AD-A084 733   ARMY ARMAMENT RESEARCH AND DEVELOPMENT COMMAND DOVER--ETC F/G 9/2
VIDEO SIGNAL PROCESSING FOR IMAGE ENHANCEMENT: A DEMONSTRATION --ETC(U)
JAN 80 G SIVAK
UNCLASSIFIED ARSCD-TR-79013 SBIE -AD-E400 399 NL

1 OF 1
AD A
084733

END
DATE
FILMED
6-80
DTIC

5.c

LEVEL III

(12)

AD

AD-E400 399

ADA084733

TEHNICAL REPORT ARSCD-TR-79013

VIDEO SIGNAL PROCESSING FOR IMAGE ENHANCMENT: A DEMONSTRATION MODEL

GARY SIVAK

JANUARY 1980

US ARMY ARMAMENT RESEARCH AND DEVELOPMENT COMMAND
FIRE CONTROL AND SMALL CALIBER
WEAPON SYSTEMS LABORATORY
DOVER, NEW JERSEY

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER Technical Report ARSCD-TR-79013	2. GOVT ACCESSION NO. AD-A084733	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Video Signal Processing for Image Enhancement: A Demonstration Model		5. TYPE OF REPORT & PERIOD COVERED Final
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Gary Sivak		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS ARRADCOM, FC&SCWSL Fire Control Division (DRDAR-SCF-I) Dover, NJ 07801		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS Project 12161101A011A
11. CONTROLLING OFFICE NAME AND ADDRESS ARRADCOM, TSD STINFO (DRDAR-TSS) Dover, NJ 07801		12. REPORT DATE January 1980
		13. NUMBER OF PAGES 55
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

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

18. SUPPLEMENTARY NOTES

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

Haar transform
Fast Haar transform
Digital image processing
Digital image enhancement
Video signal processing
Transformations (mathematics)

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

This report presents the development of a digital image enhancement demonstration model, an interactive computer program which will display the results of any of a number of digital image enhancement algorithms. The present version of the program includes Haar, Hadamard, and Fourier transform algorithms in modular, subroutine form. Other algorithms can be added on for testing on the display target, a 16-by-16 matrix of picture elements representing an image of the side view of a tank. The target,

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

20. ABSTRACT

before and after application of the given algorithm, is graphically displayed along with signal strengths, RMS noise levels, tank component edge contrasts, and signal-to-noise ratios.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

TABLE OF CONTENTS

FIGURES

TABLES

INTRODUCTION

Fire Control system needs for increased probabilities of detection, recognition, and identification require the improvement of tactical target edge and feature contrast, and signal-to-noise ratio, i.e., image enhancement. Since optical techniques can provide only limited algorithmic flexibility and image enhancement, more powerful digital techniques such as spectral filtering must be tested.

The work, begun in 1975, relating to the development of a digital image enhancement interactive demonstration program is presented in this paper. Originally existing on the West Point U.S. Military Academic computer, it now exists on the U.S. Army ARRADCOM CDC 6500 time-sharing system.

The motivation for this effort lies in work done 2 years previously by a team of investigators at Northrup Research Corporation in Anaheim, California. The researchers did obtain some image enhancement by using a hard-wire Haar transform processor for contrast improvement. Since these results were inconclusive, the decision was made to carry out an in-house, laboratory, independent research effort with the objective of using digital image enhancement techniques for contrast and signal-to-noise improvement as an aid in target detection, tracking, and weapon direction. The currently existing routine, Program IMAGE, is an outgrowth of this effort.

First, some mathematical background is presented in matrix theory, Fourier analysis, and Hadamard matrices. Next, the steps taken in the development of Program IMAGE and an explanation of the roles of and tasks performed by the various modular subroutines are discussed. Finally, several examples of actual image enhancement computer outputs are presented and explained.

MATHEMATICAL BACKGROUND

Matrix Manipulation

Arbitrary Linear Gray-Scale Transformations

Gray-level scaling, the capacity to reset the digitized values of the individual picture elements of an image to within new upper and lower bounds, is important for two reasons. First, after algorithmic processing, it may be important to rescale the output data to put the bounds within meaningful limits for easy interpretation, spotting of anomalous results, and the generation of histograms that can be quickly and easily compared with those of the original input data. Second, it is important to place the lower bound of image processing results just above the minimum level which is available on a digital to analog video display screen, to prevent "clipping" of the minimum values. Similarly, the upper data bound should be placed below the maximum gray level setting. This prevents the "whiting" or "washing out" of the results, or even worse, the loss of valuable picture information.

Problem: Given that picture element values range between A and B, rescale them to range between the values C and D. That is, for $A \leqq X \leqq B$, find the linear transformation function, $y(x)$, such that $C \leqq y \leqq D$.

and:

$$y(A) = C$$

$$y(B) = D \tag{1}$$

From the familiar point-slope equation for a linear function, for a given point (x_1, y_1) and any point (x, y) one has:

$$y - y_1 = M(x - x_1) \tag{2}$$

$$M = \frac{D - C}{B - A} \tag{3}$$

Take $x_1 = A$, and $y_1 = C$. If one took $x_1 = B$, and $y_1 = D$, the same end result would be achieved. Substituting for x_1 and y_1 into Equation 2,

$$y - C = \frac{D - C}{B - A} \cdot (x - A) \tag{4}$$

Adding C to both sides under the denominator B - A and rearranging terms slightly results in the arbitrary linear gray-scale transformation equation,

$$y = \frac{D - C}{B - A} \cdot x - \frac{A \cdot D - BC}{B - A} \tag{5}$$

By this equation, pixel data, ranging from values A to B, are linearly transformed to a set of values ranging from C to D.

As a simple example, one can calculate the transformation scaling gray-level values ranging from 0 to 1 into gray-level values ranging from 0 to 100. In this case, substitute into Equation 5:

$$A = 0 \quad B = 1 \quad C = 0 \quad D = 100 \tag{6}$$

As one might expect, in this instance, Equation 5 simply reduces to:

$$y = 100x. \tag{6}$$

Original Data Scan and Commutativity

As mentioned in the introduction, motivation for this work was derived from the Northrup Corporation, specifically, from a one-dimensional Haar transform scan that they used. The scan forms a transform image whose rows are formed by the operation of the rows of the Haar matrix T, on the rows of the image matrix I. This is in contrast to standard matrix multiplication where the rows of the product matrix are formed by operating with the rows of the transformed matrix on the columns of the matrix of the image to be transformed and/or enhanced. With a simplified one-subscript notation for easier reading, one can, for example, multiply the 2-by-2 Haar transform matrix T, by the rows of the image matrix I. To obtain the function F(T, I), a formal expression for the representation of the image transform, one has:

$$T = \begin{bmatrix} T_1 & T_2 \\ T_3 & T_4 \end{bmatrix} \qquad I = \begin{bmatrix} I_1 & I_2 \\ I_3 & I_4 \end{bmatrix} \tag{7}$$

$$F(T, I) = T \cdot I^T \qquad (8)$$

where I^T is the transpose of the image matrix I. This is required so that the rows of the transform matrix T multiply the rows of the image matrix I, and not the columns. The result, for the one-dimensional horizontal scan is:

$$F(T,I) = \begin{bmatrix} T_1 I_1 + T_2 I_2 & T_1 I_3 + T_2 I_4 \\ T_3 I_1 + T_4 I_2 & T_3 I_3 + I_4 I_4 \end{bmatrix} \qquad (9)$$

In matrix manipulation, the question arises as to when the product of two matrices is commutative, i.e., when A B = B A, for two N-by-N matrices A and B. For the 2-by-2 case:

$$A = \begin{bmatrix} A_1 & A_2 \\ A_3 & A_4 \end{bmatrix}$$

$$B = \begin{bmatrix} B_1 & B_2 \\ B_3 & B_4 \end{bmatrix} \qquad (10)$$

Take the first component of the two product matrices, for example:

$$A_1 \; B_1 + A_2 \; B_3 = B_1 \; A_1 + B_2 \; A_3$$

or (11)

$$A_2 \; B_3 = A_3 \; B_2$$

One sees, therefore, by comparing with other components that $A \cdot B = B \cdot A$ when the matrices are symmetric.

Two-dimensional Scanning and the Role of the Multipliers

Given an image matrix I, a transform matrix T, an inverse transform matrix T^{-1}, an output matrix O, and a storage matrix X, one has the relationships (ref. 1):

$$X = T \cdot I \cdot T^T$$

$$O = T^{-1} \cdot X \cdot (T^{-1})^T \qquad (12)$$

The Haar matrix T, in the 4-by-4 case, contains three M values or enhancement factors (ref. 2). It is of the form:

$$T \approx \begin{bmatrix} M_1 & M_1 & M_1 & M_1 \\ M_2 & M_2 & M_2 & M_2 \\ M_3 & M_3 & M_3 & M_3 \\ M_3 & M_3 & M_3 & M_3 \end{bmatrix} \quad (13)$$

The matrix T^T, is proportional to a matrix whose columns are identical to the rows of the one in Equation 13. The matrix of the image is of the form:

$$I = \begin{bmatrix} I_{11} & I_{12} & I_{13} & I_{14} \\ I_{21} & I_{22} & I_{23} & I_{24} \\ I_{31} & I_{32} & I_{33} & I_{34} \\ I_{41} & I_{42} & I_{43} & I_{44} \end{bmatrix} \quad (14)$$

To see how the M's behave under transformation, note that the storage matrix X is proportional to $T \cdot I \cdot T^T$.

