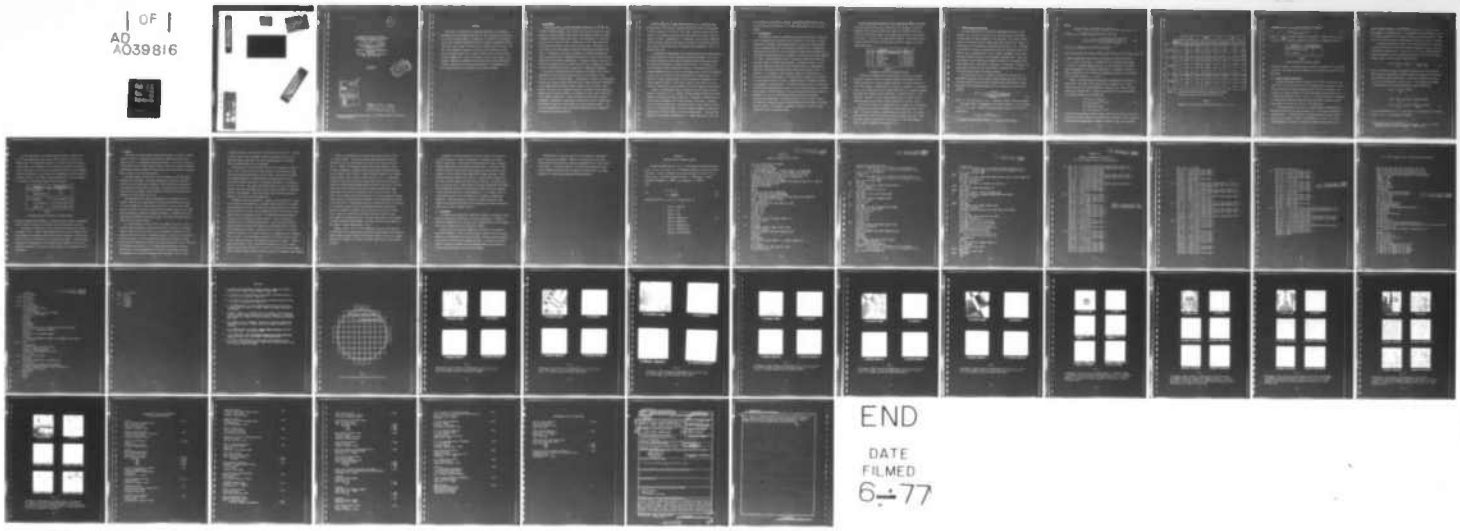
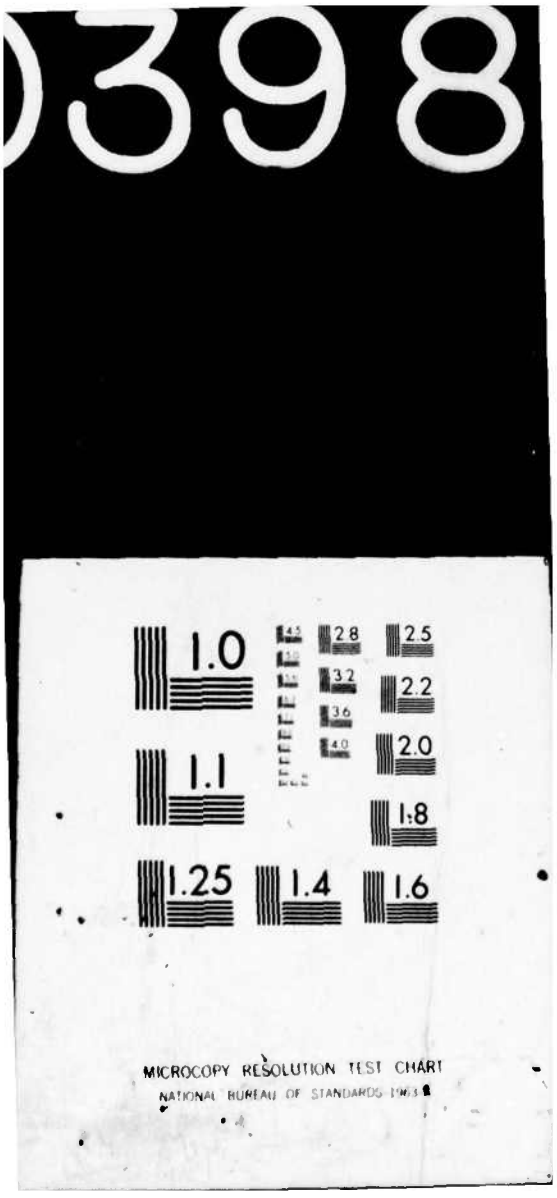
AD-A039 816 RENSSELAER POLYTECHNIC INST TROY N Y DEPT OF ELECTRI--ETC F/G 9/4
AN EMPIRICAL STUDY OF SELECTED APPROACHES TO THE DETECTION OF E--ETC(U)
MAR 77 R W FRIES, J W MODESTINO N00014-75-C-0281
UNCLASSIFIED TR-77-1 NL

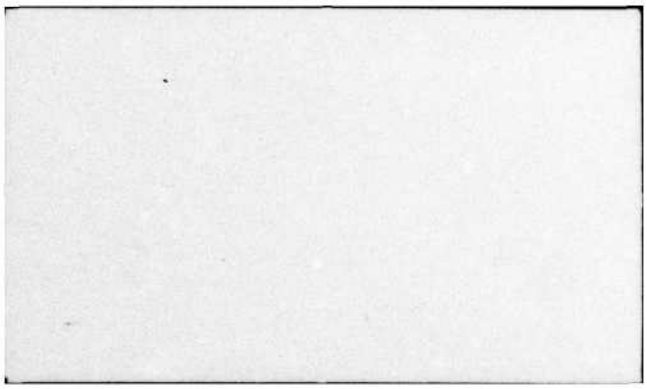
OF
AD
A039816



END

DATE
FILMED
6-77


0398
1.0
1.1
1.25
1.4
1.6
2.8
3.2
3.6
4.0
2.5
2.2
2.0
1.8
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963



(12)

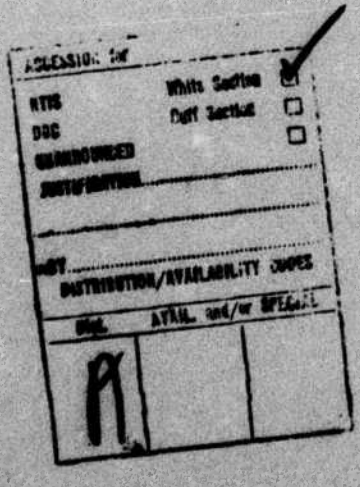
An Empirical Study of Selected Approaches to the Detection of Edges in Noisy Digitized Images

R. W. Fries and J. W. Modestino

Communication and Information Processing Group

Electrical and Systems Engineering Department
Rensselaer Polytechnic Institute
Troy, New York 12181

TR 77-1
March 1977



This work was performed under Contract No. N00014-75-C-0281 for the Office of Naval Research.

ABSTRACT

This paper is concerned with an empirical evaluation of the relative performance of several selected edge detectors for the detection of edge structure in noisy digitized images. In particular, we consider both the Hueckel operator and a recently introduced class of edge detectors implemented as two-dimensional infinite inpulse response (IIR) or recursive digital filters. The latter were originally developed on the basis of least mean-square Wiener spatial filtering concepts for an assumed stochastic model of edge structure in typical imagery data. These two schemes are compared with respect to performance, computational complexity and suitability for specific applications. Problems in making these comparisons are discussed. The implementations of both operators used for this comparison are described in detail.

I. Introduction:

A recurring problem in pattern recognition applications is the detection of edges in digitized images. Here we utilize the intuitive notion of an edge element as a picture element (pixel) lying on the boundary between two objects or regions of widely different gray levels. The requirements of a good edge detector in this context are that it respond only to true edge structure and be relatively insensitive to noise and/or spurious detail. Additionally, we would expect that the edge detector possess a computationally efficient realization. Unfortunately, very little information is available concerning the relative performance and/or computational complexities of various edge detectors operating on wide classes of imagery data. A possible exception is the work by Bullock [1] although even here a rather restricted class of images were considered and no consideration was given to questions of computational complexity.

This paper is concerned then with an empirical evaluation of the relative performance of several selected edge detectors for the detection of edge structure in noisy or textured images. In particular, we consider both the Hueckel operator [2],[3] and a recently introduced class of edge detectors [4] implemented as two-dimensional (2-D) infinite impulse response (IIR) or recursive digital filters. The latter were originally developed on the basis of least mean-square Wiener spatial filtering concepts for an assumed stochastic model of edge structure in typical imagery data. These two schemes are compared with respect to performance, computational complexity and suitability for specific applications. For comparison purposes we also consider the 4-point Laplacian operator. The specific implementations of both the Hueckel and Wiener operators employed in this study are described in detail.

Relative evaluation of edge detector performance is of necessity somewhat heuristic. The particular application generally dictates the seriousness of missed, spurious, or excessively thick edges. Indeed, for most real-world images, the differentiation between true edge structure and spurious detail is somewhat arbitrary. Nevertheless, it is expedient to rely on visual evaluation criterion,however subjective this may be, to assess the accuracy with which the edge detector output represents the essential "edge information". This is the approach taken here.

In assessing the computational burden associated with various edge detectors, it is again difficult to define meaningful quantitative measures. The cost of an operation depends upon the precision required as well as the type of operation (addition/subtraction vs. multiplication/division). The relative costs of the operations varies from machine to machine for programs written at the assembly language level and is even less determinate when a higher level language is used. Also the inherent structual complexity of the algorithm must be considered as a simple algorithm lends itself more readily to implementation in a special hardware processor. To the extent possible, these factors have been taken into consideration in the assessment of relative computational complexity.

Finally, all experimental work was conducted on the RPI image processing facility. Basically, this is a versatile hardware/software system for the digitization, processing and display of image data. The system is configured around a Varian 620/i minicomputer with 32K core storage, and includes a 5 megabyte disk for storage of image files. Additional details of this system are described in [5]. The software support is coded in assembly language to minimize processing times so that the facility can be used interactively. In the work described here, the longest processing time for a single operation on a 256 x 256

raster image is on the order of a minute. The system includes locally written routines that provide efficient I/O for fixed length random access files (e.g., images).

