean circuits will be described in detail.

The CCD Sobel circuit consists of a 3 x 3 two-dimensional transversal filter, an absolute value operator, and a summing circuit. Figure 2 is a functional block diagram for the circuit. Three lines of analog video signal are fed into the 3 x 3 CCD Sobel transversal filter. Two differential outputs are obtained and amplified before taking their absolute values and summing. The final output $|a+2b+c - (g+2h+i)| + |a+2d+g - (c+2f+i)|$ provides edge information about the image (as is shown in Section 4). Other input and output points are also available for individual circuit tests, as indicated in Figure 2.

The CCD Sobel circuit has three parallel signal channels for the three analog video lines of the image. The inputs are of the Tompsett fill-and-spill type. The two-dimensional processing results from the appropriate inter-connection of the eight



(a) $V_{SIG} < V_{REF}$

(b) $V_{SIG} > V_{REF}$

Figure 4. CCD absolute value circuit.

where A_{FZ} is the rate of the gate FZ, etc. The first term corresponds to the fat zero charge and the second to the signal charge referred to the reference level. However, if V_{SIG} is higher than V_{REF}, the potential well under B1, SIG, B2, and FZ will be filled, as shown in Figure 4(b). The output charge is proportional to

$$\left\{\left(V_{FZ} - V_{REF}\right)\left(A_{FZ} + A_{B2}\right) + \left(V_{SIG} - V_{REF}\right)\left(A_{SIG} + A_{B1}\right)\right\} .$$

If the gate areas are fabricated such that $A_{SIG} + A_{N1} = A_{FZ} + A_{B2}$, then the output charge will always be a fat zero plus the charge proportional to the magnitude of the signal with V_{REF} as reference point. That is, a charge output corresponding to the absolute value of the input signal is obtained. After the absolute values of the differences are obtained, they are summed in the charge domain and the Sobel operation is completed. The CCD local mean circuit shown in Figure 5 consists of 3 x 3 cells with nine floating gates connected together to yield an output proportional to a+b+c+d+e+f+g+h+i.

The gate interconnect of the 3 x 3 CCD two-dimensional filtering circuit has to be laid out carefully to minimize the stray capacitance and to balance the positive and negative input to the differential amplifiers. In thc CCD absolute value circuit, speed and accuracy are, in the case of $V_{SIG} > V_{REF}$, limited by the transfer inefficiency.

The CCDs are N channel and are fabricated with a two-layer polysilicon process. This process requires nine masks and two ion implantations. The CCDs have a bit

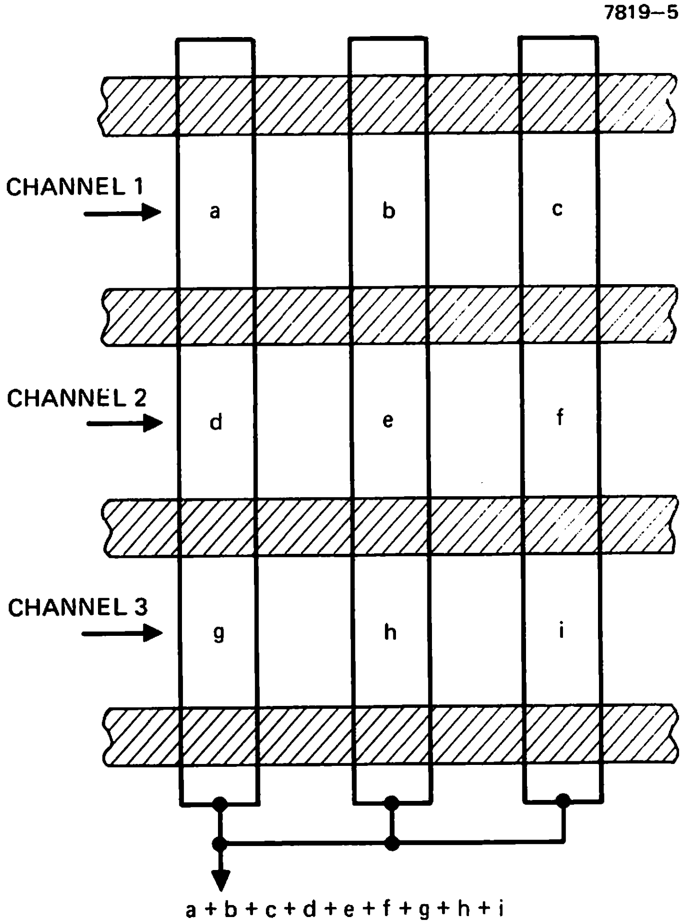


Figure 5. CCD local mean circuit.

length of 27 μm, and the minimum feature size is 2.5 μm. This results in a total area of 0.7 mm^2 for the Sobel (see Figure 6(a)), of which 0.15 mm^2 is the transversal filter. This compares with a total area of 0.6 mm^2 for the mean filter (Figure 6(b)). To achieve the necessary capacitance balance between the two difference outputs, additional metal was added, as Figure 6 shows.

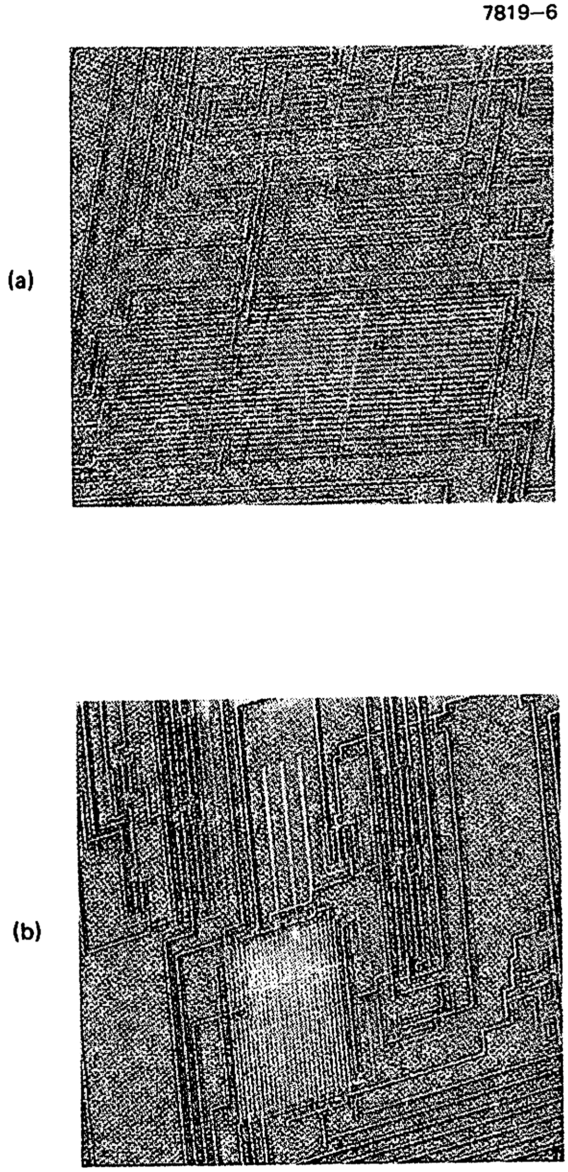


Figure 6. Photomicrographs of (a) the edge detection circuit and (b) the local mean filter.

4. Test Results

a. Measurement of Electrical Characteristics.

The two basic functions of the CCD circuits, arithmetic operations (such as absolute magnitude determination and summation) and transversal filtering, have been tested independently and the transfer characteristics measured. The weighting functions of the transversal filters for the Sobel edge detection and local mean evaluation, for example, can be written as:

$$S_x = 1/8 \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

$$S_y = 1/8 \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$$

$$W_m = 1/9 \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

where S_x and S_y provide the x and y components of the edge values, and W_m provides the mean. Both the impulse response and the linearity of these operations have been determined using the microcomputer-based test set-up shown in Figure 7.

Here the microcomputer is used to provide flexible and programmed data inputs to the CCD circuits. These data are then clocked through the devices, and the output is stored in the computer memory. This provides an accurate and rapid means of characterizing the device performance as a function of the various input parameters. The speed and accuracy of this system are basically determined by the computer cycle time and the analog-to-digital converters. The machine described here has a basic cycle time of ≈ 2 μsec and can provide an 8-bit quantization, resulting in a maximum CCD clock speed of ≈ 30 kHz.