Performing the product $I \cdot T^T$ yields for:

$$\begin{aligned} S_1 &= I_{11} + I_{12} + I_{13} + I_{14} \\ S_2 &= I_{21} + I_{22} + I_{23} + I_{24} \\ S_3 &= I_{31} + I_{32} + I_{33} + I_{34} \\ S_4 &= I_{41} + I_{42} + I_{43} + I_{44} \end{aligned} \quad (15a)$$

$$I \cdot T^T \approx \begin{bmatrix} S_1M_1 & S_1M_2 & S_1M_3 & S_1M_3 \\ S_2M_1 & S_2M_2 & S_2M_3 & S_2M_3 \\ S_3M_1 & S_3M_2 & S_3M_3 & S_3M_3 \\ S_4M_1 & S_4M_2 & S_4M_3 & S_4M_3 \end{bmatrix} \quad (15b)$$

Continuing the procedure and multiplying the matrix expression in equation 15b by the matrix T on the left results in

$$X \approx F \cdot M \qquad (16)$$

where F depends on the sum of the image data values,

$$F=I_{11}+I_{12}+I_{13}+I_{14}+I_{21}+I_{22}+I_{23}+I_{24}+I_{31}+I_{32}+I_{33}+I_{34}+I_{41}+I_{42}+I_{43}+I_{44} \qquad (17)$$

and M is a matrix

$$M = \begin{bmatrix} M_1^2 & M_1\ M_2 & M_1\ M_3 & M_1\ M_3 \\ M_1\ M_2 & M_2^2 & M_2\ M_3 & M_2\ M_3 \\ M_1\ M_3 & M_2\ M_3 & M_3^2 & M_3^2 \\ M_1\ M_3 & M_2\ M_3 & M_3^2 & M_3^2 \end{bmatrix} \qquad (18)$$

Thus, two-dimensional image transform matrices are quadratically dependent upon the enhancement factors in a symmetric matrix format, with the main diagonal being quadratic in terms of purely one kind.

Fourier Transform Background

Summation and Matrix Equivalency

A given N-by-N digital image matrix has picture element entries f_{xy} where the indices x and y indicate the row and column positions and are taken to range from 0 to N-1. Its Fourier transform is the N-by-N matrix with elements F_{uv}, and is given by the familiar two-dimensional Fourier transform expression:

$$F_{uv} = \sum_{X=0}^{N-1} \sum_{Y=0}^{N-1} f_{xy} \exp\left(\frac{-2\pi i}{N} (ux + vy)\right) \qquad (19)$$

For the matrix:

$$\begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & w^{-1} & w^{-2} & \cdots & w^{-(N-1)} \\ 1 & w^{-2} & w^{-4} & \cdots & w^{-2(N-1)} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & w^{-(N-1)} & w^{-2(N-1)} & \cdots & w^{-(N-1)^2} \end{bmatrix} \quad (20)$$

where

$$w = e^{2 \frac{\pi i}{N}} \quad (21)$$

One can write a matrix equation:

$$F = W \cdot f \cdot W \quad (22)$$

Problem: Show that Equations 19 and 22 are equivalent. Take, as an example, N = 3, and calculate the first element F_{00}. From Equation 19, it is evident that

$$F_{00} = f_{00}+f_{01}+f_{02}+f_{10}+f_{11}+f_{12}+f_{20}+f_{21}+f_{22} \quad (23)$$

In this case, from the definition of f, and Equation 20

$$F = \begin{bmatrix} f_{00} & f_{01} & f_{02} \\ f_{10} & f_{11} & f_{12} \\ f_{20} & f_{21} & f_{22} \end{bmatrix}$$

$$W = \begin{bmatrix} 1 & 1 & 1 \\ 1 & w^{-1} & w^{-2} \\ 1 & w^{-2} & w^{-4} \end{bmatrix} \quad (24)$$

To compute F_{00} by means of Equation 22, take the matrix $X = F \cdot W$ and compute the first column of X

$$x_{11} = f_{00} + f_{01} + f_{02}$$
$$x_{21} = f_{10} + f_{11} + f_{12} \qquad (25)$$
$$x_{31} = f_{20} + f_{21} + f_{22}$$

To obtain F_{00}, multiply X on the left by the first row of the Fourier transform matrix of W of Equation 20.

$$F_{00} = f_{00}+f_{01}+f_{02}+f_{10}+f_{11}+f_{12}+f_{20}+f_{21}+f_{22} \qquad (26)$$

This expression is identical to Equation 23. Thus, the summation and matrix notations are equivalent.

Delta Function and Normalization

Image enhancement can work because, from sampling theory, transforms can select out and increase the prominence of a given spatial frequency component of a signal. The Nth spatial frequency component of a signal is defined as N variations from a low to a high pixel value, over the data field length. The delta function has the property of being able to select out a given value. Therefore, the derivation of its integral expression will be reviewed and a corresponding summation expression derived for application to digital data sets.

For a function of time, f(t), and its Fourier transform , F(w),:

$$f(t) = \int F(w) e^{iwt} dw \qquad (27)$$

$$F(w) = \int f(\tau) e^{-iw\tau} d\tau \qquad (28)$$

Substituting Equation 28 into Equation 27 and reversing the order of integration because the functions are assumed to be continuous and integrable

$$f(t) = \int f(\tau) \left[\int dw e^{jw(T-\tau)} \right] d\tau \qquad (29)$$

If one writes

$$f(t) = \int f(\tau) \delta(T-\tau) d\tau \qquad (30)$$

In this case, the delta function filters out the value $f(t-\tau)$ to be input into $f(t)$. Then one has

$$\delta(t-\tau) = \int e^{iw(t-\tau)} d\tau \qquad (31)$$

This is the integral expression for the delta function. Similarly, for discrete digital functions where f_x is the digital function of the index x and F_μ is the Fourier transform equivalent

$$f_x = \sum_{\mu=0}^{N-1} F_\mu w^{\mu x} \qquad (32)$$

where $w = e^{2\pi \frac{i}{N}}$ as before. Also,

$$F_\mu = \sum_{xi=0}^{N-1} fxi \; w^{-\mu \xi xi} \qquad (33)$$

By substituting Equation 33 into Equation 32 and reversing the order of the summation indices:

$$f_x = \sum_{\xi=0}^{N-1} f_\xi \sum_{\mu=0}^{N-1} w^{\mu(x-\xi)} \qquad (34)$$

And, therefore, as in Equation 31:

$$\delta(x-\xi) = \sum_{\mu=0}^{N-1} w^{\mu(x-\xi)} \qquad (35)$$

The last point to consider, with regard to Fourier transforms, is the question of normalization for proper scaling, a point not always explicitly treated in the literature. As an example, take Equation 19, for the complex (real and imaginary), two-dimensional Fourier transform and reduce it to the one-dimensional case for the real part of the Fourier transform of a real function:

$$F_i = \sum_{j=0}^{N-1} fj \cos\left(\frac{2\pi ij}{N}\right) \qquad (36)$$

where, from sampling theory, the permissible values of i are, in general, 0 through $\frac{N}{2} - 1$.

Take, as an example, the case of N = 8. If the function of the Fourier transform that is being sought is one cycle of a cosine, then F_0, the DC term, is 0 and F_1, the weight of the fundamental cosine frequency is required to be equal to 1. Substituting $F_1 = 1$ into Equation 36 and dividing by a constant of proportionality $\frac{N}{s}$ for scaling yields

$$1 = \frac{\sum_{j=o}^{7} fj \cos\left(\frac{2 \pi j}{8}\right)}{\frac{8}{s}} \qquad (37)$$

Here, s is the unknown scaling factor and gives

$$fj = \cos\left(\frac{2 \pi j}{N}\right) \qquad (38)$$

Therefore,

$$1 = \frac{\sum_{j=0}^{7} \cos^2 \left(\frac{2 j}{8}\right)}{\frac{8}{s}} \qquad (39)$$

This equation yields:

$$s = 2. \qquad (40)$$

Thus, for proper scaling, the results of a one-dimensional Fourier transform must be divided by $\frac{N}{2}$ and, as can similarly be shown, those of an inverse Fourier transform must be divided by 2. They must share a common divisor of N. Similarly, the results of a two-dimensional Fourier transform must be divided by $\frac{N^2}{4}$ and its inverse by 4. Therefore, a common divisor of N^2 is shared.

Hadamard Transform Matrix

The well-known Hadamard matrices are based on the arrangement of Walsh functions. The first and simplest is the 2-by-2 case:

$$H_1 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \qquad (41)$$

Matrices of higher order are recursively generated or "grown" by the replacement of each element of the matrix in Equation 41 by the matrix H_1. This means, of course, that when the -1 is replaced by H_1, the signs, obviously, are simply reversed. The next Hadamard matrix, the 4-by-4 case is

$$H_2 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \tag{42}$$

Note that the matrix is symmetric and except for a factor of 4, H^{-1} is identical to H for any order. Here,

$$H_2^{-1} = \frac{1}{4} \cdot H_2 \tag{43}$$

This recursive process can be continued indefinitely. The N-by-N matrix is multiplied by the N element pixel vector, and the elements of the output column transform vector are adjusted. For example, by decreasing the magnitude of the first element, d.c. or background energy is suppressed. The inverse transform matrix is then applied, and the result is the output pixel vector.

PROGRAM IMAGE

Development of the Program

Two 16-by-16 image matrices for each printout page are used, the upper one for the low-contrast input image, the lower one for the improved contrast algorithm-processed output image.

The elements in the input and output target image of the simulated image matrices correspond to the brightness of points (or areas of points) of a larger actual TV image. The targets are characterized by a bordering background level and a central signal level outlined by a graphical drawing of a tank. The characterization of the tank target matrix is shown in figure 1. The letters G, C, T, H, W, and B show the various sections of the tank in the image field and indicate the gun, cupola, turret, hull, the wheels (or tread), and the background, respectively. (Appreciation

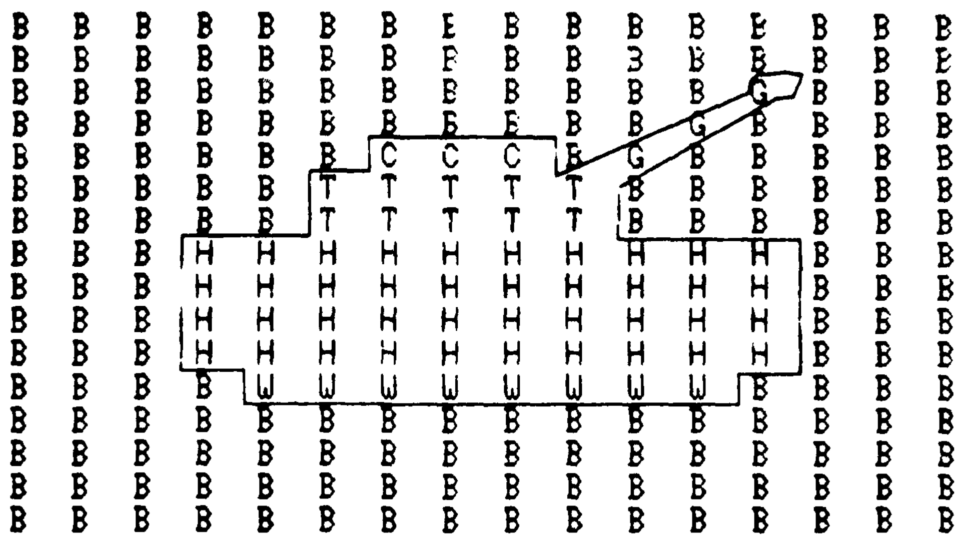

Figure 1. Tank target characterization.

is extended to a former co-worker Gerald Watson for his assistance in implementing the computer programming and the computer graphics required for a meaningful display of the output data.)