II. Preliminaries:

In choosing a criterion for comparing the relative performance of these operators, consideration must be given to the intended use of the output. An edge operator which responds to the boundary between two objects by indicating many small line segments which are close together and parallel but not necessarily connected would not present a serious problem to a higher level processor which was capable of interpreting each line segment in context with surrounding segments. To a contour tracing algorithm, however, the excess of marked edge points plus the large number of alternate paths represent a computational burden. We are implicitly assuming here that the contour tracing algorithm is implemented as some form of heuristic tree searching algorithm such as described in [6]. In addition, if a constraint of finite storage is imposed, the multiplicity of potential paths so created makes it more likely that the algorithm will discard all correct paths and fail to extract a complete contour. The Hueckel operator is an example of such a processor; its results are based on data in a local area and need further interpretation in order to provide conclusions about the content of the image. The Wiener operator, on the other hand, makes decisions about the presence of an edge on a point-by-point basis but utilizes observations over the entire image field. It has been designed so that it tends to divide the image into contiguous disjoint regions by marking as edge elements continuous strings of points. This feature is especially valuable to a contour tracing algorithm of the tree searching variety.

Obtaining meaningful measurements of the computational burden associated with an algorithm requires careful consideration of the types of operations performed and the precision required for each and is further complicated if conditional branches are present. The programs described here use two levels of precision: 16 bit and 31 bit fixed point arithmetic. The following table, although somewhat arbitrary, has been used to provide a measure of relative computational costs:

Operation	Cost
16 bit addition or subtraction	1 operation
16 bit multiply	3 operations
31 bit add-subtract	2 operations
31 bit multiply	12 operations
31 bit square root	20 operations

TABLE 1

Assignment of Relative Computational Costs

In making these comparisons it should be kept in mind that a recursive algorithm is generally easier to program optimally and furthermore is more readily implemented as a special hardware processor. The Hueckel operator is not of this category. When described in a high level language it is deceptively simple; when expressed in terms of machine level code (i.e., assembly language) it becomes very lengthy and quite complicated. The Wiener operator, on the other hand, consists of a 2-D recursion relation which is readily written in assembly language and for which a special processor could easily be built.

The method of useage of the operator must also be taken into consideration as situations may occur where information is needed about only a part of the image. The Hueckel operator is readily applied to subsections of the image since each application uses as input only data in a finite local area. Since the Wiener operator is recursive it must be applied to the entire image in order to obtain information about any part of it.

III. Wiener Operator Description:

The Wiener edge detector is derived by formulating the problem of edge detection in the presence of noise as one of least mean-square error spatial filtering where the desired operation is the Laplacian. This requires specification of a stochastic model for images possessing inherent edge structure. A useful model for this purpose has been described previously in [4]. According to this model, the plane is partitioned into disjoint rectangular regions, the sides of which are determined by a pair of mutually independent Poisson point processes evolving along each of the orthogonal coordinate axes. Gray levels are assigned to these elementary rectangles according to a 2-D discrete Gauss-Markov random field. Two descriptive parameters are available: λ represents the edge density in events per unit distance† while ρ represents the correlation coefficient of gray levels in contiguous regions.

The optimum Wiener filter is completely specified in terms of these two parameters in addition to the quantity ζ which represents the SNR of the edge structure process vis-à-vis the white background noise and/or spurious detail. As shown in [4] the resulting Wiener filter is isotropic with system transfer function

$$H_0(r) = \frac{r^2 e^{-r^2/2}}{1 + \dfrac{[r^2+2(1-\rho)^2\lambda^2]\sqrt{r^2+(1-\rho)^2\lambda^2}}{8(1-\rho)\lambda\zeta}} \quad ; r \geq 0 \tag{1}$$

where $r^2 = \omega_1^2 + \omega_2^2$ and ω_i, i=1,2, represents a spatial frequency component.

The Wiener filter is implemented in the discrete domain as a 2-D IIR digital filter (cf. [7], [8]). Briefly, the system transfer function of this filter is described by

$$H(z_1,z_2) = z_1^{-1/2} z_2^{-1/2} H_1(z_1,z_2) \tag{2}$$

† Actually in what follows we assume λ is measured in events/pixel.

where

$$H_1(z_1,z_2) \triangleq H_0(z_1,z_2)+z_1H_0(z_1^{-1},z_2)+z_2H_0(z_1,z_2^{-1})+z_1z_2H_0(z_1^{-1},z_2^{-1}) \quad (3)$$

and $H_0(z_1,z_2)$ is a simple first-order section of the form

$$H_0(z_1,z_2)=A\left[\frac{1-1/2(b_{11}+1)(z_1^{-1}+z_2^{-1})+b_{11}z_1^{-1}z_2^{-1}}{1+a_{10}(z_1^{-1}+z_2^{-1})+a_{11}z_1^{-1}z_2^{-1}}\right] \quad (4)$$

which can be implemented by the recursion relation

$$Y_{m,n}=-a_{10}(Y_{m-1,n}+Y_{m,n-1})-a_{11}Y_{m-1,n-1}+A[X_{m,n}-\tfrac{1}{2}(b_{11}+1)(X_{m-1,n}+X_{m,n-1})+b_{11}X_{m-1,n-1}] \quad (5)$$

A computer program has been written for determining the three coefficients a_{10}, a_{11} and b_{11} and the gain A according to an iterative gradient procedure to result in a frequency response for $H_1(z_1,z_2)$ which provides a least mean-square approximation to the desired response $H_0(r)$. The details of this program are described in [9]. In Table 2 we summarize the results of this procedure for selected values of ρ, λ, and ζ.

Finally, one further stage of processing is performed before declaring an edge present. The output of the Wiener filter is examined in 3 x 3 arrays centered upon the point under examination at position (i,j). Each of the four symmetrically opposite pairs of points are examined and the pair corresponding to the largest magnitude of the quantities

$$\begin{aligned}\Delta_1 &= X_{i-1,j-1}-X_{i+1,j+1}\\ \Delta_2 &= [X_{i,j-1}-X_{i,j+1}]*1.414\\ \Delta_3 &= X_{i+1,j-1}-X_{i-1,j+1}\\ \Delta_4 &= [X_{i-1,j}-X_{i+1,j}]*1.414\end{aligned} \quad (6)$$

is selected and a tentative direction code is assigned to the point on the basis of the pair of points chosen and the sign of their difference. Finally the above direction code and magnitude are accepted as marking an edge if for specified

		$\zeta = \infty$	ζ=10dB			ζ=3dB		
	Filter Coeff.	$-1<\rho<1$ [1]	ρ=-0.9	ρ=0.0	ρ=0.5	ρ=-0.9	ρ=0.0	ρ=0.5
λ=0.0125	b_{11}	-0.8256	-0.8924	-0.8252	-0.5736	-0.8264	-0.8256	-0.7556
	a_{10}	-0.2149	-0.2565	-0.3217	-0.4654	-0.3602	-0.3840	-0.4815
	a_{11}	0.0098	-0.2189	-0.1579	0.0449	-0.1540	-0.1461	0.0102
	A	0.115	0.053	0.038	0.025	0.022	0.015	0.0088
λ=0.0250	b_{11}	-0.8256	-0.9396	-0.8460	-0.8228	-0.8664	-0.8120	-0.8260
	a_{10}	-0.2149	-0.1989	-0.2849	-0.3106	-0.3075	-0.3686	-0.3782
	a_{11}	0.0098	-0.2490	-0.1687	-0.1761	-0.2040	-0.1350	-0.1534
	A	0.115	0.07	0.053	0.04	0.034	0.023	0.015
λ=0.050	b_{11}	-0.8256	-0.9244	-0.7980	-0.8480	-0.8896	-0.8228	-0.7636
	a_{10}	-0.2149	-0.1944	-0.2925	-0.2821	-0.2665	-0.3326	-0.3929
	a_{11}	0.0098	-0.1929	-0.0884	-0.1696	-0.2213	-0.1575	-0.0895
	A	0.115	0.084	0.067	0.054	0.048	0.034	0.023

Notes: 1. For $\zeta=\infty$ filter coefficients are independent of λ and ρ.

2. Additional results are available from the authors.

TABLE 2

Summary of Filter Parameters For Selected Values of λ, ρ and ζ

parameters F and T the following inequality is satisfied;

$$\Delta^2 - F[\sum_{k,\ell} X_{i+k,j+\ell}]^2 \geq T \qquad (7)$$

Here $\Delta = \max_{1\leq i\leq 4} \Delta_i$ and the summation is over the 3 x 3 array centered on location (i,j). The computations required at each pixel are summarized in Table 3.

Operation	Required/Pixel
16 bit multiplication	22
16 bit addition	39

TABLE 3

Computations/Pixel for Wiener Operator

According to Table 1, the Wiener Operator requires an equivalent of 105 operations per pixel. The program listing in Appendix II is included as a description of this algorithm.

IV. Hueckel Operator Description:

The Hueckel operator has been described in [3] and [4]. Briefly, the operator tries to fit an ideal edge element to the image data in the domain (a circular disk) such that a minimum mean-squared error criterion is satisfied. An initial step in this procedure is the least mean-square projection onto a set of eight basis functions $H_i(x,y)$, $i=1,2,\ldots,8$ or their discretized equivalents. A computationally efficient algorithm has been described for fitting an ideal edge element to the data on the basis of this projection.

The Hueckel operator as used here is a hybrid between an implementation employed by Bullock [1] and that described in [3]. The resulting algorithm is described below. It was implemented in assembly language on a Varian 620/i using 16 bit arithmetic to compute the projections A_i, $i=1,2,\ldots,8$ onto the corresponding basis functions. Due mainly to dynamic range problems it was

found necessary to employ 31 bit fixed point arithmetric in fitting an ideal edge element to the projected data. The calculation of the projections A_i, i=1, 2,...,8 takes advantage of all symmetries present in the basis functions. This part of the operator executes in approximately 25 seconds for a 256 × 256 image.