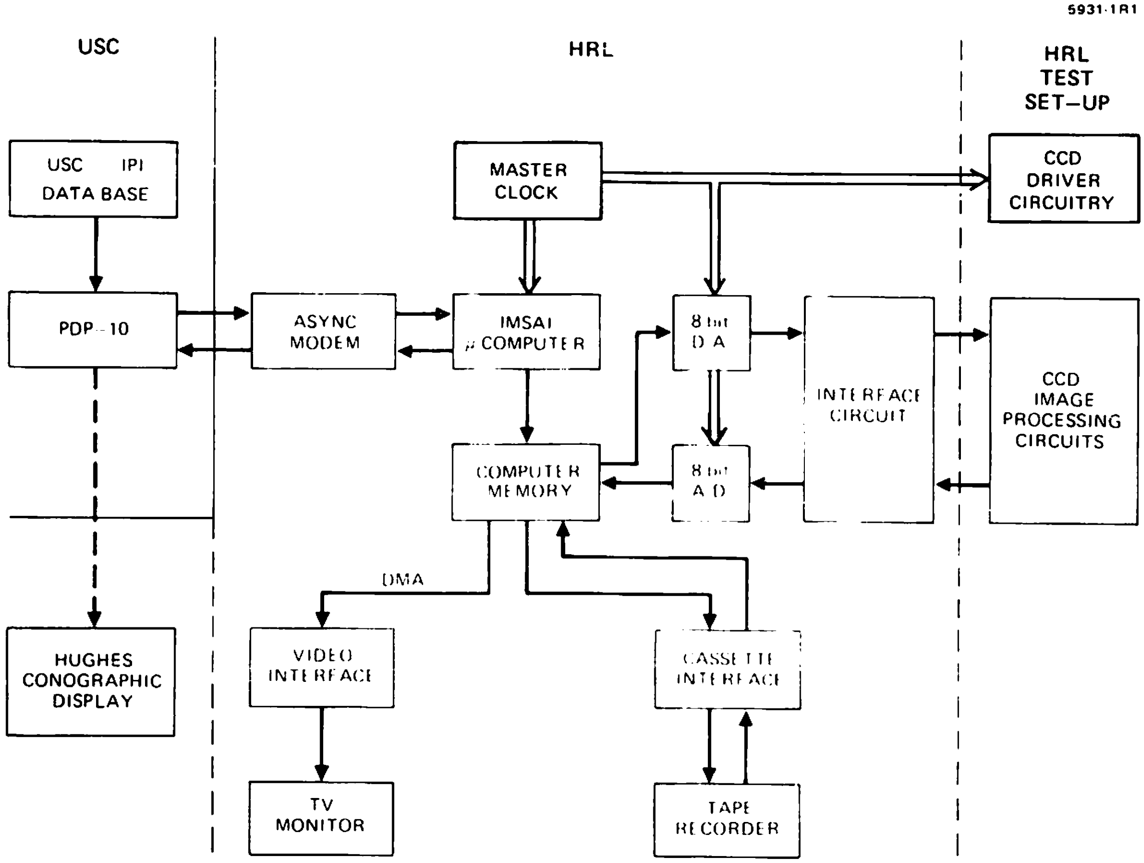


Figure 7. Microcomputer-based test facilities.

When a single input pulse with a duration of less than one-half clock cycle is used as the input, the output is equivalent to the impulse response of each component of the filters. Examples of this for the Sobel operation are shown in Figure 8.

An additional benefit of this test set-up is that a unique pattern of either analog or digital data can be generated and used as the input to the CCD circuit and the output data gated so as to uniquely determine the operation of any tap within the array. For example, if an input that linearly increases with time is clocked into the array and the output is gated so as to measure only the nth output pulse in each cycle, the weighting W_n of the nth floating electrode in the array can be uniquely determined. Measurements made in this way are shown in Figure 9, which shows the output voltage directly as a function of the input for each of the nine floating gates in the Sobel filter. The slope of each input/output characteristic gives the tap weighting for each tap. From this, inputs can be shown to be linear over approximately a 3-V range. This translates to an accuracy and dynamic range equivalent to approximately 16 gray levels.

The absolute value circuit described in Section 3 was tested using a similar technique; the results are shown in Figure 10. Here the input voltage on the gate SIG has been swept over a range of 0 V to 10 V, and the characteristic can be explained with reference to Figure 4. Initially, as the signal voltage is increased, charge flows over the input gate and is stored under gates FZ and B1. This charge is then clocked out as the clock phase changes. However, as the input voltage is increased beyond V_{in1} (Figure 9), the bucket size decreases linearly, resulting in the linear change in voltage out (AB). When the input voltage reaches V_{REF}, the bucket size is a minimum equivalent only to the fat zero. Increasing the input further causes some of the charge previously trapped under B1 to be clocked out. Thus, the output characteristic again changes linearly from B to C. Hence, when the input signal is operated about V_{REF}, the output changes linearly in proportion to $|V_{SIG} - V_{REF}|$, the output polarity being

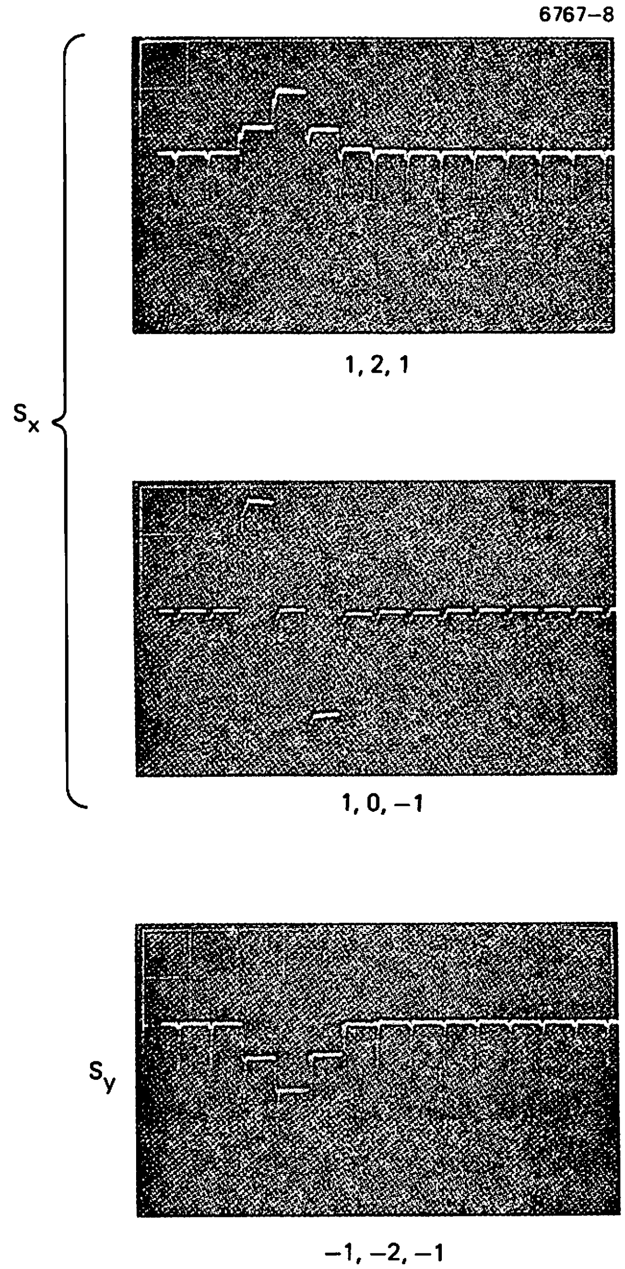


Figure 8. Measured impulse response of CCD Sobel filter.

independent of V_{SIG}. The input voltage swing, as shown in Figure 10, is approximately ± 2 V, resulting in an output change of some 400 mV. This is again equivalent to an accuracy of approximately 4 bits.

b. Performance Evaluation of the Processor.

The processor has been tested on true two-dimensional imagery using both a stored data base and a real-time input from a commercial vidicon. The use of a stored data base allows most of the problems associated with

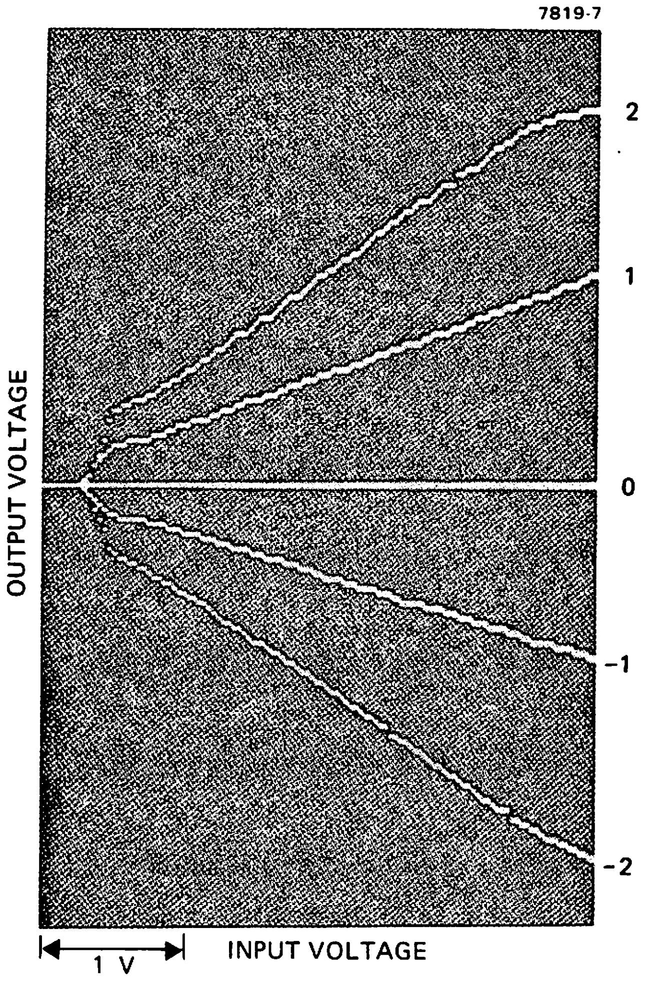


Figure 9. Measurement of the weighting functions for Sobel operators.