No matrix larger than the 16-by-16 case has been used in this effort for two reasons: First, the fast Haar and the fast Fourier transforms used algorithms that require the dimension N of the input data to be an integer power of two. Thus, the next larger utilizable square image matrix, the 32-by-32 case, contains 1024 entries and is too unwieldy to easily manipulate on paper, both for theoretical studies and numerical value checking. Also, the 16-by-16 image matrix is the largest one that can be fully displayed on the computer terminal because of constraints in the number of alphanumeric characters and spaces available per output line.

The chronological development of Program IMAGE is summarized in table 1. The first option programmed from equations supplied by the Northrup investigators, was the Haar 4-by-4 matrix that affords the capacity to vary three spatial frequencies for image enhancement, the highest being value changes between adjacent picture elements. After initial tests, another processor, a Haar 16-by-16 matrix, was added that affords greater flexibility via the capacity to manipulate two more spatial frequencies for image enhancement. Specifically, a d.c. term, and 1, 2, 4, and 8 cycles of square wave oscillation are available for the data field length of 16 points. For a more detailed discussion of this, see figure 3 and the relevant section on matrices in the referenced Haar transform report (ref. 2).

Only a background and signal level were initially displayed to get the software working as quickly as possible. Later a random noise-level option was added to give a more real-world situation whereby one could process TV pictures with moderate or severe noise levels incorporated into the data.

To chart how signal-to-noise ratios propagate through algorithms and to see if they can be improved, a subroutine was added to the program to calculate the RMS value of the processed picture patterns. Since transforms or other algorithms enhance or pull out contrast either horizontally across a line thus causing vertical edges to stand out, or down a vertical column making horizontal lines stand out, or in two dimensions, an option was added to calculate and print out contrasts for each section of the image in both

Table 1. Development of program IMAGE

Steps	Item
1	Establish computer formatting and programming.
2	4-by-4 Haar transform matrix, (Northrup Corp's suggestion).
3	16-by-16 Haar transform matrix.
4	Random noise capability.
5	Processed RMS value of image, capability.
6	Contrast calculations.
7	Average processed signal option.
8	Modularization.
9	Subroutine BEEP.
10	Corrections to RMS noise-level scaling.
11	Current status of nine available algorithms.

horizontal and vertical directions. Appreciation is extended to ARRADCOM engineer, Mr. Ray Popko, for his helpful suggestion of this idea.

A routine employing the 4-by-4 Hadamard transform in a horizontal scan was written and separately programmed. For easier access, it was added to the already existing Haar 4-by-4 matrix and the Haar 16-by-16 matrix routines, and combined into one file. In an effort to provide even more flexibility in characterizing and manipulating the spatial frequency components of image data for enhancement purposes, a Fourier transform test program was written to understand its numerical behavior in one- and two- dimensional scans.

The Fourier transform provides, for N points, $\frac{N}{2}$ spatial frequency values that can be weighted for enhancement. Upon debugging, Fourier transform and Fast Fourier transform routines were incorporated in horizontal and two-dimensional scans.

An option was then added to compute the average processed signal strength of the tank target in the data field for signal-to-noise measurement purposes.

Because of the increasing difficulty in modifying the programming, then stored on the West Point Honeywell computer, the decision was made to rewrite the entire program in terms of transform-computing and picture-processing modular subroutine packages that could simply be strung together. At this time, the cumbersome matrices and equations of the Haar 4-by-4 and 16-by-16 routines were replaced by the author's fast-Haar transform algorithm that requires no two-dimensional matrix arrays. This action reduced the bulk of the coding and reduced the storage area requirements of the program by 60 percent (ref. 2).

Lastly, a more detailed signal-to-noise analysis was conducted, resulting in the proper scaling of noise values such that their RMS value becomes equal to a desired value without error. This derivation is discussed in full detail in the section on RMS noise-level scaling in the referenced report on the application of the author's fast-Haar transform algorithm to real IR data (ref. 3).

Utilizing the Program

The divisions of the total main program/subroutine package and IMAGE program are outlined in table 2. The modular subroutine packages are arranged in the order in which they are called by the main program with Haar followed by Hadamard, followed by Fourier transform processing routines.

A summary of the current status of the program involving nine presently available algorithm options for scanning imagery data in horizontal, vertical, and two-dimensional formats is given in table 3.

The numerical data in the beginning of the program (see appendix A) define the coordinates of the lines comprising the outline of the tank target area. After this part, the program requires the answers to one question and two commands.

1. Do you want to test number sequences on your algorithm?

2. Enter background, signal, and noise level.

3. Specify which of the following algorithms by number.

Table 2. Division of program IMAGE

No.	Section	Function
1	Main program	Signal parameter input, alphanumeric output, graphics, calls to subroutines.
2	Subroutine BEEP	Prompts user when ready for input.
3	Subroutine APSIG	Calculates average processed signal level.
4	Subroutine PRMS	Calculates RMS value of processed image.
5	Subroutine CTRAST	Edge contrast of features of tank target.
6	Subroutine HAAR	Investigator's fast Haar transform algorithm.
7	Subroutine HAR16H	Effects horizontal scan using subroutine HAAR.
8	Subroutine HAR16V	Effects vertical scan using subroutine HAAR.
9	Subroutine HAR16HV	Effects horizontal then vertical scan, using Subroutine HAAR.
10	Subroutine HAR4H	Horizontal scan, in 4-pixel blocks, using Subroutine HAAR.
11	Subroutine HAD	4-pixel Hadamard transform.
12	Subroutine HAD4H	Horizontal scan in 4-pixel blocks, using Subroutine HAD.
13	Subroutine FT	Effects conventional, one-dimensional Fourier transform.
14	Subroutine FT16H	Horizontal scan using Subroutine FT.

Table 2. Division of program IMAGE (continued)

No.	Section	Function
15	Subroutine FT2D	Conventional two-dimensional Fourier transform.
16	Subroutine FT16D2	Effects two-dimensional Fourier transform of 16-by-16 image.
17	Subroutine FFT and REVERS	Implements one-dimensional Fast Fourier transform.
18	Subroutine FFTH	Effects horizontal scan processing by the Fast Fourier Transform/algorithm.
19	Subroutine FFTHV	Effects horizontal then vertical scan processing using the Fast Fourier Transform algorithm.

Table 3. IMAGE processing options

No.	Transform type	Processing procedure
1	Haar 16- element	Horizontal scan
2	Haar 16- element	Vertical scan
3	Haar 16- element	Horizontal then vertical scan
4	Haar 4- element	Horizontal scan
5	Hadamard 4-by-4	Horizontal scan
6	16- element Fourier	Horizontal scan
7	16-by-16 grid Fourier	Two-dimensional scan
8	16- element Fast Fourier	Horizontal scan
9	16- element Fast Fourier	Horizontal then vertical scan

If one replies "yes" to the "number sequence" option, the 16-by-16 input image matrix array is bypassed. One can then enter either a one-dimensional test signal vector for algorithms one and five, a 16-element input to the Haar transform algorithm or a 4-element input to the Hadamard transform routine, respectively. The resulting 16- or 4-element series is printed out after processing by enhancement weights that the user selects. If the number sequence option is not invoked, the user is required to furnish the background signal-and-noise level and a given transform. Suggested data selections that one can use to test for improved contrast with initial input contrast ranging from 5 to 50 percent, every 5 percent, are listed in table 4.

Similarly, if one is primarily interested in signal-to-noise analysis, suggested input parameters for input signal-to-noise ratios ranging from 1.5 to 6.0 every 0.5 unit are listed in table 5.

Upon assignment of the desired algorithm, the program will indicate, if needed, the enhancement weighted factors to be tested. As will be shown in the examples later, good enhancement will be achieved by the first algorithm, a 16-element Haar transform scan with, for example, the values 0.5, 1.0, 5.0, 10.0, and 15.0.

Table 4. Suggested data selections for contrast demonstrations

Background	Signal	Noise	Signal contrast (%)
19	20	0	5
9	10	0	10
17	20	0	15
8	10	0	20
9	12	0	25
14	20	0	30
13	20	0	35
12	20	0	40
11	20	0	45
10	20	0	50

Table 5. Suggested input parameter selections for signal-to-noise testing

Background	Signal	Noise	Signal-to-noise ratio
5	9	6	1.5
5	10	5	2.0
6	10	4	2.5
5	9	3	3.0
8	14	4	3.5
8	16	4	4.0
10	18	4	4.5
6	10	2	5.0
6	11	2	5.5
6	12	2	6.0

Tasks of the Modular Subroutines

Now, a brief discussion follows in order of the coding of the various modular subroutines and the tasks by which they are performed. Each algorithm is called from the main program by the parameter "INDEX" whose legal values are integers ranging from one through nine. With the addition by the user of more image processing algorithms to the program, this can easily be modified.

Subroutine BEEP

This routine prompts the user with a series of audible bells as to when the terminal is ready for data input. The three call parameters represent, in turn, the number of bells, the length of time (approximately in seconds) of each bell, and the length of the delay time (approximately in seconds) between each bell. The best results assume a CDC 6500 system tied through a SCOPE operating system to a Tektronix 4014 graphics terminal. Bell sounding and blank delay time characters may pass at different rates on other terminals. Also, on other computing machines and/or operating systems, the octal definitions of the bell and blank variables may not necessarily apply.

Subroutine APSIG

This subroutine calculates and prints out the average value of the strength of the processed signal. The pixel values for the tank target are simply added for the gun and cupola, turrent, hull, and wheels in four do loops. The average is then taken and the resulting value printed out.

Subroutine PRMS

This subroutine calculates the RMS value of the processed output image and is useful for testing noise level propagation under the influence of algorithms of interest. All of the pixel values over the entire processed output image field are squared and added, averaged, and then the square root is taken before print-out.