In computing discretized versions of the basis functions the circular disk is assumed partitioned into 69 square regions as indicated in Fig. 1. The discretized basis functions $H_i(j)$, i=1,2,...,8, j=1,2,...,69 were determined by numerical integration of the corresponding $H_i(x,y)$ over each of the 69 square regions. That is, for an interior square we define

$$H_i(j) = \int\int_{S_j} H_i(x,y)dxdy \quad ; \quad \begin{matrix} i=1,2,\ldots,8 \\ j=1,2,\ldots,69 \end{matrix} \tag{8}$$

where S_j is the square region of integration appropriate for the j'th square. Basis functions corresponding to squares along the periphery of the disk have been modified to include edge effects. In particular, sections of the disk not in any of the 69 squares as shown in Fig. 1 are included in the region of integration for the corresponding adjacent squares according to†

$$\begin{aligned} H_i(2) &= \int\int_{S_2 \cup A} H_i(x,y)dxdy \\ &\vdots \qquad\qquad \vdots \\ H_i(5) &= \int\int_{S_5} H_i(x,y)dxdy + \frac{1}{2}\int\int_{D} H_i(x,y)dxdy \\ H_i(6) &= \int\int_{S_6} H_i(x,y)dxdy + \frac{1}{2}\int\int_{D\cup E} H_i(x,y)dxdy \end{aligned} \tag{9}$$

and so forth around the disk. For completeness, the underlying basis functions $H_i(x,y)$, i=1,2,...,8 are defined in Appendix I .

† It should be noted that the basis functions $H_i(x,y)$, i=1,2,...,8 vanish identically outside the disk of radius $r = \sqrt{69/\pi}$ pixels.

For each application of the Hueckel operator an edge is indicated in a 4 x 4 block by marking each pixel whose center is within $\sqrt{2}$ pixels of the declared edge. The computations required to carry this out have not been included as part of the computation count. Since the operator must be applied once for each group of 16 pixels the total number of operations for each application has been divided by 16 to produce the computational costs per pixel as indicated in Table 4. The range of values indicated for the 31 bit arithmetic is the result of conditional branches within the program.

Operation	Required/Pixel
16 bit addition	23.44
multiplication	6.94
31 bit addition	1.63-3.25 (average 2.44)
multiplication	2.5 -4.63 (average 3.57)
square root	.125-1.56(average 0.84)

TABLE 4

Computational Costs Per Pixel for Hueckel Operator

Using the average values from Table 4 together with the normalized computational costs in Table 1 we arrive at an effective cost of 109 operations per pixel.

Two parameters CONF and DIFF were included in the descriptions provided by Hueckel [3], [4]. There is some ambiguity as to the physical significance of these parameters but generally CONF sets the desired confidence level while DIFF sets the smallest level difference to be accepted as an edge. A program listing describing the implementation of the Hueckel operator employed in this work is provided in Appendix III from which the exact useage of these parameters can be established.

V. Results:

The Wiener and Hueckel operators have been exercised on two types of images: digitized versions of actual scenes and synthetic images with additive white Gaussian noise (i.e., independent from pixel-to-pixel). In all cases we have included results for the 4-point Laplacian operator strictly for comparison purposes. The particular images considered here were chosen because they exhibit performance that is typical of what has been observed.

The times required for execution of the two programs are not sufficiently different to be noteworthy. The Hueckel operator was divided into two parts: projection onto the basis vectors (execution time 25 seconds) and computation (65 seconds) for a total time of 90 seconds. The Wiener operator likewise has been divided into two parts: linear filtering (56 seconds) and detection (28 seconds) for a total time of 84 seconds.

The performance of the various edge detectors is illustrated in Fig.'s 2 through 7 when the input image consists of the previously described stochastic edge process imbedded in a white Gaussian noise field. In all cases the parameters of the Wiener operator (T,F) and that of the Hueckel operator (CONF, DIFF) have been empirically adjusted to optimize performance. The Laplacian has been consistently thresholded on the positive peak. This causes some displacement of edges toward the brighter regions.

It is clear from Fig.'s 2 and 3 that in the absence of noise (i.e., $\zeta=\infty$) both the Wiener and Laplacian detectors perform well when there is a pronounced change in gray level across an edge (i.e., large negative correlation) such as would be the case with $\rho=-0.9$. The Hueckel operator, on the other hand, exhibits a curious inability to resolve closely spaced edges and a tendency to provide spurious edge declarations particularly in regions of high edge density. This problem becomes more pronounced as λ, the average edge density/pixel, increases. The situation is further aggravated in the presence of less pronounced edges corresponding to an increased correlation ρ across an edge. This is clearly

illustrated in Fig. 4 for the case $\lambda = 0.05$ and $\rho = 0.5$ still with $\zeta = \infty$. It should be noted, however, that the Hueckel operator is being used strictly in the edge mode. Had it been used in the line or edge-line mode it is possible that this problem might not be as pronounced.

The behavior as a function of SNR ζ with $\rho = -0.9$ is illustrated in Fig.'s 5 and 6. Here $\lambda = 0.0125$ so that with this low value of edge density the edge resolution questions raised above are not an issue. Observe that for $\zeta \leq 10$dB the ordinary Laplacian is virtually useless in discerning edge structure. The situation becomes even worse with increasing correlation ρ as illustrated in Fig. 7 for $\rho = 0.5$. The Wiener and Hueckel operators, on the other hand, are both able to detect considerable edge structure while preserving a considerable amount of noise immunity. Indeed, the Hueckel operator possesses excellent immunity to this type of salt-and-pepper noise.

Experiments were also conducted on a synthetic image consisting of a light circle (gray level 64) intersecting a polygonal figure (gray level 31) against a white Gaussian noise field. The SNR is here defined in terms of the ratio of gray level within the circle to the standard deviation of the background noise. Typical results are illustrated in Fig. 8 for SNR=6dB and selected parameter choices for both the Wiener and Hueckel operators. Again the Laplacian is virtually useless in discerning edge structure. Both the Wiener and Hueckel operators exhibit excellent noise immunity with the Hueckel operator offering some advantage in this regard. Furthermore, the noise immunity of the Hueckel detector is observed to be a definite function of the parameter DIFF by comparing Fig.'s 8c and 8d. The noise immunity properties of the Wiener detector depend in a more complicated way upon the parameters defining this operator. Parameter choices illustrated in Fig. 8 were determined empirically to be near optimum. Observe that the Wiener operator has a tendency to produce connected edge segment while the Hueckel operator tends to produce disconnected or fragmented edge segments.

A Number of experiments have also been conducted with various classes of real-world images. Results for a typical head-and-shoulders image are illustrated in Fig. 9 while results for a typical chest X-ray are shown in Fig. 10. Observe the flexibility afforded by the Wiener detector in that by logical assignment of parameter values one may extract different portions of the "relevant" edge structure in typical real-world images. This is demonstrated clearly in Fig. 10 and to a lesser extent in Fig. 9. In particular, from Fig.'s 10e and 10f note that the Wiener edge detector can be "tuned" to extract the closely spaced and sharp edge structure(as in Fig. 10e by the parameter choice λ=0.05 and ρ=-0.9)or the less frequent and more gradual edge structure(as in Fig. 10f by the parameter choice λ=0.0125 and ρ=0.5). The former would be useful if one were interested in rib structure while the latter would be useful if instead one were interested in merely extracting the outline of the chest cavity. Corresponding results are illustrated in Fig.'s 10c and 10d for the Hueckel operator where now the parameters CONF and DIFF have been empirically adjusted to emphasize fine and coarse structural detail respectively. Observe the inability of the Hueckel operator to discern closely spaced edges, the extreme sensitivity to spurious image detail and the tendency to produce disconnected edge segments. The Wiener operator is far superior in this regard. Similar comments apply to Fig. 9.

Finally, in Fig.'s 11 and 12 the performance of these two edge detectors is illustrated for two images exhibiting extensive and predominant edge structure. Observe again the superior behavior of the Wiener operator vis-à-vis the Hueckel operator in its ability to be "tuned" to extract either fine or coarse structural detail.

In summary the important features of the Wiener operator when applied to real-world images are: relative insensitivity to background noise and/or spurious image detail; the ability to discern between soft edges or shading (gradual transitions) and hard edges (step transitions); the ability to discriminate between closely spaced and more widely spaced edge structure; and a tendency to produce connected edge segments which segment the image. These properties have been shown to be of value in a contour tracing algorithm [6] for which the Wiener detector has been used as a front end. The Hueckel operator, on the other hand, is quite sensitive to spurious image detail and suffers from an inability to discern soft edges and/or closely spaced edge structure. A larger disk size and/or less coarse application (i.e., on smaller than 4 x 4 blocks) might help but only at the expense of an increase in computational complexity. The Hueckel operator also tends to produce disconnected edge segments thus requiring more higher level processing than the Wiener detector's output does in order to produce a contour.

VI. Conclusions:

Two edge detection schemes were compared with respect to performance, computational complexity and suitability for specific applications. Problems in making these comparisons were discussed and the implementations of both operators used for this comparison have been described in detail.

Both real world and synthetic images were considered for the purpose of evaluating performance. While the Hueckel operator exhibited better subjective immunity to salt-and-pepper noise than the Wiener, the Wiener operator excelled in its ability to discern between shading and edges. In addition, the Wiener operator's tendency to produce connected boundaries make it more suitable as a front-end processor for a contour tracing algorithm.

The difference in execution times for the two operators as implemented here was found to be negligible. However, the recursive structure of the Wiener operator lends itself more readily to implementation as a special hardware processor. The Hueckel operator, on the other hand, depends only upon local information and may therefore be applied to selected parts of an image resulting in a computational advantage in some applications.