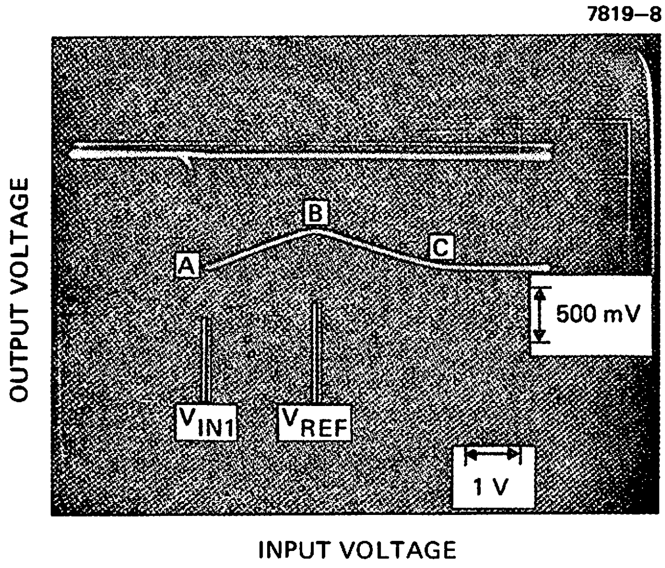


Figure 10. Transfer characteristic of absolute value circuit.

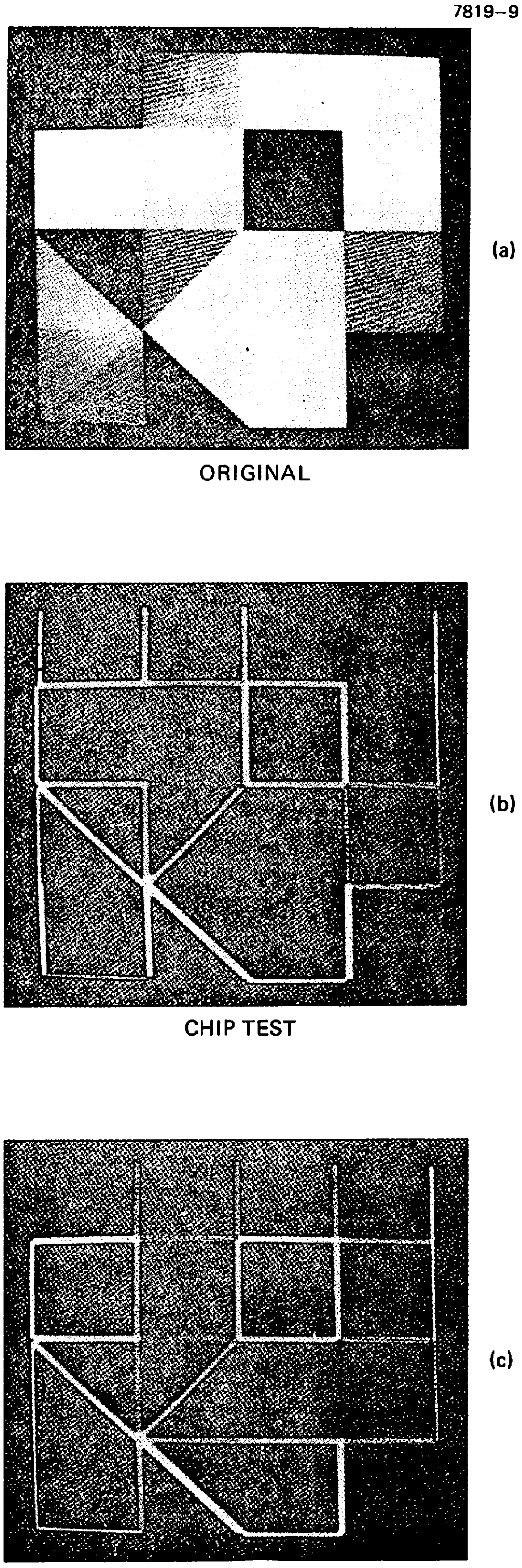
the sensor, such as illumination, resolution, and signal-to-noise ratio, to be separated from the evaluation of processor performance. The maximum data rate of this system, however, is limited to about 30 kHz. In this mode, the imagery to be processed is first digitized and stored in the computer memory, as shown in Figure 7. (In practice, a very large data base is available on magnetic tape and has been used extensively in the performance evaluation.) The stored data are then clocked out of the random access memory in synchronism with the CCD clocks and converted to analog data before entering the processor. The processed data from the CCD are converted again to digital format and stored in the computer memory in the form of 128 x 128 four bit words. Direct memory address is then used to refresh a standard video monitor.

An example of this is shown in Figure 11 for a two-dimensional test pattern. A comparison of the output (Figure 11(b)) with the computer simulation (Figure 11(c)) shows that an accuracy of approximately four bits is preserved. An example of its operation on a real image is shown in Figure 12.

In addition to the tests on stored data, we have interfaced the processor directly with a commercial vidicon camera. The standard operating frequency of this "real-time" video is ≈ 7 MHz, providing 525 x 525 picture elements at 30 frames/sec. At present, we have operated our CCD processor at a maximum clock rate of 4 MHz, which provides the full 525 vertical resolution elements but about a three-to-one resolution loss in the horizontal direction. An example of the performance in both the local-averaging and the edge-detection modes is shown in Figure 13. Two other functions, unsharp masking and binarization (both of which are performed in real-time by our CCD processor), are also shown.

5. Conclusions

The concepts and design details of a CCD image processor that performs two-dimensional linear and nonlinear operations are discussed. Our results indicate that it is feasibile to use a CCD integrated circuit approach for the image preprocessor. The operations described are used as the basis for higher-level syntactic type of image processing, which is becoming



ORIGINAL

CHIP TEST

COMPUTER TEST

Figure 11. Example of processor operation on stored test data (at 30 kHz).