Subroutine CTRAST

This subroutine calculates the signal contrast of various parts of the tank, i.e., gun, cupola, turret, hull, tread, and finally an overall average value for the entire tank target image in both vertical and horizontal directions. The definition chosen for output signal contrast is:

$$C = \frac{S-b}{S} \tag{44}$$

Where S is the signal level of a tank pixel and B is the background value of a neighboring element not on the tank but in the surrounding background/clutter region. This is done for all occurrences of signal and background adjacent element pairs for each element of each part of the tank in both horizontal and vertical directions.

For example, take S to be the value of the 13th element from the left in the third row down, element (3, 13), one of the gun elements. Take b_1 to be the value of background element (3, 12) and b_2 to be the value of background element (3, 14), the elements to the left and right of the given gun element. The horizontal contrast at the (3, 13) element, the first element on the gun tip is taken to be:

$$C = \frac{\frac{S-b_1}{S} + \frac{S-b_2}{S}}{2} \tag{45}$$

for the horizontal contrast of the first gun element. This procedure is implemented for each section of the tank in both horizontal and vertical directions. The values are expressed as percentages for printout. The Hollerith headers GUNH, GUNV, TURH, TURV, etc., indicate the type of feature and contrast specified, for example, GUNH indicates gun horizontal contrast value.

Subroutine HAAR

This subroutine implements the author's fast Haar transform algorithm (ref. 2). The input string of pixels is entered by the user into the first N of 2N - 2 positions of the Q array, the log to the base 2 of N is supplied by the user to the variable NU, and the NU + one enhancement factors are read by the user into the E array. The N output enhanced-pixels are returned by the routine to the last N of the 2N - 2 positions of the Q array.

Haar Transform Subroutines

HAR16H, HAR16V, HAR16HV, and HAR4H subroutines employ subroutine HAAR to process the 16-by-16 input image. The first scans 16 elements at a time horizontally; the second, vertically; and the third, horizontally then vertically. The fourth breaks up each 16-element horizontal picture line into 4-element sections and invokes subroutine HAAR to process the lines, 4 elements at a time.

Subroutine HAD

This subroutine invokes processing by a 4-by-4 Hadamard matrix as defined earlier. Although the call parameters are expressed in general terms, in this case, Nu must be two so that 2^{nu} that equals N is equal to four. Four enhancement weighted factors E are specified, and four input picture elements to be processed are sent into the Q array.

Subroutine HAD4H

This subroutine invokes subroutine HAD to do Hadamard transform processing on the input image. Each horizontal input picture line is broken into four equal length sections and processing is done four elements at a time on the input image.

Subroutine FT

Where N is 2^{nu} for positive integral values of NU, this subroutine takes the Fourier transform of N elements and incorporates any alterations by $\frac{n}{2}$ enhancement factors present in the E array. The inverse Fourier transform is then taken, and the new pixel values returned to the real input array XR. Assume, for example, N = 8. Any given Fourier component F_i is of the form:

$$F_o = \sum_{j=0}^{7} f_j \operatorname{trig}\left(\frac{2\pi ij}{8}\right) \qquad (46)$$

where trig is a trigonometric function (sine or cosine) of the given argument. Ordinarily, to take the Fourier transform of N data elements, N^2 time-consuming trigonometric computations are required.

When, for example, $i = 3$ and $j = 7$, trigonometric functions of the argument $\frac{21\pi}{8}$ are required. However, the same answer is obtained because of periodicity by subtracting 2N or 16 from the argument and using $\frac{5\pi}{8}$ instead. Thus, the trigonometric values of the arguments $\frac{2\pi \cdot 1}{N}$ to $\frac{2\pi \cdot (N-1)}{N}$ can be computed once and stored in a "lookup" table. By simply subtracting integral multiples of N from the trigonometric arguments in the terms of the Fourier transform, the stored results in the lookup table can be used to select out the proper trigonometric values for any combination of the indices i and j.

Also, for N = 8, the Whitaker sampling theorem, allows four frequency components of a signal to be represented by this Fourier transform, specifically a dc term and 1, 2, and 3 cycles of sinusoidal oscillation for the data field length of 8 points. Just four frequencies are allowed because at least two samples are required to fully represent a signal frequency component. However, since the computed Fourier transform array xr contains 8 values, it was found by experimentation that:

xr_1 = dc term

xr_2 = strength of first frequency

xr_3 = strength of second frequency

xr_4 = strength of third frequency (47)

xr_5 = double strength of a 4-cycle frequency

$xr_6 = xr_4$

$xr_7 = xr_3$

$xr_8 = xr_2$

Thus, a Fourier transform of N points can handle the $\frac{N}{2}$ th frequency component if one remembers that it is erroneously reported at double strength. If one desires to modify the Fourier components of a signal by multiplying them by image enhancement weighted factors, the two elements in each pair, xr_2 and xr_8, xr_3 and xr_7, and finally xr_4 and xr_6 must be multiplied by the same factor. If the upper values are ignored, enhancement is incorrect.

Remember, before the application of the inverse Fourier transform, the results must be divided by $\frac{N}{2}$ and upon return to physical space after the inverse Fourier transform is applied, the results must be divided by two for proper scaling.

Subroutine FT16H

This subroutine invokes subroutine FT to do Fourier transform processing of the image data. Each 16-element horizontal picture line is processed in turn from the top to the bottom of the 16-by-16 input image matrix.

Subroutine FT2D

This subroutine takes the two-dimensional Fourier transform of the N-by-N real array XR, alters the spatial frequency components by the image enhancement weighted factors that the user furnishes, and then takes the inverse Fourier transform. The weighted factors are supplied to the $\frac{N}{2}$ length H and V arrays for image enhancement in both horizontal and vertical directions. In this conventional Fourier transform algorithm, trigonometric values are computed once as in subroutine FT so that the number of required trigonometric calculations is reduced from N^4 to N, thus saving computer time.

Subroutine FT16D2

This subroutine collects the input image data and applies subroutine FT2D to it to enhance the 16-by-16 image array via the two-dimensional Fourier transform algorithm.

Fast Fourier Transform

This particular algorithm was adapted and debugged for Fortran operation from an algorithm in the "focal" language. It was originally implemented by an investigator at the Digital Equipment Corp. in Maynard, Massachusetts (ref. 4). Whereas the original algorithm implemented the fast Fourier transform only, the author has modified it to incorporate image enhancement weighted factors to take the inverse fast Fourier transform and to scale the data upon return to pixel space. The user is required to furnish NU, that is, the log to the base 2 of the number of points N. Also, the array XR containing N pixel values to be modified and the array E containing the $\frac{N}{2}$ weighted factors are required. Note that subroutine REVERS is required by the fast Fourier algorithm to properly compute the trigonometric values.

Fast Fourier Transform Scans

Subroutine FFTH and FFTHV invoke the fast Fourier transform algorithm to operate on the input target image. The first scans the image horizontally from top to bottom. The second scans through the image in two directions, first across horizontal lines from top to bottom, then down vertical columns from left to right to operate on the data both horizontally and vertically.

Examples of Image Enhancement Demonstrations

Figures 2, 3, 4, and 5 are examples of actual image enhancement incorporating the input image array with background value 19, signal value 20, and rms noise level 0.

Figure 2 shows digital processing in the horizontal direction, one complete 16-element picture line at a time, by the fast Fourier transform invoked by subroutine FFTH. The attempt fails catastrophically; extensive false detail is generated; for example, the gun in rows 3, 4, and 5 of the images is completely destroyed. One can speculate that the incapacity of the fast Fourier transform to handle this two-tone image is due to the limitations of the transform bandwidth. Where N = 16, the transform can properly sample only 8 spatial frequency components, a dc term, and one through 7 cycles of sinusoidal oscillation for each data field length of 16 points. Therefore, seven terms are not enough in the Fourier series expansions of the image signal to accurately represent the bandwidth needed to characterize the sharp transitions at the edges of the various tank target elements.

Figure 3 elicits better results and shows digital processing in the horizontal direction by the Haar transform invoked by subroutine HAR4H, i.e., four elements for each transform and 4 transforms for each horizontal picture line. The m-values, 0.5, 10 and 15, are the enhancement parameters and weight the strengths of the dc term and terms of one and two cycles of square-wave oscillation,

FFT 8 HOR ENH
M1 = 5
M2 = 1 0
M3 = 2 0
M4 = 3 0
M5 = 4 0
M6 = 5 0
M7 = 6 0
M8 = 7 0

BACKGR= 19 0
SIGNAL= 20 0
RMS= 0 0
CONTRAST IS 5 0%
S/N RATIO IS ****
APSIG IS = 10 63
PRMS IS= 9 84
GUNH= 38 46%
GUNV= 34 62%
COUH= 27 27%
COUV= 12 12%
TURH= 27 27%
TURV= 27 27%
HULH= 33 33%
HULV= 22 62%
WHLH= 33 33%
WHLV= 4 17%
TNKH= 31 93%
TNKV= 20 16%

19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19
19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19
19	19	19	19	19	19	19	19	19	19	19	19	20	19	19	19
19	19	19	19	19	19	19	19	19	19	19	20	19	19	19	19
19	19	19	19	19	19	20	20	20	19	20	19	19	19	19	19
19	19	19	19	19	20	20	20	20	20	19	19	19	19	19	19
19	19	19	19	19	20	20	20	20	20	19	19	19	19	19	19
19	19	19	20	20	20	20	20	20	20	20	20	20	19	19	19
19	19	19	20	20	20	20	20	20	20	20	20	20	19	19	19
19	19	19	20	20	20	20	20	20	20	20	20	20	19	19	19
19	19	19	20	20	20	20	20	20	20	20	20	20	19	19	19
19	19	19	19	20	20	20	20	20	20	20	20	19	19	19	19
19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19
19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19
19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19
19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19

10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10
9	10	9	10	9	10	9	10	9	10	9	8	13	8	9	10
10	9	10	9	10	9	10	9	10	9	8	13	8	9	10	9
9	10	9	10	9	9	11	11	11	7	13	9	9	10	9	10
10	9	10	9	8	11	11	10	11	11	8	9	10	9	10	9
10	9	10	9	8	11	11	10	11	11	8	9	10	9	10	9
9	9	8	12	10	10	10	10	10	10	10	10	12	8	9	9
9	9	8	12	10	10	10	10	10	10	10	10	12	8	9	9
9	9	8	12	10	10	10	10	10	10	10	10	12	8	9	9
9	9	8	12	10	10	10	10	10	10	10	10	12	8	9	9
9	9	9	8	12	10	10	10	10	10	10	12	8	9	9	9
10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10