Appendix I

Basis Functions for Hueckel Operator

The basis functions $H_i(x,y)$, $i=1,2,\ldots,8$ from which the discretized versions $H_i(j)$, $i=1,2,\ldots,8$, $j=1,2,\ldots,69$ are computed according to (8)-(9) are defined as follows. Let $(\hat{x}, \hat{y})$ represent the position relative to the center of the disk in pixels. The disk is assumed of radius $r = \sqrt{69/\pi}$ pixels. Define normalized coordinates $x = \sqrt{\pi/69}\,\hat{x}$ and similarly $y = \sqrt{\pi/69}\,\hat{y}$. Also

$$\delta^2 = x^2 + y^2 \tag{I-1}$$

and

$$Q = 2\left(\frac{1-\delta^2}{3\pi}\right)^{1/2} \tag{I-2}$$

Set $H_i(x,y)=0$ for $\delta^2 > 1$, $i=1,2,\ldots,8$ while for $\delta^2 \leq 1$:

$$\begin{aligned}
H_1(x,y) &= Q(5\delta^2-2) \\
H_2(x,y) &= 3Qx \\
H_3(x,y) &= 3Qy \\
H_4(x,y) &= \sqrt{18}\,Q(x^2-y^2) \\
H_5(x,y) &= \sqrt{18}\,Q(2xy) \\
H_6(x,y) &= \sqrt{45}\,Q(2\delta^2-1)x \\
H_7(x,y) &= \sqrt{45}\,Q(2\delta^2-1)y \\
H_8(x,y) &= Q(-2+17\delta^2-21\delta^4)
\end{aligned} \tag{I-3}$$

Appendix II

Program to Implement Wiener Operator

```
C       NPIX IS THE SIZE OF THE IMAGE
C       X IS THE INPUT IMAGE
C       Y IS THE FILTERED IMAGE
C       E IS AN ARRAY OF FLAGS:   0 FOR NO EDGE; -1 FOR AN EDGE
C       YL AND YR ARE ARRAYS USED TO HOLD CURRENT AND PREVIOUS
C       OUTPUTS OF THE LEFT AND RIGHT MOVING FILTER SECTIONS.
        DIMENSION X(NPIX,NPIX),Y(NPIX,NPIX),E(NPIX,NPIX)
        DIMENSION YL(2,NPIX),YR(2,NPIX)
C       CUR AND PREV POINT TO THE CURRENT AND PREVIOUS PARTS OF YL AND YR
C       THEY ARE INITIALIZED BELOW.
        INTEGER E,CUR,PREV
        CUR=1
        PREV=2
C       B11,A10,A11,AND A ARE PARAMETERS
C       OF THE FILTER THAT MUST BE ENTERED
C       F AND TRESH ARE PARAMETERS OF THE EDGE DETECTION ALGORITHM
        READ    B11,A10,A11,A,F,THRESH
        B10=-.5*(B11+1.)
C       INITIALIZE PREVIOUS LINE ARRAYS TO ZERO
        DO 100 J=1,NPIX
        YL(PREV,J)=0.
        YL(CUR,J)=0.
        YR(PREV,J)=0.
        YR(CUR,J)=0.
100     CONTINUE
C       SET FIRST LINE OF FILTERED IMAGE TO 0
        DO 200 J=1,NPIX
        Y(1,J)=0.
200     CONTINUE
C       MOVE DOWN THROUGH IMAGE LINE BY LINE
        DO 600 I=2,NPIX-1
C       MAKE OLD CURRENT LINE NEW PREVIOUS LINE
        IHOLD=CUR
        CUR=PREV
        PREV=IHOLD
C       SET FIRST AND LAST POINTS OF FILTERED IMAGE TO 0
        Y(I,1)=0.
        Y(I,NPIX)=0.
C       MOVE THROUGH THE LINE POINT BY POINT
        DO 400 J=2,NPIX-1
```

```
C       RIGHT MOVING FILTER SECTION
        YR(CUR, J)=-A10*(YR(CUR, J-1)+YR(PREV, J))-A11*YR(PREV, J-1)
       C           +X(I, J)+B10*((X(I, J-1)+X(I-1, J))+B11*X(I-1, J-1)
C       LEFT MOVING FILTER SECTION
        JL=NPIX-J+1
        YL(CUR, JL)=-A10*(YL(CUR, JL+1)+YL(PREV, JL))-A11*YL(PREV, JL+1)
       C            +X(I, JL)+B10*(X(I, JL+1)+X(I-1, JL))+B11*X(I-1, JL+1)
C       END POINT BY POINT LOOP
400     CONTINUE
C       SUM LEFT AND RIGHT MOVING FILTER OUTPUTS
        DO 500 J=2, NPIX-1
        Y(I, J)=YR(CUR, J)+YL(CUR, J+1)
500     CONTINUE
C       END LINE BY LINE LOOP MOVING DOWN
600     CONTINUE
C       ZERO LAST LINE OF FILTERED OUTPUT
        DO 700 J=1, NPIX
        Y(NPIX, J)=0.
700     CONTINUE
C       GET READY TO MOVE UP THROUGH THE IMAGE
C       ZERO PREVIOUS LINE ARRAY
        DO 1100 J=1, NPIX
        YL(CUR, J)=0.
        YL((PREV, J)=0.
        YR(CUR, J)=0.
        YR(PREV, J)=0.
1100    CONTINUE
C       MOVE UP THROUGH THE IMAGE LINE BY LINE
        DO 1600 II=2, NPIX-2
        I=NPIX+1-II
C       MAKE OLD CURRENT LINE NEW PREVIOUS LINE
        IHOLD=CUR
        CUR=PREV
        PREV=IHOLD
C       MOVE THROUGH THE LINE POINT BY POINT
        DO      1400 J=2, NPIX-1
C       RIGHT MOVING FILTER SECTION
        YR(CUR, J)=-A10*(YR(CUR, J-1)+YR(PREV, J))-A11*YR(PREV, J-1)
       C           +X(I, J)+B10*((X(I, J-1)+X(I+1, J))+B11*X(I+1, J-1)
C       LEFT MOVING FILTER SECTION
```

```
      JL=NPIX-J+1
      YL(CUR,JL)=-A10*(YL(CUR,JL+1)+YL(PREV,JL))-A11*YL(PREV,JL+1)
     C            +X(I,JL)+B10*(X(I,JL+1)+X(I+1,JL))+B11*X(I+1,JL+1)
C     END POINT BY POINT LOOP
1400  CONTINUE
C     ADD OUTPUT OF LEFT DOWN PLUS RIGHT DOWN TO LEFT UP PLUS RIGHT UP
C     AND MULTIPLY SUM BY A
      DO 1500 J=2,NPIX-1
      Y(I-1,J)=A*(Y(I-1,J)+YR(CUR,J)+YL(CUR,J+1))
1500  CONTINUE
C     END LINE BY LINE LOOP MOVING UP
1600  CONTINUE
C     END OF FILTERING - START OF DETECTION
C     ZERO FIRST AND LAST LINES OF EDGE DETECTOR OUTPUT
      DO 2100 J=1,NPIX
      E(I,J)=0
      E(NPIX,J)=0
2100  CONTINUE
C     MOVE THROUGH THE IMAGE LINE BY LINE
      DO 2400 I=2,NPIX-1
C     ZERO EDGE DETECTOR OUTPUT FOR FIRST AND LAST POINTS
      E(I,1)=0
      E(I,NPIX)=0
C     MOVE THROUGH THE LINE POINT BY POINT
      DO 2300 J=2,NPIX-1
      DIF1=ABS(Y(I-1,J-1)-Y((I+1,J+1))
      DIF2=ABS(Y(I-1,J)-Y(I+1,J))*1.414
      DIF3=ABS(Y(I-1,J+1)-Y(I+1,J-1))
      DIF4=ABS(Y((I,J-1)-Y(I,J+1))*1.414
      DIF=AMAX(DIF1,DIF2,DIF3,DIF4)
      SUM=Y(I,J-1)+Y(I,J)+Y(I,J+1)
     C    +Y(I-1,J-1)+Y(I-1,J)+Y(I-1,J+1)
     C    +Y(I+1,J-1)+Y(I+1,J)+Y(I+1,J+1)
      SUM=SUM*SUM*F
      FLAG=0
      IF (DIF**2-SUM.GT.THRESH) FLAG=-1
      E(I,J)=FLAG
C     END POINT BY POINT LOOP
2300  CONTINUE
C     END LINE BY LINE LOOP
2400  CONTINUE
      STOP
```