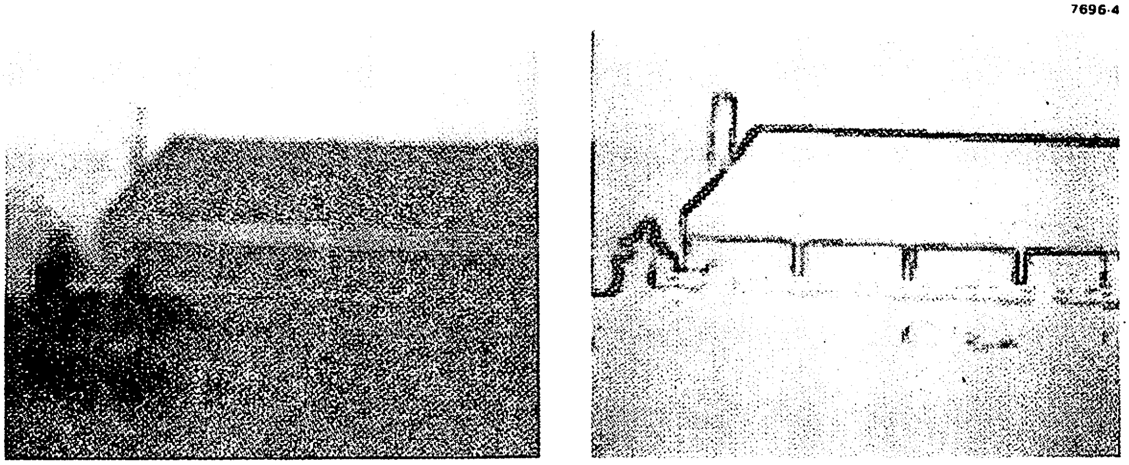
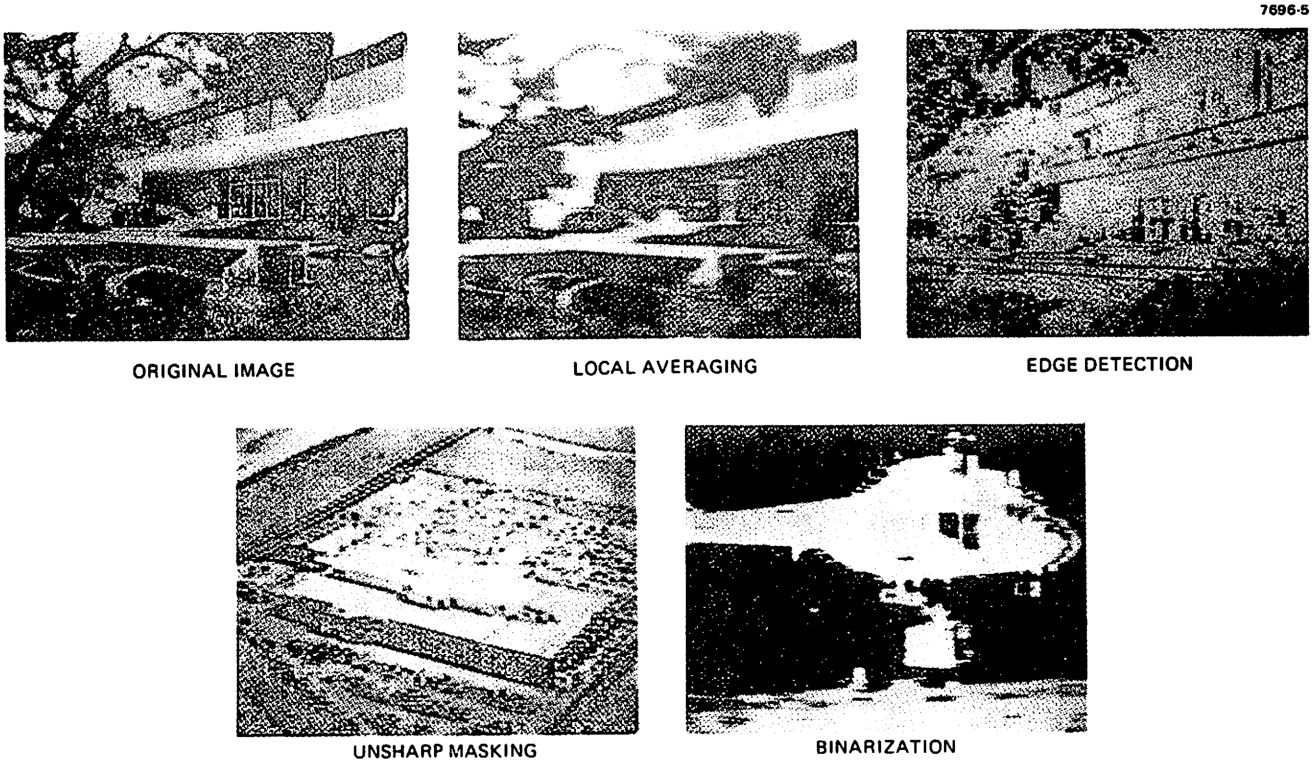


Figure 12. Example of processor performance operating on stored imagery.



ORIGINAL IMAGE

LOCAL AVERAGING

EDGE DETECTION

UNSHARP MASKING

BINARIZATION

Figure 13. Performance of processor at real-time data rates.

increasingly important in military systems for target acquisition and tracking. Typically, however, most of the processing time is taken up in the "preprocessing" type operations, and our present indications are that the CCD techniques are able to operate with at least 4-bit accuracy at speeds 100 to 300 times faster than the conventional general-purpose processor used to date.

References

[1] Pratt, W.K., Digital Image Processing, John Wiley & Sons, New York, 1978.

[2] Andrews, H.C., and Hunt, B.R., Digital Image Restoration, Prentice-Hall, Englewood Cliffs, N.J., 1977.

[3] Duda, R.O., and Hart, P.E., Pattern Classification and Scene Analysis, John Wiley & Sons, New York, 1973.

Acknowledgement

We wish to thank P.R. Prince and K. Nummedal for their help in this effort.

A CHARGE-COUPLED DEVICE IMAGE PROCESSOR FOR SMART SENSOR APPLICATIONS*

G. R. Nudd, P. A. Nygaard,† G. D. Thurmond, and S. D. Fouse
Hughes Research Laboratories
Malibu, California 90265

Abstract

This paper describes the development and performance demonstration of two charge-coupled device (CCD) chips for image processing. The aim of the work is to demonstrate the feasibility of developing custom CCD architectures that will enable the time-consuming, low-level processing functions (such as feature extraction) to be performed in real time. We describe the circuit concepts and device layout for six commonly used algorithms and include photographs of the raw and processed imagery.

1. Introduction

The successful application of many of the currently discussed image analysis techniques will depend ultimately on the speed and accuracy with which they can be implemented. Many of the more sophisticated algorithms developed to date have been demonstrated only on general-purpose computers, and the resultant processing rates, even for resolutions equivalent to standard quality television, are several orders of magnitude too low for "real time" display. Simple preprocessing functions, such as edge detection and local averaging,(1,2) might take from several seconds to minutes per frame. For example, the edge-detection technique described below has been implemented in machine language on an 8080A microprocessor and takes about 100 sec/frame. More sophisticated algorithms, which might require the adaptive combination of several functions (e.g., edges, means) with the center picture element, will require correspondingly larger processing times. In general, it is desirable to operate at speeds equivalent to commercial television, with approximately 525 x 525 pixels at 30 frames/sec. This not only provides the spatial and temporal resolution required by most systems but enables standard television hardware to be used at the interfaces.

Some development work in which commercially available circuits were used to perform specific image preprocessing functions has been reported.(3) In general, these approaches result in a large number of integrated circuit (IC) packages and are not well suited to "smart sensor" applications. Until relatively recently, the computational complexity of the algorithms has precluded the use of a single IC to process the data. However, the rapid progress in technologies such as charge transfer devices and metal oxide semiconductors (MOS) with inherently low power-delay products has resulted in a very significant increase in the available throughput.

Charge-coupled devices (CCDs) are of particular significance to image processing because they can be used in both image detection and processing. Further, they can be configured to provide an especially simple and direct means of performing two-dimensional convolutions, which form the basis of much low-level image processing. The low power requirements, typically two orders of magnitude lower than the conventional bipolar circuits used in mainframe computers, and the small device geometries mean that highly parallel approaches can be used to achieve full-frame processing. One such concept is shown in Figure 1. Here a CCD imager or analog store is used to store a full frame, and the data from the N rows are clocked out in parallel into N parallel processing circuits. Each circuit might perform the Sobel(4) operator, for example, and process the data for an entire line, with the processed output appearing at the clock rate f_c. Thus, an entire frame would be processed in Nf_c sec. For a 525 x 525 frame, this could be about 50 μsec. Further, by performing the processing directly in the charge domain, thus avoiding the charge-sensing amplifiers necessary to drive conventional discretes, greater accuracy and linearity and larger dynamic range can be obtained, both of which are crucial for high sensitivity.

* This work was supported in part by a subcontract from the Image Processing Institute of the University of Southern California under Contract No. F33615-76-C-1203 from the Defense Advanced Research Projects Agency, and Contract No. DAAK70-77-C-0216 from the Night Vision Laboratories, Ft. Belvoir, Virginia.

† Mr. Nygaard is currently with the Carlsbad Research Center, Hughes Aircraft Company, Carlsbad, California.

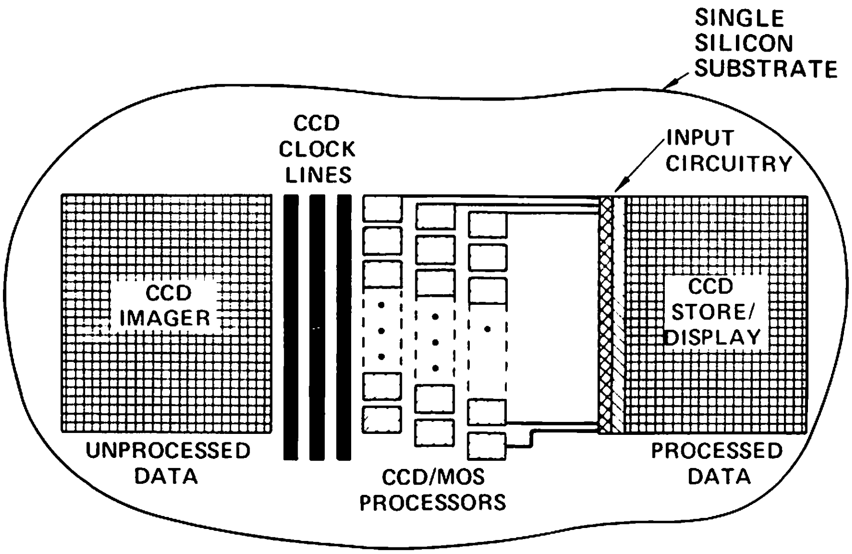


Figure 1. Parallel processing concept.

This paper describes the development of a two-phase, n-channel CCD processor that can perform seven basic preprocessing algorithms. The devices are fabricated on single 200 mil x 200 mil chips and have been operated at 2 MHz with an accuracy of 4-bits. Examples of their performance when interfaced to a stored data base via a 8080A microprocessor and when operated directly from a commerical vidicon camera are included.