Figure 2. Fast Fourier transform, horizontal scan.

```
HAR4H SHOR   ENH
M1  =     5
M2  =  10 0
M3  =  15 0

BACKGR= 19 0
SIGNAL= 20 0
RMS=  0 0
CONTRAST IS   5 0%
S/N RATIO IS ****
APSIG IS =   12 53
PRMS IS=   10 37
GUNH=   71 02%
GUNV=   63 53%
COUH=   77 45%
COUV=   41 83%
TURH=   83 33%
TURV=   76 67%
HULH=   75 00%
HULV=   57 14%
WHLH=    0 00%
WHLV=    0 00%
TNKH=   61 37%
TNKV=   47 83%
```

19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19
19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19
19	19	19	19	19	19	19	19	19	19	19	19	20	19	19	19
19	19	19	19	19	19	19	19	19	19	19	20	19	19	19	19
19	19	19	19	19	19	20	20	20	19	20	19	19	19	19	19
19	19	19	19	19	20	20	20	20	20	19	19	19	19	19	19
19	19	19	19	19	20	20	20	20	20	19	19	19	19	19	19
19	19	19	20	20	20	20	20	20	20	20	20	20	19	19	19
19	19	19	20	20	20	20	20	20	20	20	20	20	19	19	19
19	19	19	20	20	20	20	20	20	20	20	20	20	19	19	19
19	19	19	20	20	20	20	20	20	20	20	20	20	19	19	19
19	19	19	19	20	20	20	20	20	20	20	20	19	19	19	19
19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19
19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19
19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19
19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19

10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10	10	10	10	10	20	5	7	7
10	10	10	10	10	10	10	10	7	7	5	20	10	10	10	10
10	10	10	10	5	5	15	15	17	2	17	2	10	10	10	10
10	10	10	10	0	15	12	12	15	15	5	5	10	10	10	10
10	10	10	10	0	15	12	12	15	15	5	5	10	10	10	10
7	7	5	20	10	10	10	10	10	10	10	10	20	5	7	7
7	7	5	20	10	10	10	10	10	10	10	10	20	5	7	7
7	7	5	20	10	10	10	10	10	10	10	10	20	5	7	7
7	7	5	20	10	10	10	10	10	10	10	10	20	5	7	7
10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10

Figure 3. Haar transform, 4-element horizontal scan.

HAD4H SHOR. ENH
M1 = 5
M2 = 15 0
M3 = 10 0
M4 = 10 0

BACKGR= 19 0
SIGNAL= 20 0
RMS= 0.0
CONTRAST IS 5 0%
S/N RATIO IS ****
APSIG IS = 12 22
PPMS IS= 10 25
GUNH= 66 45%
GUNV= 59 59%
COUH= 77 45%
COUV= 43 79%
TUPH= 79 76%
TUPV= 75 48%
HULH= 66 67%
HULV= 52 54%
WHLH= 0 00%
WHLV= 0 00%
TNKH= 58 07%
TNKV= 46 28%

19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19
19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19
19	19	19	19	19	19	19	19	19	19	19	19	20	19	19	19
19	19	19	19	19	19	19	19	19	19	19	20	19	19	19	19
19	19	19	19	19	19	20	20	20	19	20	19	19	19	19	19
19	19	19	19	19	20	20	20	20	20	19	19	19	19	19	19
19	19	19	19	19	20	20	20	20	20	19	19	19	19	19	19
19	19	19	20	20	20	20	20	20	20	20	20	20	19	19	19
19	19	19	20	20	20	20	20	20	20	20	20	20	19	19	19
19	19	19	20	20	20	20	20	20	20	20	20	20	19	19	19
19	19	19	20	20	20	20	20	20	20	20	20	20	19	19	19
19	19	19	19	20	20	20	20	20	20	20	20	19	19	19	19
19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19
19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19
19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19
19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19

10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10	10	10	10	10	18	6	8	6
10	10	10	10	10	10	10	10	6	8	6	18	10	10	10	10
10	10	10	10	5	5	15	15	17	2	17	2	10	10	10	10
10	10	10	10	1	14	11	14	15	15	5	5	10	10	10	10
10	10	10	10	1	14	11	14	15	15	5	5	10	10	10	10
6	8	6	18	10	10	10	10	10	10	10	10	18	6	8	6
6	8	6	18	10	10	10	10	10	10	10	10	18	6	8	6
6	8	6	18	10	10	10	10	10	10	10	10	18	6	8	6
6	8	6	18	10	10	10	10	10	10	10	10	18	6	8	6
10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10

Figure 4. Hadamard transform, horizontal scan.

HAR16H 5HOR. ENH
M1 = 5
M2 = 1 0
M3 = 5 0
M4 = 10 0
M5 = 15 0

BACKGR= 19 0
SIGNAL= 20 0
RMS= 0 0
CONTRAST IS 5 0%
S/N RATIO IS ****
APSIG IS = 13.84
PRMS IS= 10 31
GUNH= 71 11%
GUNV= 63 98%
COUH= 72 92%
COUV= 47 69%
TURH= 75 37%
TURV= 72 98%
HULH= 83 33%
HULV= 59 52%
WHLH= 41 67%
WHLV= 16 67%
TNKH= 68 88%
TNKV= 52 17%

19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19
19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19
19	19	19	19	19	19	19	19	19	19	19	19	20	19	19	19
19	19	19	19	19	19	19	19	19	19	19	20	19	19	19	19
19	19	19	19	19	19	20	20	20	19	20	19	19	19	19	19
19	19	19	19	19	20	20	20	20	20	19	19	19	19	19	19
19	19	19	19	19	20	20	20	20	20	19	19	19	19	19	19
19	19	19	20	20	20	20	20	20	20	20	20	20	19	19	19
19	19	19	20	20	20	20	20	20	20	20	20	20	19	19	19
19	19	19	20	20	20	20	20	20	20	20	20	20	19	19	19
19	19	19	20	20	20	20	20	20	20	20	20	20	19	19	19
19	19	19	19	20	20	20	20	20	20	20	20	19	19	19	19
19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19
19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19
19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19
19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19

10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10
9	9	9	9	9	9	9	9	9	9	9	9	20	5	8	8
9	9	9	9	9	9	9	9	8	8	5	20	9	9	9	9
8	8	8	8	6	6	16	16	18	3	18	3	8	8	8	8
8	8	8	8	2	17	14	14	16	16	6	6	8	8	8	8
8	8	8	8	2	17	14	14	16	16	6	6	8	8	8	8
5	5	3	18	12	12	12	12	12	12	12	12	18	3	5	5
5	5	3	18	12	12	12	12	12	12	12	12	18	3	5	5
5	5	3	18	12	12	12	12	12	12	12	12	18	3	5	5
5	5	3	18	12	12	12	12	12	12	12	12	18	3	5	5
7	7	7	7	12	12	12	12	12	12	12	12	7	7	7	7
10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10

Figure 5. Haar transform, 16-element horizontal scan.

respectively, for each data field length of four pixels. Notice that contrast is improved both by boosting the signal level for the gun and the other tank components through the second two of the three m-values, and by reducing the background almost 50 percent through the first m-value which takes out some of the dc term.

Figure 4 shows horizontal processing by the Hadamard 4-by-4 matrix, through subroutine HAD4H, with m-values 0.5, 15, 10, and 10. The first governs the background; the second, the highest spatial frequency; the third, the next highest, and so on. The four spatial frequency components are not as systematically arranged as in the Haar transform because the sequencies or frequencies of value change are out of order for this particular algorithm for the Hadamard transform.

Finally, figure 5 shows processing in the horizontal direction by subroutine HAR16H, i.e., via the HAAR transform, taking 16 elements or each line of pixels at a time. The M-values are 0.5, 1, 5, 10, and 15, and govern the d.c. term and terms of 1, 2, 4, and 8 cycles of spatial frequency oscillation for each data field length of 16 pixel points, respectively. This example produces the best contrast improvement results because of the greater flexibility afforded by the larger number of available spatial frequency enhancement parameters that can be varied. For example, because, in figure 3, the Haar 4-element scan "sees" the data in blocks of four, it never sees the transition between the background and the tread of the tank. Round off in the integer expression of the output obscures the tread by blending it into the background, giving elements the same pixel value of ten. The 16-element scan of subroutine HAR16H does see the tread and displays it with an intensity about 30 percent higher than the background.

Note again that in these scans, where contrast differences are pulled out or enhanced horizontally, the results over the scan of several horizontal lines is the enhancement of vertical edges or lines oriented preferentially in the vertical direction.

CONCLUSION

An interactive program with the capability of test-demonstrating the effects of image enhancement and processing algorithms currently exists on the CDC 6500 computer system at the ARRADCOM Dover site. The current status of the program exhibits nine horizontal, vertical, and two-dimensional scanning algorithms employing three transform processing techniques: specifically the Haar, Fourier, and Hadamard types in modular form. Other algorithms of interest to the user can be simply tacked on as Fortran callable subroutines. Then algorithms can employ the package's capability of contrast, and signal-to-noise measurement of input imagery data.

REFERENCES

1. Thomas S. Huang, William H. Schreiber, Oleh J. Tretiak, "Image Processing," Proceedings of the I.E.E.E., November, 1971, pp 1586-1609.

2. Gary Sivak, "The Haar Transform Its Theory and Computer Implementation," ARRADCOM Technical Report No. ARSCD-TR-78005, Sept. 1978.

3. Gary Sivak, "The Application of the Haar Transform to IR Imagery."

4. James E. Rothman, "The Fast Fourier Transform and Its Implementation," Digital Equipment Corp., Maynard, Mass.

5. Gary Sivak, "Advanced Video Signal Processing Techniques for Image Enhancement," Frankford Arsenal Technical Symposium, 1976, pp 95-114.