Appendix III

Program to Implement Hueckel Operator

A) 16-Bit Arithmetic Used in Calculating A1-A8

```
A1 = +H1( 1)*( X( 1)+X(13)+X(49)+X(65)+X(69)+X(57)+X(21)+X( 5) )
     +H1( 2)*( X( 2)+X(30)+X(48)+X(66)+X(68)+X(40)+X(22)+X( 4) )
     +H1( 3)*( X( 3)+X(31)+X(67)+X(39) )
     +H1( 6)*( X( 6)+X(12)+X(64)+X(58) )
     +H1( 7)*( X( 7)+X(11)+X(14)+X(50)+X(63)+X(59)+X(56)+X(20) )
     +H1( 8)*( X( 8)+X(10)+X(29)+X(47)+X(62)+X(60)+X(41)+X(23) )
     +H1( 9)*( X( 9)+X(32)+X(61)+X(38) )
     +H1(19)*( X(19)+X(15)+X(51)+X(55) )
     +H1(18)*( X(18)+X(16)+X(28)+X(46)+X(52)+X(54)+X(42)+X(24) )
     +H1(17)*( X(17)+X(33)+X(53)+X(37) )
     +H1(25)*( X(25)+X(27)+X(45)+X(43) )
     +H1(26)*( X(26)+X(34)+X(44)+X(36) )
     +H1(35)*( X(35) )
A2 = +H2( 5)*( X( 5)+X(69)-X( 1)-X(65) )
     +H2(21)*( X(21)+X(57)-X(13)-X(49) )
     +H2( 4)*( X( 4)+X(68)-X(66)-X( 2) )
     +H2(22)*( X(22)+X(40)-X(30)-X(48) )
     +H2(39)*( X(39)-X(31) )
     +H2( 6)*( X( 6)+X(58)-X(12)-X(64) )
     +H2( 7)*( X( 7)+X(59)-X(11)-X(63) )
     +H2(19)*( X(19)+X(55)-X(15)-X(51) )
     +H2(20)*( X(20)+X(56)-X(14)-X(50) )
     +H2( 8)*( X( 8)+X(60)-X(10)-X(62) )
     +H2(23)*( X(23)+X(41)-X(29)-X(47) )
     +H2(38)*( X(38)-X(32) )
     +H2(18)*( X(18)+X(54)-X(16)-X(52) )
     +H2(24)*( X(24)+X(42)-X(28)-X(46) )
     +H2(37)*( X(37)-X(33) )
     +H2(25)*( X(25)+X(43)-X(27)-X(45) )
     +H2(36)*( X(36)-X(34) )
A3 = +H3( 1)*( X( 1)+X( 5)-X(65)-X(69) )
     +H3(21)*( X(21)+X(13)-X(49)-X(57) )
     +H3( 2)*( X( 2)+X( 4)-X(66)-X(68) )
     +H3(22)*( X(22)+X(30)-X(48)-X(40) )
     +H3( 3)*( X( 3)-X(67) )
     +H3( 6)*( X( 6)+X(12)-X(64)-X(58) )
     +H3( 7)*( X( 7)+X(11)-X(63)-X(59) )
     +H3(20)*( X(20)+X(14)-X(56)-X(50) )
     +H3( 8)*( X( 8)+X(10)-X(62)-X(60) )
     +H3(23)*( X(23)+X(29)-X(47)-X(41) )
```

```
     +H3( 9)*( X( 9)-X(61) )
     +H3(19)*( X(19)+X(15)-X(51)-X(55) )
     +H3(18)*( X(18)+X(16)-X(52)-X(54) )
     +H3(24)*( X(24)+X(28)-X(46)-X(42) )
     +H3(17)*( X(17)-X(53) )
     +H3(25)*( X(25)+X(27)-X(45)-X(43) )
     +H3(26)*( X(26)-X(44) )
A4 = +H4(21)*( X(21)+X(13)+X(49)+X(57)-X(65)-X(69)-X( 5)-X( 1) )
     +H4(22)*( X(22)+X(30)+X(48)+X(40)-X(66)-X(68)-X( 4)-X( 2) )
     +H4(39)*( X(39)+X(31)-X(67)-X( 3) )
     +H4(20)*( X(20)+X(14)+X(50)+X(56)-X(11)-X(63)-X(59)-X( 7) )
     +H4(23)*( X(23)+X(29)+X(47)+X(41)-X(10)-X(62)-X(60)-X( 8) )
     +H4(38)*( X(38)+X(32)-X(61)-X( 9) )
     +H4(24)*( X(24)+X(28)+X(46)+X(42)-X(16)-X(52)-X(54)-X(18) )
     +H4(37)*( X(37)+X(33)-X(53)-X(17) )
     +H4(36)*( X(36)+X(34)-X(44)-X(26) )
A5 = +H5( 5)*( X( 5)+X(49)+X(65)+X(21)-X( 1)-X(13)-X(69)-X(57) )
     +H5( 4)*( X( 4)+X(48)+X(66)+X(22)-X(30)-X(68)-X(40)-X( 2) )
     +H5( 6)*( X( 6)+X(64)-X(12)-X(58) )
     +H5( 7)*( X( 7)+X(63)+X(50)+X(20)-X(11)-X(14)-X(59)-X(56) )
     +H5( 8)*( X( 8)+X(47)+X(62)+X(23)-X(10)-X(29)-X(60)-X(41) )
     +H5(19)*( X(19)+X(51)-X(15)-X(55) )
     +H5(18)*( X(18)+X(46)+X(52)+X(24)-X(16)-X(28)-X(54)-X(42) )
     +H5(25)*( X(25)+X(45)-X(27)-X(43) )
A6 = +H6( 5)*( X( 5)+X(69)-X( 1)-X(65) )
     +H6(21)*( X(21)+X(57)-X(13)-X(49) )
     +H6( 4)*( X( 4)+X(68)-X( 2)-X(66) )
     +H6(22)*( X(22)+X(40)-X(30)-X(48) )
     +H6(39)*( X(39)-X(31) )
     +H6( 6)*( X( 6)+X(58)-X(12)-X(64) )
     +H6( 7)*( X( 7)+X(59)-X(11)-X(63) )
     +H6(19)*( X(19)+X(55)-X(51)-X(15) )
     +H6(20)*( X(20)+X(56)-X(14)-X(50) )
     +H6( 8)*( X( 8)+X(60)-X(10)-X(62) )
     +H6(23)*( X(23)+X(41)-X(29)-X(47) )
     +H6(38)*( X(38)-X(32) )
     +H6(18)*( X(18)+X(54)-X(16)-X(52) )
     +H6(24)*( X(24)+X(42)-X(28)-X(46) )
     +H6(37)*( X(37)-X(33) )
     +H6(25)*( X(25)+X(43)-X(27)-X(45) )
```

```
      +H6(36)*( X(36)-X(34) )
A7 =  +H7( 1)*( X( 1)+X( 5)-X(65)-X(69) )
      +H7(21)*( X(21)+X(13)-X(49)-X(57) )
      +H7( 2)*( X( 2)+X( 4)-X(66)-X(68) )
      +H7(22)*( X(22)+X(30)-X(48)-X(40) )
      +H7( 3)*( X( 3)-X(67) )
      +H7( 6)*( X( 6)+X(12)-X(64)-X(58) )
      +H7( 7)*( X( 7)+X(11)-X(63)-X(59) )
      +H7(20)*( X(20)+X(14)-X(56)-X(50) )
      +H7( 8)*( X( 8)+X(10)-X(62)-X(60) )
      +H7(23)*( X(23)+X(29)-X(47)-X(41) )
      +H7( 9)*( X( 9)-X(61) )
      +H7(19)*( X(19)+X(15)-X(51)-X(55) )
      +H7(18)*( X(18)+X(16)-X(52)-X(54) )
      +H7(24)*( X(24)+X(28)-X(46)-X(42) )
      +H7(17)*( X(17)-X(53) )
      +H7(25)*( X(25)+X(27)-X(45)-X(43) )
      +H7(26)*( X(26)-X(44) )
A8 =  +H8( 1)*( X( 1)+X(13)+X(49)+X(65)+X(69)+X(57)+X(21)+X( 5) )
      +H8( 2)*( X( 2)+X(30)+X(48)+X(66)+X(68)+X(40)+X(22)+X( 4) )
      +H8( 3)*( X( 3)+X(31)+X(67)+X(39) )
      +H8( 6)*( X( 6)+X(12)+X(64)+X(58) )
      +H8( 7)*( X( 7)+X(11)+X(14)+X(50)+X(63)+X(59)+X(56)+X(20) )
      +H8( 8)*( X( 8)+X(10)+X(29)+X(47)+X(62)+X(60)+X(41)+X(23) )
      +H8( 9)*( X( 9)+X(32)+X(61)+X(38) )
      +H8(19)*( X(19)+X(15)+X(51)+X(55) )
      +H8(18)*( X(18)+X(16)+X(28)+X(46)+X(52)+X(54)+X(42)+X(24) )
      +H8(17)*( X(17)+X(33)+X(53)+X(37) )
      +H8(25)*( X(25)+X(27)+X(45)+X(43) )
      +H8(26)*( X(26)+X(34)+X(44)+X(36) )
      +H8(35)*( X(35) )
```

B) 31-Bit Arithmetic Used in Fitting Ideal Edge Element

```
REAL A1,A2,A3,A4,A5,A6,A7,A8,A26,A37,A27,IPU2
REAL E1,E2,E3,E4,E5,E6,G1,G2,G3,CX,CCX,CY,CCY
REAL MU,NU,U0,U1,U2,U3,KA,W,PP,BP,DP,DIFF,DIFFS
REAL EDJS,TP,CONF,CONFM1,RM,BM,DM,TM,THRESH,MAXBRT
THRESH = .0001
MAXBRT = 25.
DIFFS=DIFF**2
CONFM1=1-CONF
A26=A2**2+A6**2
A37=A3**2+A7**2
A27=A37+A26
IPU2=A1**2+A27+A4**2+A5**2+A8**2
IF IPU2.LT.27*PI/64 GO TO DISA
E1=(1/3)**.5*A1
E2=(2/3)**.5*A4
E3=(2/3)**.5*A5
E4=2*E1*E2+(A26-A37)/2
E5=2*E1*E3+A2*A3+A6*A7
G1=SIGN(E2*E4+E3*E5)*SQRT(E2**2+E3**2)
CCX=G1*E2+E4
CCY=G1*E3+E5
G2=SQRT(CCX**2+CCY**2)
CCX=CCX/G2
CCY=CCY/G2
E6=E2*CCX+E3*CCY
SIG2=E1**2+A27/2+A8**2+E6**2+E4*CCX+E5*CCY
IF SIG2/IPU2+(1-CONF)*SIG2**2/(SIG2**2+DIFF**2).LE.1.0 GO TO DISA
G3=(2*(1+CCX))**.5
IF G3-THRESH<0 THEN CY=SIGN(CCY) ELSE CY=CCY/G3
CX=G3/2
U0=A2*CX+A3*CY
U1=(E1+E6)*3**-1/2
U2=(A6*CX+A7*CY)*5**-.5
U3=(U1-2*A8)/5
NU=U1**2-U2*U0
IF ABS(NU)-THRESH.GT.0 GO TO NTRIC
IF ABS(MU).LE.THRESH GO TO DISA
IF ABS(U0).GE.THRESH GO TO TAU0NZ
IF ABS(U1).GE.THRESH GO TO TAU1NZ
IF ABS(U2).LT.THRESH GO TO DISA
```

```
        KA=U3/U2
        GO TO KTEST
TAU0NZ  KA=U1/U0
        GO TO KTEST
TAU1NZ  KA=U2/U1
        GO TO KTEST
NTRIC   MU=U2**2-U1*U3
        KA=(U1*U2-U0*U3)/2*NU
        W=KA**2-MU/NU
        IF ABS(W)-THRESH.LT.0 GO TO KTEST
        IF W.LT.0 GO TO COMPW
        W=SQRT(W)
        RM=KA-W
        RP=KA+W
        BM=1-RM**2
        BP=1-RP**2
        IF ABS(RM).GE..87 .OR. ABS(RP).GE..87 GO TO ALTRP
        DM=2(3#)**-.5*(U0*RP-U1)/BM**2
        TM=DM/W
        DP=2(3#)**-.5*(U1-RM*U0)/BP**2
        TP=DP/W
        IF ABS(TM).GE MAXBRT OR ABS(TP).GE MAXBRT GO TO ALTRP
        GO TO TTEST
ALTRP   TM=0
        RP=RM=U1/U0
        IF ABS(RP).GE.0.92 GO TO DISA
        TP=U0/((1-RP**2)**2*SQ3P4)
        IF ABS(TP).GE.MAXBRT GO TO DISA
        GO TO TTEST
KTEST   IF KA.GE.1 GO TO DISA
        RM=RP=KA
        TM=TP=U0*2/SQRT(3*PI)/(1-KA**2)**2
TTEST   EDJS=TM+TP
        IF ABS(TM+TP).GT.MAXBRT GO TO ALTRP
        R=(ABS(TM)*RM+ABS(TP)*RP)/(ABS(TM)+ABS(TP))
        IF EDJS<>0 GO TO CRIT
        EDJS=-EDJS
        CX=-CX
        CY=-CY
        R=-R
```

```
CRIT    GO TO EXIT
COMPH   R=KA
        EDJS=0
EXIT    FLAG=-1
        RETURN
DISA    FLAG=0
        RETURN
```

References

1. B. Bullock, "The performance of edge operators on images with texture." Hughes Aircraft Company Technical Report, October 1974.

2. M. H. Hueckel, "An operator which locates edges in digitized pictures," J. ACM, Vol. 18, pp 113-125, January 1971.

3. M. H. Hueckel, "A local visual operator which recognizes edges and lines," J. ACM, Vol. 20, pp 634-647, October 1973.

4. J. W. Modestino and R. W. Fries, "Edge detection in noisy images using recursive digital filtering," Computer Graphics and Image Processing, to be published.