2. Algorithm Definition

In principle, the CCD processing techniques developed here can be applied to any size kernel. The optimum choice, which will depend on the overall system requirements, will in general represent a trade-off among noise immunity, resolution, and dynamic range. We have arbitrarily selected a 3 x 3 array, as shown in Figure 2.

f (x – 1, y + 1)	f (x, y + 1)	f (x + 1, y + 1)
f (x – 1, y)	f (x, y)	f (x + 1, y)
f (x – 1, y – 1)	f (x, y – 1)	f (x + 1, y - 1)

3 x 3 ARRAY OF PIXELS

Figure 2. Kernel used for processing algorithms.

The aim of this work was to implement the algorithms on a single silicon chip and demonstrate the use of the 3 x 3 kernel as a control signal for linear and nonlinear as well as spatially invariant and variant (or adaptive) signal-processing functions in two dimensions.

Many algorithms have been developed for these tasks, and the nonlinear techniques probably have been the most successful. However, all algorithms could be envisioned as being designed around three control signals or two-dimensional functions. These three signals are

$$\begin{aligned} \text{Original image} &= f(x,y) \\ \text{Blurred image} &= f_m(x,y) \\ \text{Sobel of image} &= f_s(x,y) \end{aligned} \qquad (1)$$

The blurred image, representing the local coverage, is obtained by convolving f(x,y) with a 3 x 3 kernel with entries that are all unity. The Sobel image, representing the edge detected output, is obtained by passing the 3 x 3 Sobel operator over the original image. Thus,

A CHARGE-COUPLED DEVICE IMAGE PROCESSOR FOR SMART SENSOR APPLICATIONS*

G. R. Nudd, P. A. Nygaard,† G. D. Thurmond, and S. D. Fouse
Hughes Research Laboratories
Malibu, California 90265

Abstract

This paper describes the development and performance demonstration of two charge-coupled device (CCD) chips for image processing. The aim of the work is to demonstrate the feasibility of developing custom CCD architectures that will enable the time-consuming, low-level processing functions (such as feature extraction) to be performed in real time. We describe the circuit concepts and device layout for six commonly used algorithms and include photographs of the raw and processed imagery.

1. Introduction

The successful application of many of the currently discussed image analysis techniques will depend ultimately on the speed and accuracy with which they can be implemented. Many of the more sophisticated algorithms developed to date have been demonstrated only on general-purpose computers, and the resultant processing rates, even for resolutions equivalent to standard quality television, are several orders of magnitude too low for "real time" display. Simple preprocessing functions, such as edge detection and local averaging,(1,2) might take from several seconds to minutes per frame. For example, the edge-detection technique described below has been implemented in machine language on an 8080A microprocessor and takes about 100 sec/frame. More sophisticated algorithms, which might require the adaptive combination of several functions (e.g., edges, means) with the center picture element, will require correspondingly larger processing times. In general, it is desirable to operate at speeds equivalent to commercial television, with approximately 525 x 525 pixels at 30 frames/sec. This not only provides the spatial and temporal resolution required by most systems but enables standard television hardware to be used at the interfaces.

Some development work in which commercially available circuits were used to perform specific image preprocessing functions has been reported.(3) In general, these approaches result in a large number of integrated circuit (IC) packages and are not well suited to "smart sensor" applications. Until relatively recently, the computational complexity of the algorithms has precluded the use of a single IC to process the data. However, the rapid progress in technologies such as charge transfer devices and metal oxide semiconductors (MOS) with inherently low power-delay products has resulted in a very significant increase in the available throughput.

Charge-coupled devices (CCDs) are of particular significance to image processing because they can be used in both image detection and processing. Further, they can be configured to provide an especially simple and direct means of performing two-dimensional convolutions, which form the basis of much low-level image processing. The low power requirements, typically two orders of magnitude lower than the conventional bipolar circuits used in mainframe computers, and the small device geometries mean that highly parallel approaches can be used to achieve full-frame processing. One such concept is shown in Figure 1. Here a CCD imager or analog store is used to store a full frame, and the data from the N rows are clocked out in parallel into N parallel processing circuits. Each circuit might perform the Sobel(4) operator, for example, and process the data for an entire line, with the processed output appearing at the clock rate f_c. Thus, an entire frame would be processed in Nf_c sec. For a 525 x 525 frame, this could be about 50 μsec. Further, by performing the processing directly in the charge domain, thus avoiding the charge-sensing amplifiers necessary to drive conventional discretes, greater accuracy and linearity and larger dynamic range can be obtained, both of which are crucial for high sensitivity.

* This work was supported in part by a subcontract from the Image Processing Institute of the University of Southern California under Contract No. F33615-76-C-1203 from the Defense Advanced Research Projects Agency, and Contract No. DAAK70-77-C-0216 from the Night Vision Laboratories, Ft. Belvoir, Virginia.

† Mr. Nygaard is currently with the Carlsbad Research Center, Hughes Aircraft Company, Carlsbad, California.

$$f_m(x,y) = f(x,y) \circledast \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \tag{2}$$

$$f_s(x,y) = \left| f(x,y) \circledast \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \right| + \left| f(x,y) \circledast \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \right| \quad . \tag{3}$$

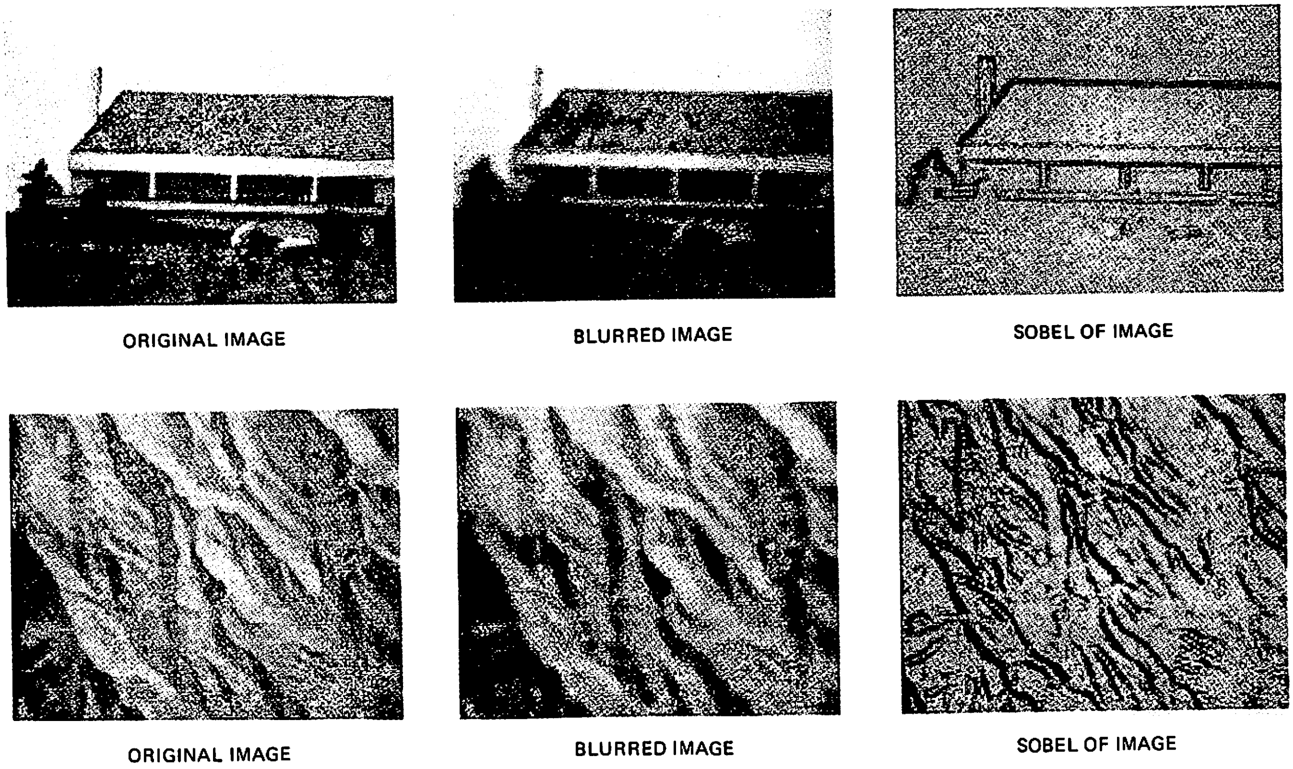
Figure 3 presents these three functions on two images, a "house" and an "aerial reconnaissance" scene. The former has a large dynamic range, while the latter has a small dynamic range.

An example of a simple use of these control signals is the linear combination of the mean and Sobel image:

$$g(x,y) = (1-\lambda)\ f_m(x,y) + \lambda f_s(x,y) \quad , \tag{4}$$

where

$$0 \leq \lambda \leq 1 \quad .$$



ORIGINAL IMAGE

BLURRED IMAGE

SOBEL OF IMAGE

ORIGINAL IMAGE

BLURRED IMAGE

SOBEL OF IMAGE

Figure 3. Computer simulation of the three control signals.