APPENDIX A. PROGRAM IMAGE

```
      PROGRAM IMAGE(INPUT,OUTPUT,TAPE4=65,TAPE61=100,TAPE62=100)
      DIMENSION IN(16,16),IOUT(16,16),IX(56),IY(56)
      DATA IX/1641,2986,2986,3154,3154,2658,2662,3094,
     &3138,3146,3162,3114,3082,3038,3018,3018,2490,2490,1981,1981,
     &1817,1817,1469,1469,1641,1641,1753,1753,1729,2994,2994,3162,
     &3162,2662,2662,3134,3146,3158,3130,3082,3050,3014,2998,2518,
     &2494,2494,1977,1977,1813,1813,1477,1477,1641,1641,1753,1753/
      DATA IY/1991,1991,2071,2071,2427,2427,2567,2811,
     &2815,2831,2875,2883,2879,2875,2855,2827,2599,2703,2703,2611,
     &2611,2435,2435,2079,2079,1991,1991,1991,408,404,492,492,
     &844,848,996,1204,1232,1268,1284,1288,1284,1272,1252,1020,
     &1020,1116,1116,1028,1028,848,848,488,488,408,408,408/
      CALL INITT(30)
    1 PRINT *,"DO YOU WANT TO TEST NUMBER SEQUENSES ON YOUR ALGORITHM?"
      CALL BEEP(1,.1,0.0)
      READ 50,NS
   50 FORMAT (A3)
      IF(NS .EQ. "YES") GO TO 2
      PRINT *,"ENTER BACKGROUND, SIGNAL AND NOISE LEVELS. "
      CALL BEEP(3,.1,.2)
      READ *,B,S,RMS $CONTRA = 100.*(S-B)/S
      IF (RMS.EQ.0.) STN=100.  $IF (RMS.NE.0.0) STN=S/RMS
    2 PRINT *,"SPECIFY WHICH OF THE FOLLOWING TRANSFORMS BY NUMBER:"
      PRINT *,"1-HAR(16,16) MATRIX, HORIZONTAL SCAN."
      PRINT *,"2-HAR(16,16) MATRIX, VERTICAL SCAN."
      PRINT *,"3-HAR(16,16) MATRIX, HORIZONTAL THEN VERTICAL SCAN."
      PRINT *,"4-HAR(4,4)   MATRIX, HORIZONTAL SCAN."
      PRINT *,"5-HADAMARD(4,4) MATRIX, HORIZONTAL SCAN."
      PRINT *,"6-FOURIER TRANSFORM, HORIZONTAL SCAN."
      PRINT *,"7-FOURIER TRANSFORM, 2-DIMENSIONAL SCAN."
      PRINT *,"8 FFT HORIZ. SCAN"
      PRINT *,"9-FFT HORIZ. THEN VERT. SCAN"
      CALL BEEP(1,.1,.1)
   13 READ *,INDEX
      IF(INDEX .LE. 9) GO TO 14
      PRINT *,"INCORRECT INCEX, TRY AGAIN. "
      GO TO 13
   14 IF(NS .EQ. "YES") GO TO 5
      DO 6 I=1,16
      DO 6J=1,16
    6 IN(I,J)=B
      DO 7 I=1,3
      IN(2+I,14-I)=S
    7 IN(5,6+I)=S
      DO 8 I=6,7
      DO 8 J=6,10
    8 IN(I,J)=S
      DO 9 I=8,11
```

```
      9 J=4,13
  9 IN(I,J)=S
    DO 10 J=5,12
 10 IN(12,J)=S
    SQ = RMS*SQRT((2.*16.+1.)/(6.*16.))
    IF(RMS .EQ. 0.0) C = 0.0
    IF(RMS .NE. 0.0) C = RMS*RMS/SQ*2.0
    DO 11 I=1,16
    DO 11 J=1,16
 11 IN(I,J)=IN(I,J)+(RANF(9)-0.5)*C+0.5
  5 IF (INDEX.EQ.1) CALL HAR16H(IN,IOUT,NS)
    IF (INDEX.EQ.2) CALL HAR16V(IN,IOUT,NS)
    IF (INDEX.EQ.3) CALL HAR16HV(IN,IOUT,NS)
    IF (INDEX.EQ.4) CALL HAR4H(IN,IOUT,NS)
    IF (INDEX.EQ.5) CALL HAD4H(IN,IOUT,NS)
    IF (INDEX.EQ.6) CALL FT16H(IN,IOUT,NS)
    IF (INDEX.EQ.7) CALL FT16D2(IN,IOUT,NS)
    IF (INDEX.EQ.8) CALL FFTH(IN,IOUT,NS)
    IF (INDEX.EQ.9) CALL FFTHV(IN,IOUT,NS)
 41 IF(NS .EQ. "YES") GO TO 42
    PRINT 43,B,S,RMS
 43 FORMAT (1H "BACKGR= "F4.1/1H "SIGNAL= "F4.1/1H "RMS= "F4.1)
    PRINT 44,CONTRA,STN
 44 FORMAT(1H ,"CONTRAST IS:",F4.1"%"/1H ,"S/N RATIO IS:",F4.1)
    CALL APSIG(IOUT)
    CALL PRMS(IOUT)
    CALL CTRAST(IOUT)
 51 CALL HOME
    CALL ANMODE
    DO 52 I=1,16
 52 PRINT53,(IN(I,J),J=1,16)
 53 FORMAT(1H ,17X,16I3)
    PRINT *," "
    DO 54 I=1,16
 54 PRINT53,(IOUT(I,J),J=1,16)
    CALL MOVABS(IX(1),IY(1))
    DO 55 I = 2,28
 55 CALL DRWABS(IX(I),IY(I))
    CALL MOVABS(IX(29),IY(29))
    DO 56 I = 30,56
 56 CALL DRWABS(IX(I),IY(I))
    CALL ANMODE
    CALL BEEP(2,.1,.2)
    READ *,DUMMY
 42 GO TO 1
    END
    SUBROUTINE BEEP(N, TB, TD)
    BELL = 02070215555555555555B
    BLANK = 55555555555555555555B
    I = IFIX(10.1*TB)
    J = IFIX(5.1*TD)
```

```
      CALL CONNEC(4,2)
      PRINT *," "
      DO 1 K = 1,N
      DO 2 L = 1,I
    2 WRITE (4) BELL
      IF (J.EQ.0) GO TO 1
      DO 3 M = 1,J
    3 PRINT 4,BLANK
    1 CONTINUE
      CALL DISCON(4,2)
    4 FORMAT (1H+,A1)
      RETURN
      END
      SUBROUTINE APSIG(IOUT)
      DIMENSION IOUT(16,16)
      PSIG=0.0
      DO 1 I=1,3
      PSIG=PSIG+IOUT(2+I,14-I)
    1 PSIG=PSIG+IOUT(5,6+I)
      DO 2 I=6,7
      DO 2 J=6,10
    2 PSIG=PSIG+IOUT(I,J)
      DO 3 I=8,11
      DO 3 J=4,13
    3 PSIG=PSIG+IOUT(I,J)
      DO 4 J=5,12
    4 PSIG=PSIG+IOUT(12,J)
      PSIG=PSIG/64.0
      PRINT5,PSIG
    5 FORMAT(1H ,"APSIG IS = ",F6.2)
      RETURN
      END
      SUBROUTINE PRMS(IOUT)
      DIMENSION IOUT(16,16)
      CRMS=0.0
      DO 1 I=1,16
      DO 1 J=1,16
    1 CRMS=CRMS+IOUT(I,J)*IOUT(I,J)
      CRMS=SQRT(CRMS/256.0)
      PRINT2,CRMS
    2 FORMAT(1H ,"PRMS IS= ",F6.2)
      RETURN
      END
      SUBROUTINE CTRAST(IOUT)
      DIMENSION IOUT(16,16),R(16,16),A(12),G(12)
      DATA A/"GUNH=","GUNV=","COUH=","COUV=","TURH=","TURV=",
     &"HULH=","HULV=","WHLH=","WHLV=","TNKH=","TNKV="/
      DO 2 I = 1,16
      DO 2 J=1,16
      R(I,J)=IOUT(I,J)
    2 IF(R(I,J).EQ.0.) R(I,J)=1.E-14
```

```
      G(1)=((2.*R(3,13)-R(3,12)-R(3,14))/R(3,13)+(2*R(4,12)-R(4,11)-R(4,
     113))/R(4,12)+(2*R(5,11)-R(5,10)-R(5,12))/R(5,11))/6.0*100.0
      G(2)=((2*R(3,13)-R(2,13)-R(4,13))/R(3,13)+(2*R(4,12)-R(3,12)-R(5,1
     12))/R(4,12)+(2*R(5,11)-R(4,11)-R(6,11))/R(5,11))/6.0*100.0
      G(3)=((R(5,7)-R(5,6))/R(5,7)+(R(5,9)-R(5,10))/R(5,9))/2.0*100.0
      G(4)=((R(5,7)-R(4,7))/R(5,7)+(R(5,8)-R(4,8))/R(5,8)+(R(5,9)-R(4,9)
     1)/R(5,9))/3.0*100.0
      G(5)=((R(6,6)-R(6,5))/R(6,6)+(R(6,10)-R(6,11))/R(6,10)+(R(7,6)-R(7
     1,5))/R(7,6)+(R(7,10)-R(7,11))/R(7,10))/4.0*100.0
      G(6)=((R(6,6)-R(5,6))/R(6,6)+(R(6,10)-R(5,10))/R(6,10))/2.0*100.0
      G(7)=((R(8,4)-R(8,3))/R(8,4)+(R(8,13)-R(8,14))/R(8,13)+(R(9,4)-R(9
     1,3))/R(9,4)+(R(9,13)-R(9,14))/R(9,13)+(R(10,4)-R(10,3))/R(10,4)+(R
     2(10,13)-R(10,14))/R(10,13)+(R(11,4)-R(11,3))/R(11,4)+(R(11,13)-R(1
     31,14))/R(11,13))/8.0*100.0
      G(8)=((R(8,4)-R(7,4))/R(8,4)+(R(8,5)-R(7,5))/R(8,5)+(R(8,11)-R(7,1
     11))/R(8,11)+(R(8,12)-R(7,12))/R(8,12)+(R(8,13)-R(7,13))/R(8,13)+(R
     2(11,4)-R(12,4))/R(11,4)+(R(11,13)-R(12,13))/R(11,13))/7.0*100.0
      G(9)=((R(12,5)-R(12,4))/R(12,5)+(R(12,12)-R(12,13))/R(12,12))*50.0
      G(10)=((R(12,5)-R(13,5))/R(12,5)+(R(12,6)-R(13,6))/R(12,6)+(R(12,7