5. E. Sims, N. Corby, L. A. Gerhardt, and J. W. Modestino, "An interactive system for digital image processing and pattern recognition," Proceedings fifth symposium on Imagery in Pattern Recognition, College Park, Maryland, April 1975.

6. G. P. Ashkar and J. W. Modestino, "The contour extraction problem with biomedical applications," Computer Graphics and Image Processing, to be published.

7. A. V. Oppenheim and R. W. Schafer, Digital Signal Processing, Englewood Cliffs, New Jersey: Prentice-Hall, 1975.

8. L. R. Rabiner and B. Gold, Theory and Application of Digital Signal Processing, Englewood Cliffs, New Jersey: Prentice-Hall, 1975.

9. R. W. Fries, "Edge detection in noisy images using recursive digital filtering," M.S. Thesis, Department of Electrical and Systems Engineering, R.P.I., Troy, New York, Nov. 1975.

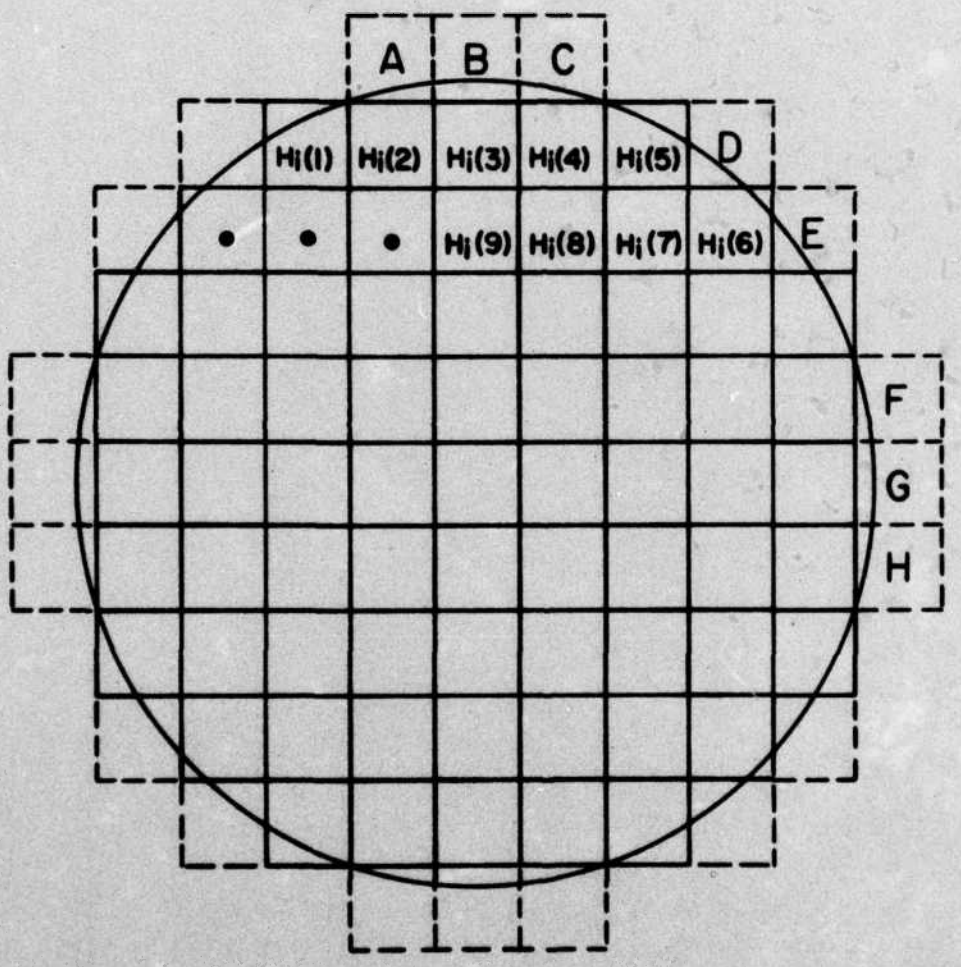
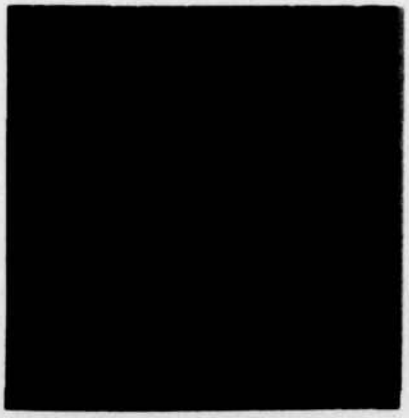


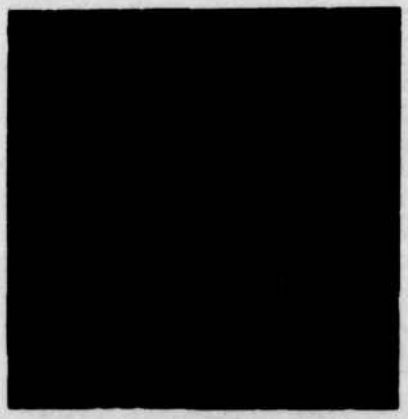
Fig. 1

Scheme for determining discretized basis functions.

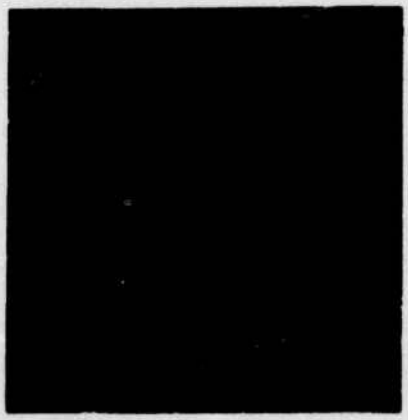
a) Original Image



b) Laplacian



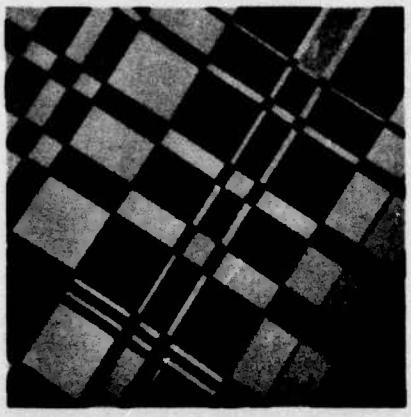
c) Wiener Operator



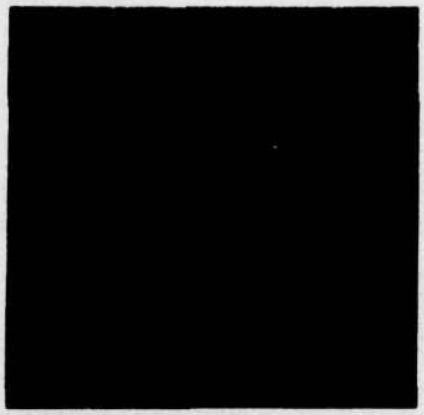
d) Hueckel Operator

Fig. 2

Perfcrmance of Edge Detectors on Random Field for λ=.025, ρ=-.9, ζ=∞ : (a) Original image; (b) Laplacian; (c) Wiener operator for F=.3 and T=125 ; (d) Hueckel Operator for CONF=.75 and DIFF=1.



a) Original Image



b) Laplacian

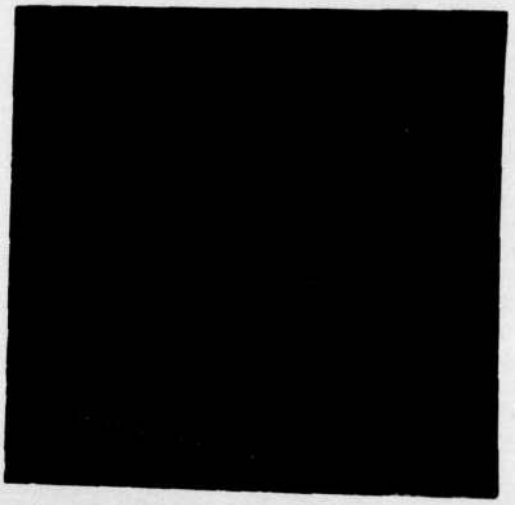
c) Wiener Operator

d) Hueckel Operator

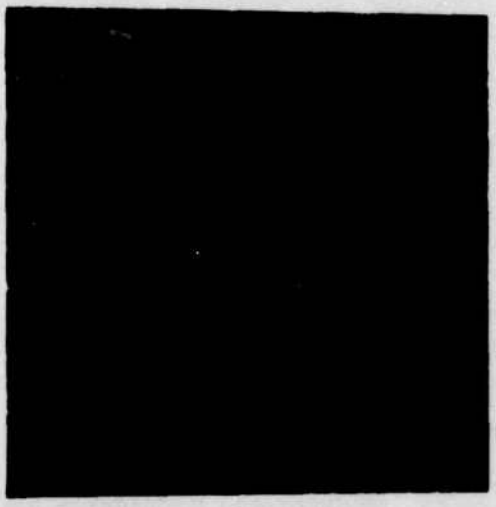
Fig. 3

Performance of Edge Detectors on Random Field for λ=.05, ρ=-.9, ζ=∞: (a) Original image; (b) Laplacian; (c) Wiener operator for F=.3 and T=125; (d) Hueckel operator for CONF=.75 and DIFF=1.

a) Original Image



b) Laplacian

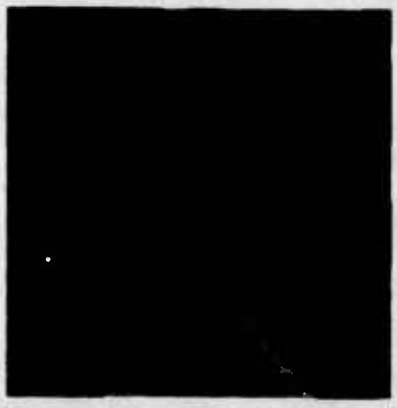


c) Wiener Operator

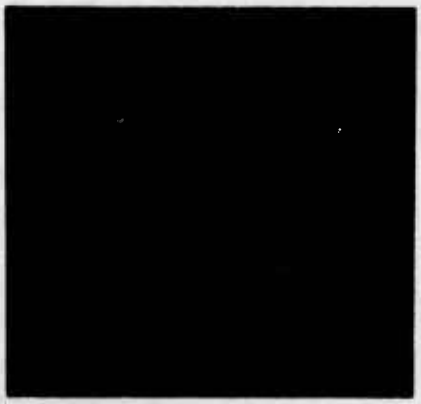
d) Hueckel Operator

Fig. 4

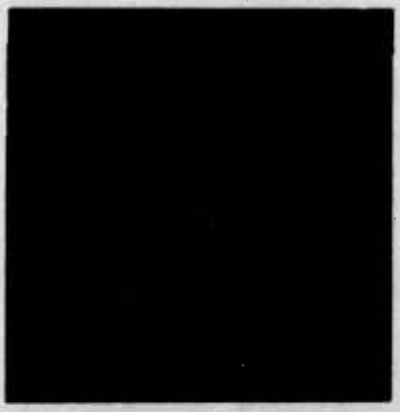
Performance of Edge Detectors on Random Field for λ=.05, ρ=.5, ζ=∞: (a) Original image; (b) Laplacian; (c) Wiener operator for F=.9 and T=3.13; (d) Hueckel operator for CONF=.75 and DIFF= 1.