This is shown in Figure 4(a), in which the Sobel edges are emphasized. A more familiar use of these control signals for edge enhancement is in the "unsharp masking" application,(1) shown in Figure 4(b), in which a percentage of the blurred image is simply subtracted from the original:

$$g(x,y) = f(x,y) - \lambda f_m(x,y) \tag{5}$$

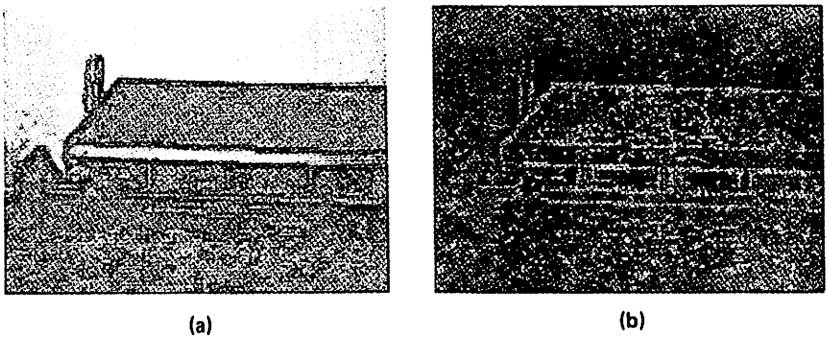


Figure 4. Examples of linear combination of control signals. (a) Mean plus edges. (b) Unsharp masking (original-mean).

The examples shown in Figure 4 are of nonadaptive algorithms, which lack the sophistication needed for the inherently nonstationary nature of imagery. An adaptive stretching algorithm in which the brightness of a gray level is doubled depending on the value of $f_m(x,y)$ was implemented:

$$g(x,y) = 2\begin{cases} \min(f(x,y),\ r) & f_m(x,y) < r \\ \max(f(x,y) - r,\ 0) & f_m(x,y) \geq r \end{cases}, \tag{6}$$

where r = mid-gray. Hence, if the local mean is less than mid-gray, the center pixel is essentially passed through a function memory that has a gain. The objective is to enhance the darks and simultaneously enhance the brights without saturation in either case.

A further adaptive function, one which compares the original picture element with the local mean and provides a binary output according to

$$g(x,Y) = \begin{cases} 1 & f(x,y) \geq f_m(x,y) \\ 0 & f(x,y) < f_m(x,y) \end{cases}, \tag{7}$$

has also been implemented.

Many other adaptive functions can be performed using these basic control functions,and the results of this work will be reported in another paper.

3. CCD Circuit Development

The CCD technology developed in 1970(5) is particularly advantageous for the development of "smart sensors." It is an inherently low-power technology with a power-delay product of $\approx 10^{-2}$ pJ (as compared to 50 pJ for a typical bipolar), which allows high throughput and circuit complexity to be attained. In addition, the basic CCD structure can be used to both sense and process the data. CCD arrays are currently available that provide optical imaging capabilities equivalent to standard television. It is envisioned that, with little added complexity, the type of processing functions discussed here could be incorporated directly at the focal plane providing a range of processed and unprocessed outputs.

The basic CCD structure is illustrated in Figure 5, where a two-phase, n-type surface-channel delay line is shown. The concept involves the shifting of a controlled charge packet at the Si-SiO_2 surface. The stepped gate structure ensures that the charge moves

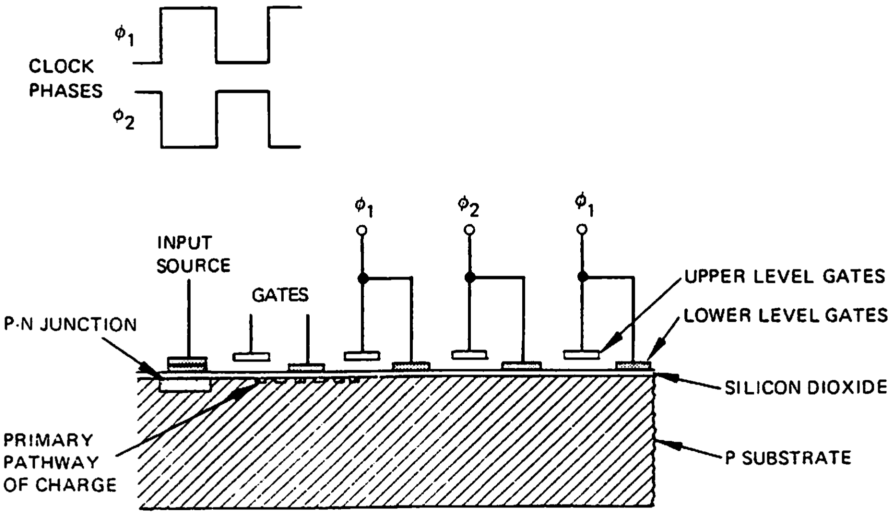


Figure 5. Concept of the CCD analog delay line.

in only one direction in response to the clocking waveforms illustrated. In our application, a charge equivalent to the intensity of each successive picture element is clocked into the device at each cycle. This requires that the clock speed be equivalent to the bandwidth of the analog video signal. The key to most signal-processing applications is a floating gate structure in which a single electrode capacitively couples to the signal charge across the oxide:

$$V_s = Q_s/C_{ox} \quad ,$$

where Q_s is the signal charge, C_{ox} is the oxide capacitance, and V_s is the corresponding output signal. Since the oxide capacitance C_{ox} is equal to $\varepsilon A/t_{ox}$, where A is the gate area, the outputs can be weighted by simply varying the gate length W_i as shown in Figures 6 and 7. The output then becomes $V_o = Q_s(t) * W(*)$, where $*$ is the convolution junction. This is the basis of the CCD analog transversal filter. In our case, since we are operating on two-dimensional data equivalent to the 3 x 3 array, the parallel structure shown in Figure 6 is necessary. Here, three adjacent lines of video data are clocked in synchronism through the structure, and the output is taken from the array of nine identical gates shown. In this case, the output as a function of time t can be written

$$V_m(t) = W \cdot \begin{bmatrix} V_1(t) + V_1(t - T) + V_1(t - 2T) \\ V_2(t) + V_2(t - T) + V_2(t - 2T) \\ V_3(t) + V_3(t - T) + V_3(t - 2T) \end{bmatrix} \quad , \tag{8a}$$

where t is the running time parameter, T is the clock period, and W is the effective gate weighting. In the spatial domain, this is equivalent to

$$f(x,y) \otimes \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad , \tag{8b}$$

which corresponds directly to the blurr or local average function described in Eq. 2.

The equivalent structure for the Sobel edge detection is shown in Figure 7. Two types of operations are involved in the calculation of the Sobel algorithm. The first is the charge sensing and weighting necessary for the detection of the orthogonal edge components:

$$\begin{aligned} S_x &= f(x-1,y+1) + 2f(x,y+1) + f(x+1,y-1) - f(x-1,y-1) + 2f(x,y-1) + f(x+1,y-1) \\ S_y &= f(x-1,y+1) + 2f(x-1,y) + f(x-1,y-1) - f(x+1,y+1) + 2f(x+1,y) + f(x+1,y-1) \; . \end{aligned} \tag{9}$$

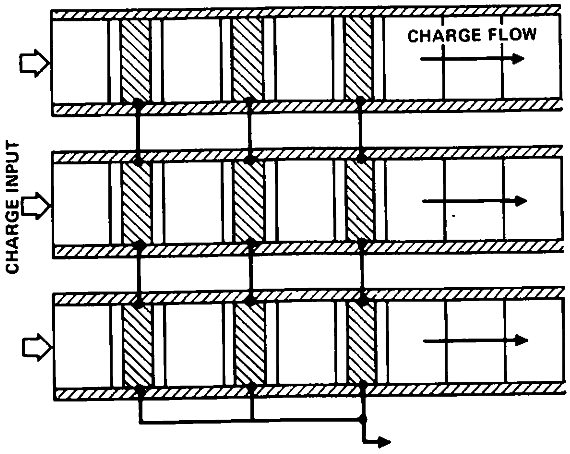


Figure 6. Floating gate array used to obtain blurr $f_m(x,y)$.