     1)-R(13,7))/R(12,7)+(R(12,8)-R(13,8))/R(12,8)+(R(12,9)-R(13,9))/R(1
     22,9)+(R(12,10)-R(13,10))/R(12,10)+(R(12,11)-R(13,11))/R(12,11)+(R(
     312,12)-R(13,12))/R(12,12))/8.0*100.0
      G(11)=(G(1)+G(3)+G(5)+G(7)+G(9))/5.
      G(12)=(G(2)+G(4)+G(6)+G(8)+G(10))/5.
      DO   3 I=1,12
    3 PRINT  5,A(I),G(I)
    5 FORMAT(1H ,A5,F7.2,"%")
      RETURN
      END
      SUBROUTINE HAAR(Q,E,NU)
      DIMENSION T(16),Q(30),E(5)
      N =2**NU
      NO =N
      ME = NU+2
      IP = (-2)*N
      IR = 0
    5 NO = NO/2
      IP = IP + 4*NO
      IR = IR + 2*NO
      ME = ME-1
      DO 6 J =1,NO
      L = 2*J
      K = L-1
      NOJ = NO+J
      T(NOJ) = E(ME)*(Q(IP+K)-Q(IP+L))
    6 Q(IR+J) =Q(IP+K)+Q(IP+L)
      IF (NO.GT.2) GO TO 5
      T(2) =E(2)*(Q(IR+1)-Q(IR+2))
      T(1) =E(1)*(Q(IR+1)+Q(IR+2))
      Q(1) =(T(1)+T(2))/FLOAT(N)
      Q(2) =(T(1)-T(2))/FLOAT(N)
```

```
      IP= -1
      IR = 0
    7 N2 = NO
      X = FLOAT(N2)/FLOAT(N)
      NO = NO*2
      IP = IP + NO/4
      IR = IR + NO/2
      DO 8 J =1,NO
      ND = (J+1)/2
      ID = N2 + ND
      Y = (-1.0)**(J+1)*X
    8 Q(IR+J) = Q(IP+ND)+Y*T(ID)
      IF (NO.LT.N) GO TO 7
      RETURN
      END
      SUBROUTINE HAR16H(IN, IOUT, NS)
      DIMENSION IN(16,16),IOUT(16,16),Q(30),E(5)
      PRINT *,"ENTER 5 M-VALUES."
      CALL BEEP(5,.1,.1)
      READ *,(E(J),J=1,5)
      CALL ERASE
      PRINT *,"HAR16H 5HOR. ENH"
      PRINT 1, ((J,E(J)),J=1,5)
    1 FORMAT (5(1H ,"M",I1," = ",F4.1,/))
      NU = 4
      M = 14
      IF(NS .EQ. "YES") GO TO 2
      DO 3 I = 1,16
      DO 4 J = 1,16
    4 Q(J) = IN(I,J)
      CALL HAAR(Q,E,NU)
      DO 5 K = 1,16
    5 IOUT(I,K)=Q(M+K)+0.5
    3 CONTINUE
      GO TO 6
    2 PRINT *,"ENTER 16 INPUTS."
      CALL BEEP(16,.1,.2)
      READ *,(Q(J),J=1,16)
      CALL HAAR(Q,E,NU)
      PRINT *,(Q(M+J),J=1,16)
    6 RETURN
      END
      SUBROUTINE HAR16V(IN, IOUT, NS)
      DIMENSION IN(16,16),IOUT(16,16),Q(30),E(5)
      IF(NS .EQ. "YES") GO TO 2
      PRINT *,"ENTER 5 M-VALUES."
      CALL BEEP(5,.1,.1)
      READ *,(E(J),J = 1,5)
      CALL ERASE
      PRINT *,"HAR16V 5VER. ENH"
      PRINT 1,((J,E(J)),J=1,5)
```

```
1 FORMAT (5(1H ,"M",I1," = ",F4.1/))
  NU = 4
  M = 14
  DO 3 I = 1,16
  DO 4 J = 1,16
4 Q(J) = IN(J,I)
  CALL HAAR(Q,E,NU)
  DO 5 K = 1,16
5 IOUT(K,I)=Q(M+K)+0.5
3 CONTINUE
2 RETURN
  END
  SUBROUTINE HAR16HV(IN,IOUT,NS)
  DIMENSION IN(16,16),IOUT(16,16),Q(30),H(5),V(5)
  IF(NS .EQ. "YES") GO TO 1
  PRINT *,"ENTER 5 HORIZONTAL M-VALUES."
  CALL BEEP(5,.1,.1)
  READ *,(H(J),J=1,5)
  PRINT *,"ENTER 5 VERTICAL M-VALUES"
  CALL BEEP(5,.1,.1)
  READ *,(V(J),J=1,5)
  CALL ERASE
  PRINT *,"HAR16HV 5HOR. ENH"
  PRINT 2,((J,H(J)),J=1,5)
  PRINT *,"5 VERT ENHANCERS"
  PRINT 2,((J,V(J)),J=1,5)
2 FORMAT (5(1H ,"M",I1," = ",F4.1,/))
  M=14
  NU = 4
  DO 3 I = 1,16
  DO 4 J = 1,16
4 Q(J) = IN(I,J)
  CALL HAAR(Q,H,NU)
  DO 5 K = 1,16
5 IOUT(I,K) = Q(M+K)+.5
3 CONTINUE
  DO 6 I = 1,16
  DO 7 J = 1,16
7 Q(J) = IOUT(J,I)
  CALL HAAR(Q,V,NU)
  DO 8 K = 1,16
8 IOUT(K,I) = Q(M+K)+.5
6 CONTINUE
1 RETURN
  END
  SUBROUTINE HAR4H(IN, IOUT, NS)
  DIMENSION IN(16,16),IOUT(16,16),Q(6),E(3)
  IF(NS.EQ."YES") GO TO 6
  PRINT *,"ENTER 3 M-VALUES."
  CALL BEEP(3,.1,.1)
  READ *,(E(J),J=1,3)
```

```
      CALL ERASE
      PRINT *,"HAR4H 5HOR. ENH"
      PRINT 1,((J,E(J)),J=1,3)
    1 FORMAT (3(1H ,"M",I1," = ",F4.1,/))
      NU = 2
      M = 2
      DO 3 I = 1,16
      DO 3 J = 1,4
      DO 4 K = 1,4
    4 Q(K) = IN(I,4*(J-1)+K)
      CALL HAAR(Q,E,NU)
      DO 5 L = 1,4
      K = 4*(J-1)+L
    5 IOUT(I,K) = Q(M+L)+.5
    3 CONTINUE
    6 RETURN
      END
      SUBROUTINE HAD(Q,E,NU)
      DIMENSION T(4),Q(4),E(4)
      N = 2**NU
      IT = 0
    1 IT = IT +1
      T(1) = Q(1) + Q(2) + Q(3) + Q(4)
      T(2) = Q(1) - Q(2) + Q(3) - Q(4)
      T(3) = Q(1) + Q(2) - Q(3) - Q(4)
      T(4) = Q(1) - Q(2) - Q(3) + Q(4)
      DO 2 I = 1,N
      IF(IT .EQ. 1) T(I) = T(I) * E(I)
      IF(IT .EQ. 2) T(I) = T(I) / N
    2 Q(I) = T(I)
      IF(IT .LT. 2) GO TO 1
      RETURN
      END
      SUBROUTINE HAD4H(IN, IOUT, NS)
      DIMENSION IN(16,16),IOUT(16,16),Q(4),E(4)
      NU = 2
      N = 2**NU
      M = 0
      PRINT *,"ENTER ",N," HORIZONTAL ENHANCERS. "
      CALL BEEP(N,.1,.1)
      READ *,(E(J),J=1,N)
      CALL ERASE
      PRINT *,"HAD46 EHOR. ENH"
      PRINT 1,((J,E(J)),J=1,N)
    1 FORMAT (4(1H ,"M",I1," = ",F4.1,/))
      IF(NS.EQ."NO") GO TO 2
      DO 3 I = 1,16
      DO 3 J = 1,N
      DO 4 K = 1,N
    4 Q(K) = IN(I, N*(J-1) + K)
      CALL HAD(Q,E,NU)
```

```
      DO 5 L = 1,N
      K = 4*(J-1) + L
    5 IOUT(I,K) = Q(M+L)+.5
    3 CONTINUE
      GO TO 6
    2 PRINT *,"ENTER ",N," INPUTS. "
      CALL BEEP(N,.1,.1)
      READ *,(Q(J),J=1,N)
      CALL HAD(Q,E,NU)
      PRINT *,(Q(M+J),J=1,N)
    6 RETURN
      END
      SUBROUTINE FT(XR,NU,E)
      DIMENSION CO(16),SI(16),XR(16),XI(16),E(8),FR(16),FI(16)
      N = 2**NU
      NP = N/2
      TPN = 8. * ATAN(1.) / N
      DO 1 I = 1,N
      X = TPN * (I-1)
      CO(I) = COS(X)
    1 SI(I) = SIN(X)
      IC = 0
    6 IC = IC + 1
      SIGNR = FLOAT((-1)**IC)
      SIGNI = -1. * SIGNR
      IF(IC .EQ. 1) SCALE = NP
      IF(IC .EQ. 2) SCALE = 2.0
      DO 2 I = 1,N
      FR(I) = FI(I) = 0.
      DO 2 J = 1,N
      M1 = (I-1) * (J-1)
      M2 = MOD(M1,N) + 1
      FR(I) = FR(I) + CO(M2)*XR(J) + SIGNR*SI(M2)*XI(J)
    2 FI(I) = FI(I) + CO(M2)*XI(J) + SIGNI*SI(M2)*XR(J)
      DO 3 I = 1,N
      XR(I) = FR(I) / SCALE
    3 XI(I) = FI(I) / SCALE
      IF(IC .EQ. 2) GO TO 4
      DO 5 I = 1,NP
      J = N+2 -I
      XR(I) = XR(I) * E(I)
      XI(I) = XI(I) * E(I)
      IF(I .GT. 1) XR(J) = XR(J) * E(I)
    5 IF(I .GT. 1) XI(J) = XI(J) * E(I)
      GO TO 6
    4 RETURN
      END
      SUBROUTINE FT16H(IN,IOUT,NS)
      DIMENSION IN(16,16),IOUT(16,16),XR(16),E(8)
      NU = 4
      N = 2**NU
```

```
      NP = N/2
      IF(NS .EQ. "YES") GO TO 1
      PRINT *,"ENTER ",NP," HORIZONTAL ENHANCERS. "