a) Original Image



b) Laplacian

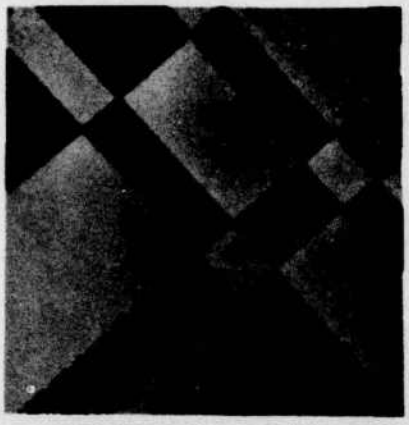


c) Wiener Operator

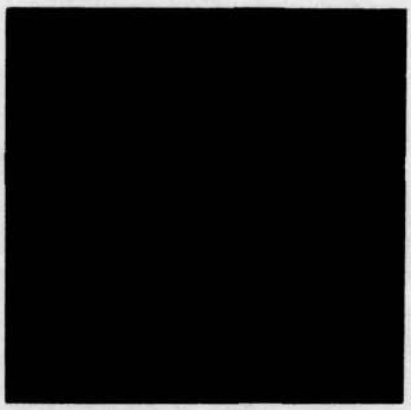
d) Hueckel Operator

Fig. 5

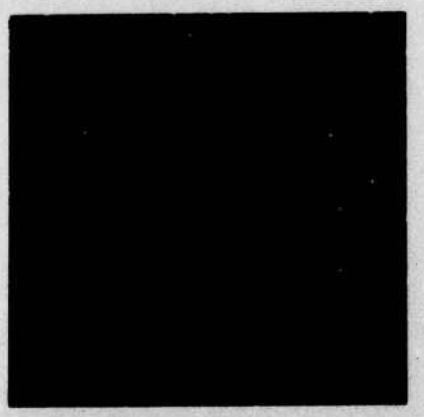
Performance of Edge Detection on Random Field for λ=.0125, ρ=-.9, λ=10dB: (a) Original image; (b) Laplacian; (c) Wiener operator for F=.5 and T= 344; (d) Hueckel operator for CONF=.95 and DIFF= 9.



a) Original Image



b) Laplacian



c) Wiener Operator

d) Hueckel Operator

Fig. 6

Performance of Edge Detectors on Random Field for λ=.0125, ρ=-.9, ζ=3dB: (a) Original image; (b) Laplacian; (c) Wiener operator for F=.25 and T= 235; (d) Hueckel operator for CONF=.95 and DIFF= 9.

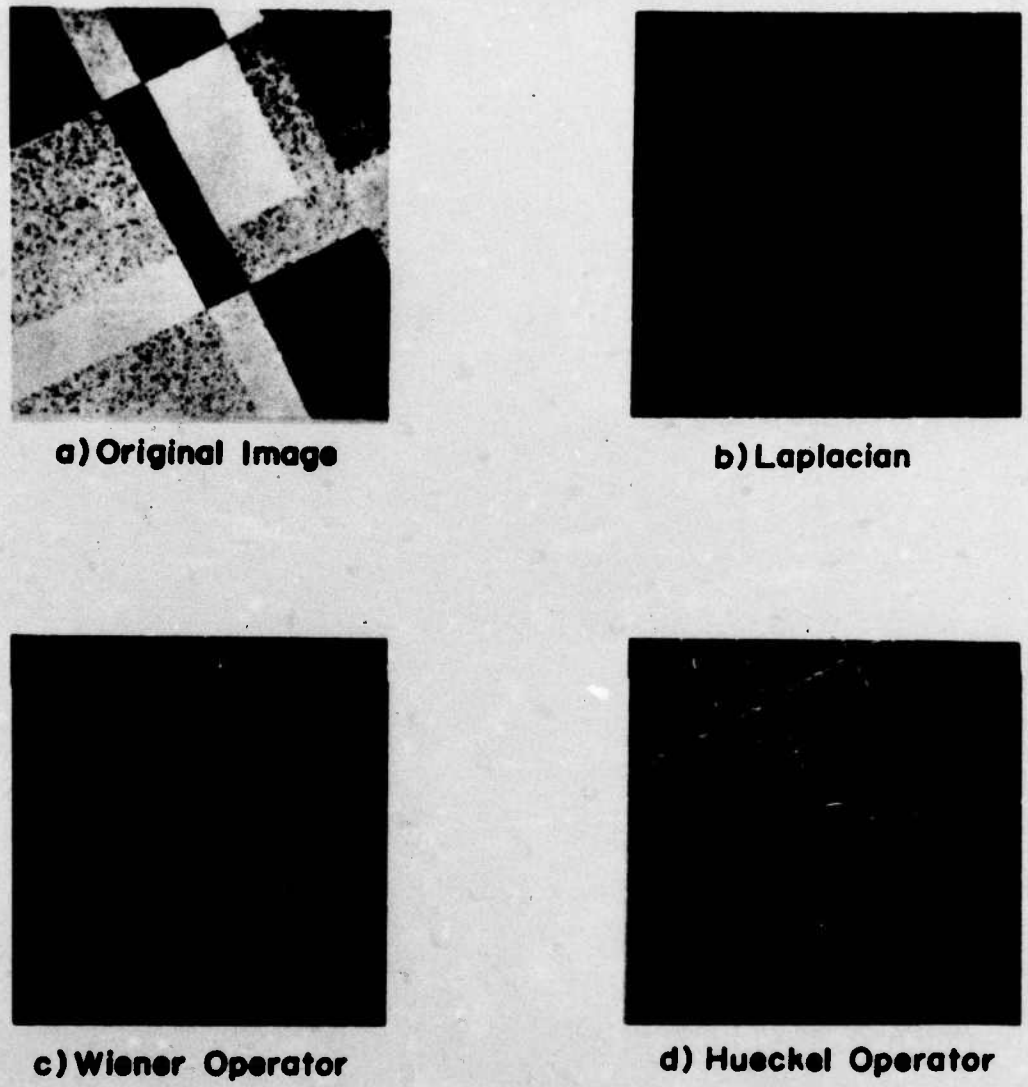


Fig. 7

Performance of Edge Detection on Random Field for λ=.0125, ρ=.5, ζ=3dB: (a) Original image; (b) Laplacian; (c) Wiener operator for F=.7 and T= 9.4; (d) Hueckel operator for CONF=.75 and DIFF= 5.

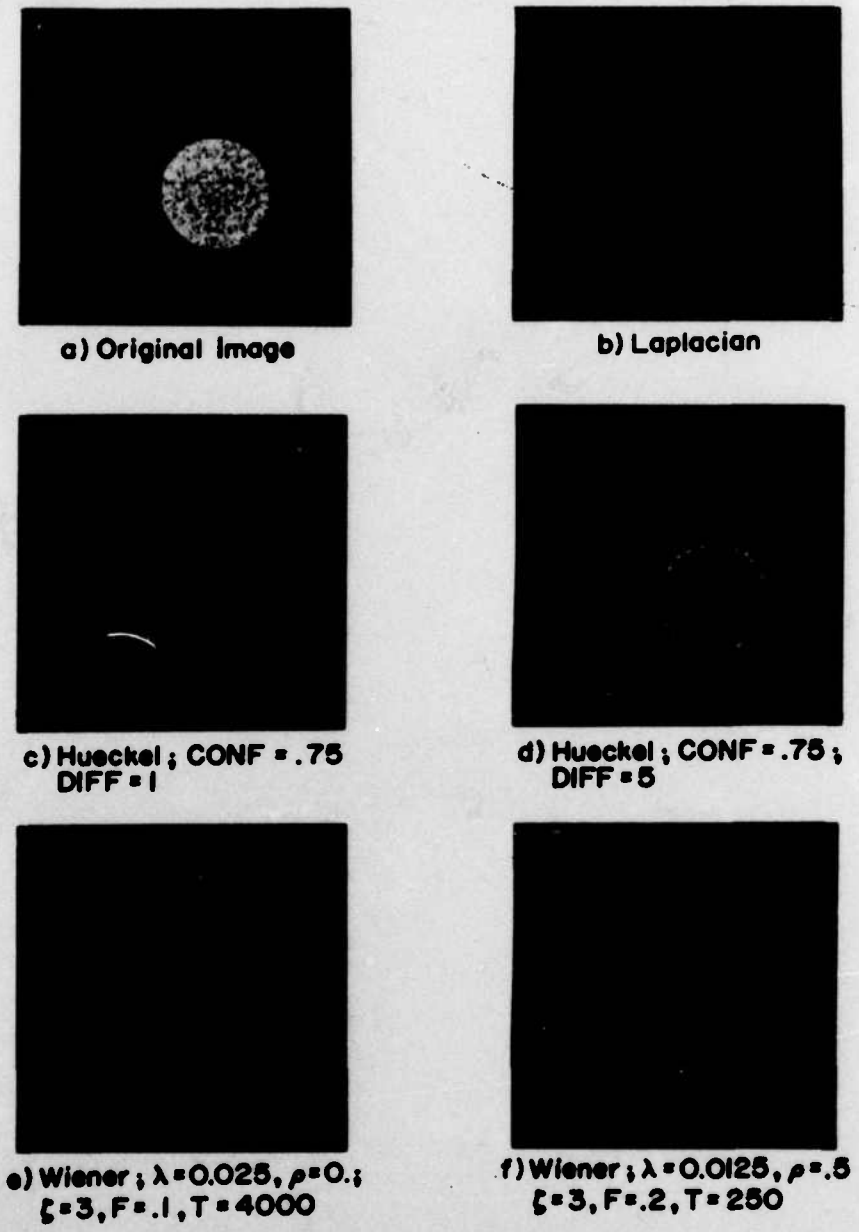


Fig. 8

Performance of Edge Detectors on Synthetic Image: (a) Original image; (b) Laplacian; (c) Hueckel operator for CONF=.75 and DIFF= 1; (d) Hueckel operator for CONF=.75 and DIFF= 5; (e) Wiener operator for λ=.025, ρ=0, ζ=3dB, F=.1 and T=625; (f) Wiener operator for λ=.0125, ρ=.5, ζ=3dB, F=.2 and T= 3.91.