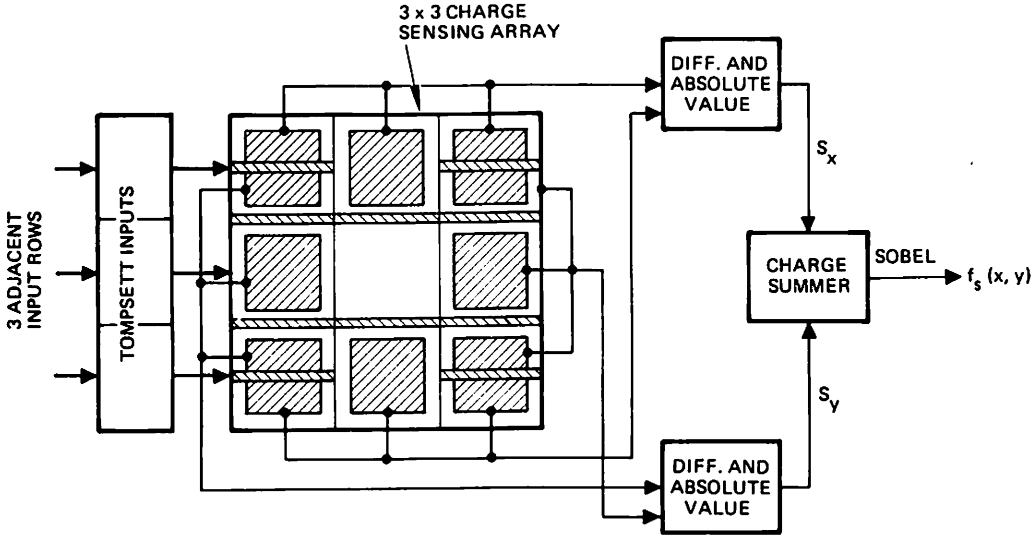


Figure 7. Schematic of CCD Sobel circuit.

The second is the purely algebraic operations, including absolute-value determination and summation. The gate interconnections shown perform a two-dimensional filtering operation with weightings equivalent to

$$W_X = \begin{bmatrix} 1/2 & 1 & 1/2 \\ 0 & 0 & 0 \\ -1/2 & -1 & -1/2 \end{bmatrix}, \quad W_Y = \begin{bmatrix} -1/2 & 0 & 1/2 \\ -1 & 0 & 1 \\ -1/2 & 0 & 1/2 \end{bmatrix} \qquad (10)$$

for the x and y directions, respectively. Variations in the area of the floating gates (as shown in the figure) are used to achieve the weighting coefficients (1/2, 1, etc.). The outputs S_X and S_Y go into a CCD absolute-value circuit and are summed as shown to form the complete Sobel $f_s(x,y)$. In this way, a valid output is computed each clock cycle.

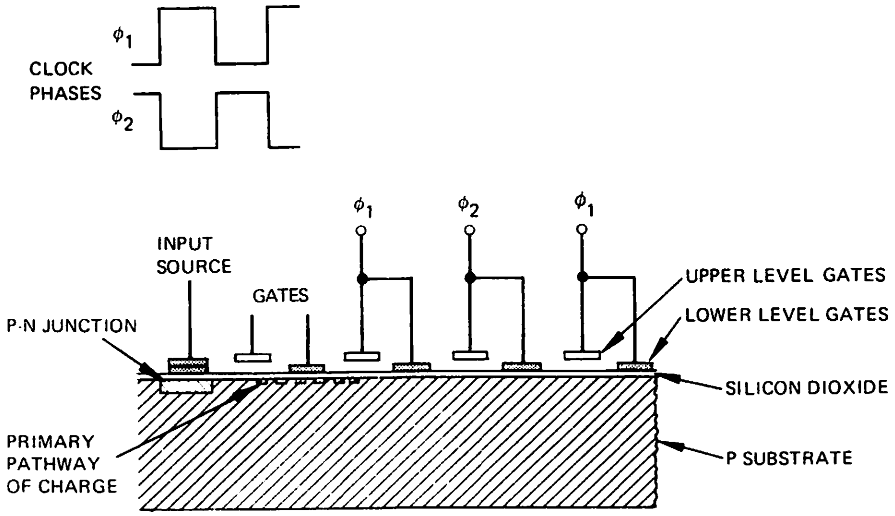


Figure 5. Concept of the CCD analog delay line.

in only one direction in response to the clocking waveforms illustrated. In our application, a charge equivalent to the intensity of each successive picture element is clocked into the device at each cycle. This requires that the clock speed be equivalent to the bandwidth of the analog video signal. The key to most signal-processing applications is a floating gate structure in which a single electrode capacitively couples to the signal charge across the oxide:

$$V_s = Q_s/C_{ox} \quad ,$$

where Q_S is the signal charge, C_{OX} is the oxide capacitance, and V_S is the corresponding output signal. Since the oxide capacitance C_{OX} is equal to $\varepsilon A/t_{OX}$, where A is the gate area, the outputs can be weighted by simply varying the gate length W_i as shown in Figures 6 and 7. The output then becomes $V_O = Q_S(t) * W(*)$, where * is the convolution junction. This is the basis of the CCD analog transversal filter. In our case, since we are operating on two-dimensional data equivalent to the 3 x 3 array, the parallel structure shown in Figure 6 is necessary. Here, three adjacent lines of video data are clocked in synchronism through the structure, and the output is taken from the array of nine identical gates shown. In this case, the output as a function of time t can be written

$$V_m(t) = W \cdot \begin{bmatrix} V_1(t) + V_1(t - T) + V_1(t - 2T) \\ V_2(t) + V_2(t - T) + V_2(t - 2T) \\ V_3(t) + V_3(t - T) + V_3(t - 2T) \end{bmatrix} \quad , \tag{8a}$$

where t is the running time parameter, T is the clock period, and W is the effective gate weighting. In the spatial domain, this is equivalent to

$$f(x,y) \otimes \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad , \tag{8b}$$

which corresponds directly to the blurr or local average function described in Eq. 2.

The equivalent structure for the Sobel edge detection is shown in Figure 7. Two types of operations are involved in the calculation of the Sobel algorithm. The first is the charge sensing and weighting necessary for the detection of the orthogonal edge components:

$$\begin{aligned} S_x &= f(x-1,y+1) + 2f(x,y+1) + f(x+1,y-1) - f(x-1,y-1) + 2f(x,y-1) + f(x+1,y-1) \\ S_y &= f(x-1,y+1) + 2f(x-1,y) + f(x-1,y-1) - f(x+1,y+1) + 2f(x+1,y) + f(x+1,y-1) \; . \end{aligned} \tag{9}$$

4. Performance Evaluation of the Processor

The two test chips that perform all the algorithms discussed in Section 2 are n-channel, two-phase CCDs with an approximate area of 200 mils x 200 mils. In our present experiments, we have operated the circuits in two modes. The first used the University of Southern California's stored data base. The data is formatted by an 8080A microprocessor prior to processing; this limits the speed of operation to about 10 kHz and 4-bits intensity resolution.[6] The performance of the devices in this mode is shown in Figure 8. We estimate the overall accuracy of the devices to be better than the maximum 4-bit display capability. More recently, we increased the clock rates to approximately 4 MHz with a resulting accuracy and dynamic range equivalent to 16 gray levels. A schematic of this test set up is shown in Figure 9. The input signal is derived from a vidicon camera with an interlace format as shown. A CCD analog store, the Fairchild* CCD 221 chip with a 488 x 380 element array, was used to provide a single field delay that enabled the interlace format to be removed. Two analog line delays were then used to provide access to three adjacent lines of video to allow continguous processing. These three lines then provided the data input to the processing array; the output is then mixed to form standard composite video for display on the video monitor.

To provide the maximum quality output from this system, the master clocks should be operated at 7.5 MHz. This would result in a resolution of approximately 525 x 525 lines. In the experiments reported here, we operated at approximately a 3.75-MHz clock rate. This resulted in a full resolution of 525 lines in the vertical direction, but a degraded performance in each horizontal line. The resulting asymmetry can be seen in Figure 10, which shows the output for the processor operating directly from the vidicon and typical data. At these rates, the dynamic range is equivalent to approximately 16 gray levels. The local averaging and edge detection are as effective at these rates as with the stored data experiment. The unsharp masking shown provides a similar effect to the Sobel edge detection but

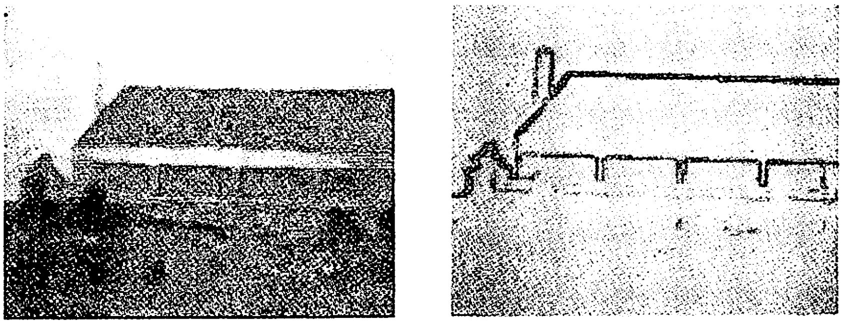


Figure 8. Example of processor performance operating a stored data at 10-kHz rates.