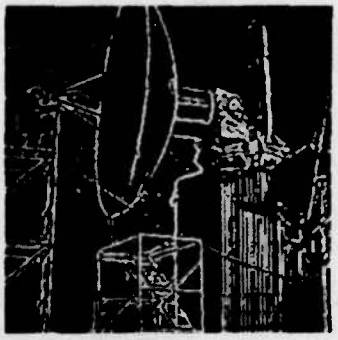
Fig. 9

Performance of Edge Detectors on Typical Head and Shoulder Image: (a) Original image; (b) Laplacian; (c) Hueckel operator for CONF=.75 and DIFF= 1; (d) Hueckel operator for CONF=.85 and DIFF= 6; (e) Wiener operator for λ=.025, ρ=0, ζ=3dB, F=.5 and T=1.56; (f) Wiener operator for λ=.0125, ρ=.5, ζ=3dB, F=.2 and T=.08.

Fig. 10

Performance of Edge Detectors on Typical Chest X-ray: (a) Original image; (b) Laplacian; (c) Hueckel operator for CONF=.75 and DIFF= 1; (d) Hueckel operator for CONF=.85 and DIFF= 6; (e) Wiener operator for λ=.05, ρ=-.9, ζ=10dB, F=.5 and T=1.56; (f) Wiener operator for λ=.0125, ρ=.5, ζ=3dB, F=.2 and T=.08.

a) Original Image



b) Laplacian

c) Hueckel - Fine Detail

d) Hueckel - Coarse Detail

e) Wiener - Fine Detail

f) Wiener - Coarse Detail

Fig. 11

Performance of Edge Detectors on Outdoor Scene 1: (a) Original; (b) Laplacian; (c) Hueckel operator for CONF=.75 and DIFF= 1; (d) Hueckel operator for CONF=.85 and DIFF=6; (e) Wiener operator for λ=.05, ρ=-.9, ζ=10dB, F=.3 and T= 125; (f) Wiener operator for λ=.0125, ρ=.5, ζ=3dB, F=.2 and T= 3.91.

a) Original Image

b) Laplacian

c) Hueckel - Fine Detail

d) Hueckel - Coarse Detail

e) Wiener - Fine Detail

f) Wiener - Coarse Detail

Fig. 12

Performance of Edge Detection on Outdoor Scene 2: (2) Original; (b) Laplacian; (c) Hueckel operator for CONF=.75 and DIFF= 1; (d) Hueckel operator for CONF=.85 and DIFF= 6; (e) Wiener operator for λ=.05, ρ=-.9, ζ=10dB, F=.3 and T= 125; (f) Wiener operator for λ=.0125, ρ=.5, ζ=3dB, F=.2 and T=3.91.

DISTRIBUTION LIST FOR ONR ELECTRONIC
AND SOLID STATE SCIENCES

Director Advanced Research Projects Agency Attn: Technical Library 1400 Wilson Boulevard Arlington, Virginia 22209	1 copy
Office of Naval Research Electronics Program Office (Code 427) 800 North Quincy Street Arlington, Virginia 22217	1 copy
Office of Naval Research Code 105 800 North Quincy Street Arlington, Virginia 22217	6 copies
Director Naval Research Laboratory 455 Overlook Avenue, S.W. Washington, D. C. 20375	
Attn: Technical Library	6 copies
Code 5200	1 copy
5210	1 copy
5270	1 copy
6400	1 copy
Office of the Director of Defense Research and Engineering Information Office Library Branch The Pentagon Washington, D. C. 20301	1 copy
Defense Documentation Center Cameron Station Alexandria, Virginia 22314	12 copies
Commanding Officer Office of Naval Research Branch Office 536 South Clark Street Chicago, Illinois 60605	1 copy
San Francisco Area Office Office of Naval Research 50 Fell Street San Francisco, California 94102	1 copy

Commanding Officer Office of Naval Research Branch Office 1030 East Green Street Pasadena, California 91101	1 copy
Commanding Officer Office of Naval Research Branch Office 495 Summer Street Boston, Massachusetts 02210	1 copy
New York Area Office Office of Naval Research 715 Broadway, 5th Floor New York, New York 10003	1 copy
ODDR&E Advisory Group on Electron Devices 201 Varick Street New York, New York 10014	1 copy
Naval Air Development Center Attn: Technical Library Johnsville Warminster, Pennsylvania 18974	1 copy
Naval Weapons Center China Lake, California 93555 Attn: Technical Library Code 6010	 1 copy 1 copy
Naval Research Laboratory Underwater Sound Reference Division Technical Library P. O. Box 8337 Orlando, Florida 32806	1 copy
Navy Underwater Sound Laboratory Technical Library Fort Trumbull New London, Connecticut 06320	1 copy
Commandant, Marine Corps Scientific Advisor (Code AX) Washington, D. C. 20380	1 copy
Naval Ordinance Station Technical Library Indian Head, Maryland 20640	1 copy
Naval Postgraduate School Monterey, California 93940 Attn: Technical Library Electrical Engineering Department	 1 copy 1 copy

Naval Missile Center Technical Library (Code 5632.2) Point Mugu, California 93010	1 copy
Naval Electronics Laboratory Center San Diego, California	
Attn: Technical Library	1 copy
Code 2300	1 copy
2600	1 copy
4800	1 copy
Naval Undersea Warfare Center Technical Library 3202 East Foothill Boulevard Pasadena, California 91107	1 copy
Naval Weapons Laboratory Technical Library Dahlgren, Virginia 22448	1 copy
Naval Ship Research and Development Center Central Library Code L42 and L43) Washington, D. C. 20007	1 copy
Naval Surface Weapons Center White Oak Laboratory Silver Spring, Maryland 20910	
Attn: Technical Library	1 copy
Code 200	1 copy
212	1 copy
Deputy Chief of Naval Operations (Development) Technical Analysis and Advisory Group (Code NOP-077D) Washington, D. C. 20350	1 copy
Commander Naval Air Systems Command Washington, D. C.	
Attn: Code 310	1 copy
360	1 copy
Commander Naval Electronics Systems Command Washington, D. C. 20360	
Attn: Code 304	1 copy
310	1 copy
Commander Naval Sea Systems Command Washington, D. C. 20360	1 copy
Naval Surface Weapons Center Attn: Library Dahlgren, Virginia 22448	1 copy

Air Force Office of Scientific Research — 1 copy
Attn: Electronic and Solid State Sciences Division
Department of the Air Force
Washington, D. C. 20333

Air Force Weapon Laboratory — 1 copy
Technical Library
Kirtland Air Force Base
Albuquerque, New Mexico 87117

Air Force Avionics Laboratory — 1 copy
Air Force Systems Command
Technical Library
Wright-Patterson Air Force Base
Dayton, Ohio 45433

Air Force Cambridge Research Laboratory — 1 copy
L. G. Hanscom Field
Technical Library
Cambridge, Massachusetts 02138

Harry Diamond Laboratories — 1 copy
Technical Library
Connecticut Avenue at Van Ness, N. W.
Washington, D. C. 20438

U. S. Army Research Office — 1 copy
Box CM, Duke Station
Durham, North Carolina 27706

Director — 1 copy
U. S. Army Engineering Research
and Development Laboratories
Fort Belvoir, Virginia 22060
Attn: Technical Documents Center

Director National Bureau of Standards — 1 copy
Attn: Technical Library
Washington, D. C. 20234

Superintendent — 1 copy
Materials Sciences Division
Naval Research Laboratory
4555 Overlook Avenue S.W.
Washington, D.C. 20375

SUPPLEMENTAL LIST FOR SYSTEMS AREA

Office of Naval Research — 2 copies
800 N. Quincy Street
Arlington, Virginia 22217
Attn: Code 430

Naval Research Laboratory — 1 copy
4555 Overlook Avenue, S. W.
Washington, D. C. 20375
Attn: Code 5400

Naval Electronics Laboratory Center
San Diego, California 92152
Attn: Code 3000 — 1 copy
5000 — 1 copy
5600 — 1 copy

Air Force Office of Scientific Research — 1 copy
Mathematical and Information Sciences Directorate
1400 Wilson Blvd.
Washington, D. C. 20333

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE	READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER: TR-77-1	2. GOVT ACCESSION NO. / 3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle): AN EMPIRICAL STUDY OF SELECTED APPROACHES TO THE DETECTION OF EDGES IN NOISY DIGITIZED IMAGES	5. TYPE OF REPORT & PERIOD COVERED: Technical Report
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): R. W. Fries and J. W. Modestino	8. CONTRACT OR GRANT NUMBER(s): N00014-75-C-0281
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Electrical & Systems Engineering Dept., School of Engineering, Rensselaer Polytechnic Institute, Troy, NY 12181	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: NR 375-971
11. CONTROLLING OFFICE NAME AND ADDRESS: Dept. of the Navy, Office of Naval Research, Electronic & Solid State Sciences Program, Arlington, VA 22217	12. REPORT DATE: March 1977
	13. NUMBER OF PAGES: 38
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office): 43p.	15. SECURITY CLASS. (of this report)
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Distribution unlimited; approved for public release

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Image Processing
Edge Detection
Two-Dimensional Filtering

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This paper is concerned with an empirical evaluation of the relative performance of several selected edge detectors for the detection of edge structure in noisy digitized images. In particular, we consider both the Hueckel operator and a recently introduced class of edge detectors implemented as two-dimensional infinite inpulse response (IIR) or recursive digital filters. The latter were originally developed on the basis of least mean-square Wiener spatial filtering concepts for an assumed stochastic model of edge structure in typical imagery

next page

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE S/N 0102-014-6601

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

401653

cont →

data. These two schemes are compared with respect to performance, computational complexity and suitability for specific applications. Problems in making these comparisons are discussed. The implementations of both operators used for this comparison are described in detail.


ATE
ME