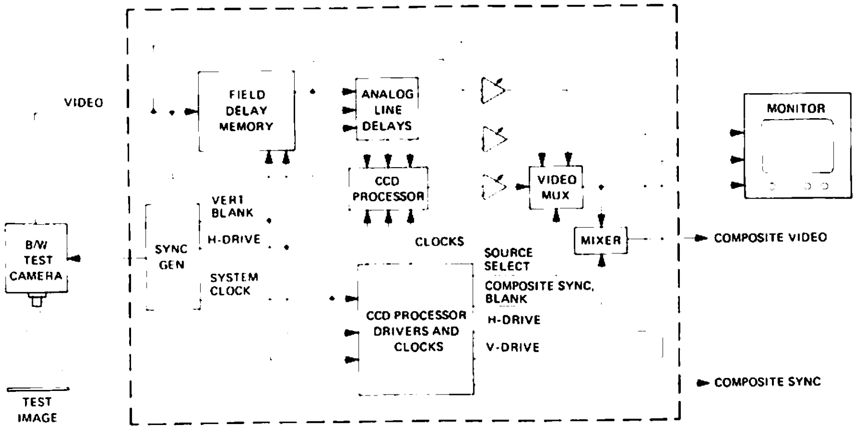


Figure 9. Schematic of real-time image processing system.

* This device was furnished by Fairchild Imaging System as part of a subcontract for the Night Vision Laboratories Contract No. DAAK70-77-C-0216.

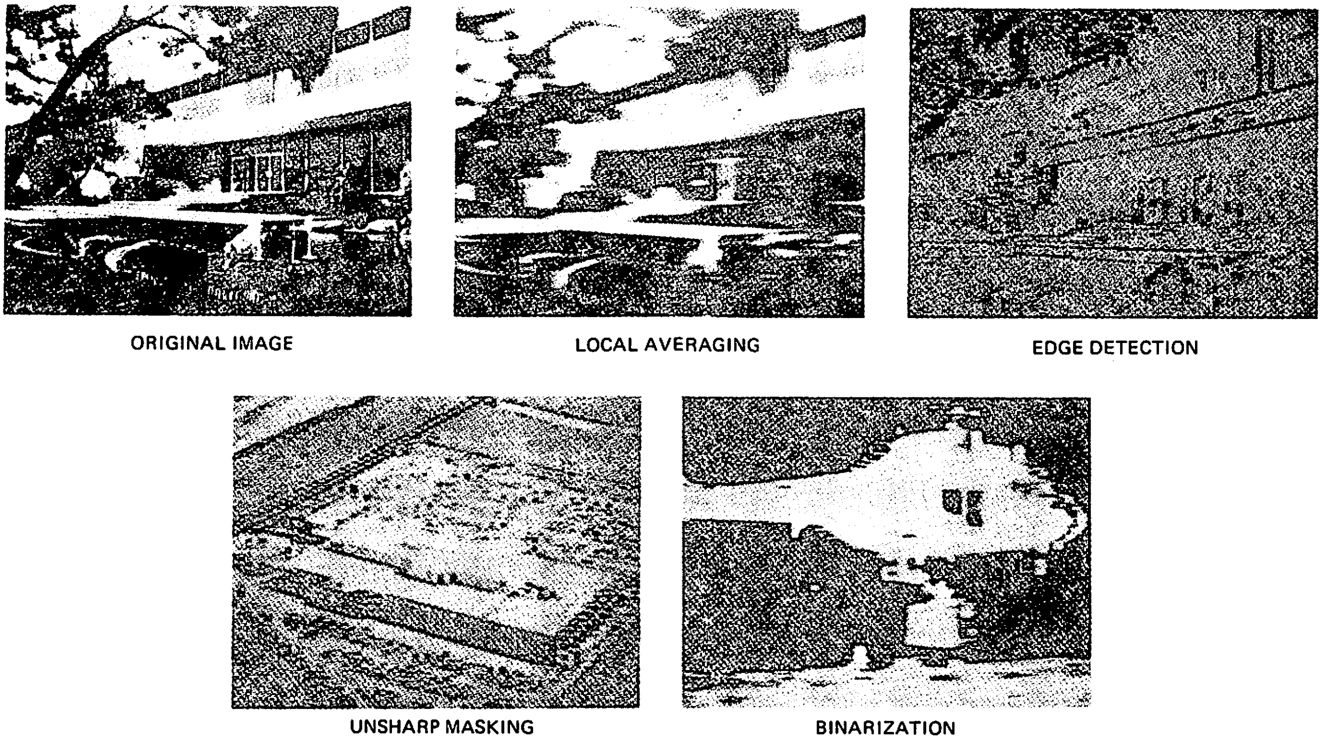


Figure 10. Performance of processor at real-time data rates.

provides some gray levels at reduced dynamic range. This can be important to limited bandwidth systems. The binarizer is adjusted to effectively produce a silhouette of the dark areas, which is useful in higher level processing for target identification, for example. The binarization algorithm is inherently unstable in that the center pixel and the surrounding mean typically have very similar intensities. In a uniform background, for example, the mean and center are identical, which results in binarizing only the thermal noise. In our experiments, this stability is removed by a varying offset at the comparator.

5. Conclusions

The results shown here indicate the feasibility of using a single integrated circuit to perform some of the preprocessing functions of use in higher-level or symbolic processing. Further, it is evidently feasible to operate at real-time rates and perform the computations without storage. Our work is continuing in an effort to combine these low-level operations to adaptive processing and the higher-level manipulation.

6. References

1. Pratt, W. K., Digital Image Processing, John Wiley & Sons, New York, 1978.
2. Andrews, H. C., and Hunt, B. R., Digital Image Restoration, Prentice-Hall, Englewood Cliffs, N.J., 1977.
3. Reis, J. J., Robinson, G. S., and Lucaro, A. B., Paper 155-04 presented at this Symposium, to be published in Proceedings, Vol. 155.
4. Duda, R. O., and Hart, P. E., Pattern Classification and Scene Analysis, John Wiley & Sons, New York, 1973.
5. Boyle, W. S., and Smith, G. E., Bell Syst. Tech. J. 49, pp 587-593, 1970.
6. Nudd, G. R., "CCD Image Processing Circuitry," pp 142-173, University of Southern California Semiannual Technical Report, March 1977, Andrews, H. C., Project Director.

5. Recent Institute Personnel Publications

1. A. Armand, D. Boswell, J. Michaelson, A.A. Sawchuk, B.H. Soffer and T.C. Strand, "Real-Time Nonlinear Processing with Halftone Screens," 1978 Annual Meeting, Optical Society of America, San Francisco, October 1978, Journal Optical Society of America, Vol. 68, p. 1361.

2. A. Armand, D. Boswell, A.A. Sawchuk, B.H. Soffer and T.C. Strand, "New Methods for Real-Time Nonlinear Optical Processing," 1978 Annual Meeting, Optical Society of America, San Francisco, October 1978, Journal Optical Society of America, Vol. 68, p. 1361.

3. J. Bescos, I. Glaser and A.A. Sawchuk, "Restoration of Color Images Degraded by Chromatic Aberrations," Submitted to Applied Optics.

4. L.M. Frantz, A.A. Sawchuk, and W. Von der Ohe, "Real-Time Optical Phase Measurement," 1978 Annual Meeting, Optical Society of America, San Francisco, October 1978, Journal Optical Society of America, Vol. 68, p. 1414.

5. C.K. Hsueh and A.A. Sawchuk, "Computer Generated Double Phase Holograms," Applied Optics, Vol. 17, pp. 3874-3883, December 15, 1978.

6. Ramakant Nevatia, "Image Understanding Systems," Edited, Proceedings of SPIE Symposium, San Diego, Ca., Aug. 1978.

7. Ramakant Nevatia and K. Ramesh Babu, "Linear Feature Extraction," Proceedings of ARPA Image Understanding Workshop, Pittsburgh, Pa., Nov. 1978, pp. 73-78.

8. Ramakant Nevatia and K. Ramesh Babu, "Linear Feature Extraction and Description," submitted to the International Joint Conference on

Artificial Intelligence, Aug. 1979.

9. Ramakant Nevatia and Keith Price, "Locating Structures in Aerial Images," Proceedings of the Fourth International Joint Conference on Pattern Recognition, Kyoto, Japan, No. 1978, pp. 686-690.

10. Ramakant Nevatia, Keith Price and Felicia Vilnrotter, "Describing Natural Textures," submitted to the International Joint Conference on Artificial Intelligence, Aug. 1979.

11. R. Ohlander, K. Price and R. Reddy, "Picture Segmentation Using a Recursive Region Splitting Method," Computer Graphics and Image Processing, Vol. 8, 1978, pp. 313-333.

12. Keith Price and R. Reddy, "Matching Segments of Images," IEEE Trans. PAMI, Vol. 1, 1979, pp. 110-116.

13. A.A. Sawchuk, "Artificial Stereo," Applied Optics, Vol. 17, pp. 3869-3873, December 15, 1978.

14. A.A. Sawchuk, M.J. Peyrovian and F. Davarian, "Discrete Representation of Image Degradation Using Monospline Quadrature Formulas," IEEE Transactions on Acoustics, Speech and Signal Processing, Vol. ASSP-26, pp. 473-475, October 1974.