      CALL BEEP(NP,.1,.1)
      READ *,(E(J),J=1,NP)
      CALL ERASE
      DO 2 I=1,N
      DO 3 J=1,N
    3 XR(J)=IN(I,J)
      CALL FT(XR,NU,E)
      DO 4 K=1,N
    4 IOUT(I,K)=XR(K)+.5
    2 CONTINUE
      PRINT *,"FT. ",NP," HOR. ENH"
      PRINT 5,((J,E(J)),J=1,NP)
    5 FORMAT (8(1H ,"M",I1," = ",F4.1,/))
    1 RETURN
      END
      SUBROUTINE FT2D(XR,NU,H,V)
      DIMENSION CO(16),SI(16),XR(16,16),XI(16,16),H(8),V(8),
     &FR(16,16), FI(16,16)
      N = 2**NU
      NP = N/2
      TPN = 8. * ATAN(1.) / N
      DO 1 I = 1,N
      X = TPN*(I-1)
      CO(I) = COS(X)
    1 SI(I) = SIN(X)
      IC = 0
    3 IC = IC + 1
      SIGNR = FLOAT((-1)**IC)
      SIGNI = -1.0 * SIGNR
      IF(IC.EQ. 1) SCALE = N*N/4
      IF(IC .EQ. 2) SCALE = 4.0
      DO 4 L = 1,N
      DO 4 K = 1,N
      FR(L,K) = FI(L,K) = 0.
      DO 4 J = 1,N
      DO 4 I = 1,N
      M1 = (L-1) * (J-1) +  (K-1) * (I-1)
      M2 = MOD(M1,N) +1
      FR(L,K) = FR(L,K)+CO(M2)*XR(J,I)+SIGNR*SI(M2)*XI(J,I)
    4 FI(L,K)=FI(L,K)+CO(M2)*XI(J,I)+SIGNI*SI(M2)*XR(J,I)
      DO 5 I = 1,N
      DO 5 J = 1,N
      XR(I,J) = FR(I,J) / SCALE
    5 XI(I,J) = FI(I,J) / SCALE
      IF(IC .EQ. 2) GO TO 6
      DO 7 I = 1,NP
      K = N+2 -I
      DO 7 J = 1,N
```

```
      XR(J,I) = XR(J,I) * H(I)
      XI(J,I) = XI(J,I) * H(I)
      XR(I,J) = XR(I,J) * V(I)
      XI(I,J) = XI(I,J) * V(I)
      IF(I.GT.1) XR(J,K) = XR(J,K) * H(I)
      IF(I.GT.1) XI(J,K) = XI(J,K) * H(I)
      IF(I.GT. 1) XR(K,J) = XR(K,J) * V(I)
    7 IF(I.GT.1) XI(K,J) = XI(K,J) * V(I)
      GO TO 3
    6 RETURN
      END
      SUBROUTINE FT16D2(IN, IOUT, NS)
      DIMENSION IN(16,16),IOUT(16,16),XR(16,16),H(8),V(8)
      NU = 4
      N = 2**NU
      NP = N/2
      IF(NS.EQ."YES") GO TO 1
      PRINT *,"ENTER ",NP," HORIZONTAL SPATIAL FREQUENCY ENHANCERS. "
      CALL BEEP(N, .1, .1)
      READ *,(H(J), J=1,NP)
      PRINT *,"ENTER ",NP," VERTICAL SPATIAL FREQUENCY ENHANCERS. "
      CALL BEEP(NP, .1, .1)
      READ *,(V(J),J=1,NP)
      DO 2 I = 1,N
      DO 2 J = 1,N
    2 XR(I,J) = IN(I,J)
      CALL FT2D(XR,NU,H,V)
      DO 3 I = 1,N
      DO 3 J = 1,N
    3 IOUT(I,J) = XR(I,J)+.5
      CALL ERASE
      PRINT *,"FT2D ",NP," HOR. ENH"
      PRINT 10,((J,H(J)),J=1,NP)
   10 FORMAT(8(1H ,"M",I1," = ",F4.1,/))
      PRINT *,"FT2D ",NP," VERT. ENH"
      PRINT 10,((J,V(J)),J=1,NP)
    1 RETURN
      END
      SUBROUTINE FFT(XR,NU,E)
      DIMENSION XR(16),XI(16),E(8)
      INTEGER S,Q,H,P,C,U
      IC= 0
      N = 2**NU
      TPN = 8. * ATAN(1.) / FLOAT(N)
      X=3.0/FLOAT(N)
      DO 23 I=1,N
   23 XI(I)=0.0
   24 IC=IC+1
      SIGNR = FLOAT((-1)**IC)
      SIGNI = -1. * SIGNR
      H=1-NU
```

```
      L=1
      NP=N/2
      Q=NP
      S=NP
    2 XSTOR=XR(Q)+XR(Q+S)
      XR(Q+S)=XR(Q)-XR(Q+S)
      XR(Q)=XSTOR
      IF (IC.EQ.1) GO TO 3
      XSTOR=XI(Q)+XI(Q+S)
      XI(Q+S)=XI(Q)-XI(Q+S)
      XI(Q)=XSTOR
    3 IF (Q.LE.1) GO TO 5
    4 Q=Q-1
      GO TO 2
    5 IF (L.EQ.NU) GO TO 14
    6 L=L+1
      S=S/2
      H=H+1
      P=N
    7 C=1
    8 U=FLOAT(P-1)*2.0**FLOAT(H)+X
      CALL REVERS(U,K,IR,NU)
      CO=COS(TPN*K)
      SI = SIN(TPN*K)
      GR = CO * XR(P) +SIGNR * SI * XI(P)
      GI = CO * XI(P) + SIGNI * SI * XR(P)
      Q=P-S
      XR(P)=XR(Q)+GR
      XR(Q)=XR(Q)-GR
      XI(P)=XI(Q)+GI
      XI(Q)=XI(Q)-GI
    9 P=P-1
   10 IF(C.EQ.S)GO TO 12
   11 C=C+1
      GO TO  8
   12 IF(P.EQ.S)GO TO 5
   13 P=P-S
      GO TO 7
   14 Q=N
   15 P=P-1
      Q=Q-1
      CALL REVERS(Q,P,IR,NU)
      Q=Q+1
      P=P+1
   16 IF (P.GE.Q) GO TO 18
   17 XSTOR=XR(P)
      XR(P)=XR(Q)
      XR(Q)=XSTOR
      XSTOR=XI(P)
      XI(P)=XI(Q)
      XI(Q)=XSTOR
```

```
18 IF(Q.EQ.1) GO TO 27
19 Q=Q-1
   GO TO 15
27 IF (IC.EQ.1) SCALE = NP
   IF (IC.EQ.2) SCALE=2.0
   DO 28 I=1,N
   XR(I)=XR(I)/SCALE
28 XI(I)=XI(I)/SCALE
   IF (IC.EQ.2) GO TO 29
   DO 30 I=1,NP
   J = N+2 -I
   XR(I) = XR(I) * E(I)
   XI(I) = XI(I) * E(I)
   IF(I.GT.1) XR(J) = XR(J) * E(I)
30 IF(I.GT.1) XI(J) = XI(J) * E(I)
   GO TO 24
29 RETURN
   END
   SUBROUTINE REVERS(IU,K,IR,NU)
   DIMENSION IR(NU)
   J = IU
   K = ND = 0
   DO 1 I=1,NU
 1 IR(I)=0
 2 ND=ND+1
   IQ=J/2
   IR(ND)=J-2*IQ
   J=IQ
   IF (J.GT.0) GO TO 2
   DO 3 I=1,NU
 3 K=K+IR(I)*2**(NU-I)
   RETURN
   END
   SUBROUTINE FFTH(IN,IOUT,NS)
   DIMENSION IN(16,16),IOUT(16,16),XR(16),E(8)
   NU = 4
   N = 2**NU
   NP = N/2
   IF(NS .EQ. "YES") GO TO 1
   PRINT *,"ENTER ",NP," ENHANC. PRAMS."
   CALL BEEP(NP,.1,.1)
   READ *,(E(J),J=1,NP)
   CALL ERASE
   DO 2 I=1,N
   DO 3 J=1,N
 3 XR(J)=IN(I,J)
   CALL FFT(XR,NU,E)
   DO 4 K=1,N
 4 IOUT(I,K)=XR(K)+.5
 2 CONTINUE
   PRINT *,"FFT ",NP," HOR. ENH"
```

```
      PRINT 6,((J,E(J)),J=1,NP)
    6 FORMAT(8(1H ,"M",I1," = ",F4.1,/))
    1 RETURN
      END
      SUBROUTINE FFTHV(IN, IOUT, NS)
      DIMENSION IN(16,16),IOUT(16,16),XR(16),H(8),V(8)
      NU = 4
      N = 2**NU
      NP = N/2
      IF(NS .EQ. "YES") GO TO 1
      PRINT *,"ENTER ",NP," HORIZONTAL ENHANCERS. "
      CALL BEEP(NP, 0.1, 0.1)
      READ *,(H(J),J=1,NP)
      PRINT *,"ENTER ",NP," VERTICAL ENHANCERS. "
      CALL BEEP(NP, 0.1, 0.1)
      READ *,(V(J),J=1,NP)
      CALL ERASE
      DO 2 I = 1,N
      DO 3 J = 1,N
    3 XR(J)=IN(I,J)
      CALL FFT(XR,NU,H)
      DO 4 K = 1,N
    4 IOUT(I,K)=XR(K)+.5
    2 CONTINUE
      DO 5 I = 1,N
      DO 6 J=1,N
    6 XR(J)=IOUT(J,I)
      CALL FFT(XR,NU,V)
      DO 7 K =1,N
    7 IOUT(K,I) = XR(K)+.5
    5 CONTINUE
      PRINT *,"FFT HORIZ."
      PRINT 9,((J,H(J)),J=1,NP)
    9 FORMAT (8(1H ,"M",I1," = ",F4.1,/))
      PRINT *,"FFT VERT."
      PRINT 9,((J,V(J)),J=1,NP)
    1 RETURN
      END
```

APPENDIX B. PROGRAM FIG1

```
      PROGRAM FIG1(INPUT,OUTPUT,TAPE4=65,TAPE61=100,TAPE62=100)
      DIMENSION IX(56),IY(56),IN(16,16)
      DATA IX/1641,2986,2986,3154,3154,2658,2662,3094,
     &3138,3146,3162,3114,3082,3038,3018,3018,2490,2490,1981,1981,
     &1817,1817,1469,1469,1641,1641,1753,1753,1729,2994,2994,3162,
     &3162,2662,2662,3134,3146,3158,3130,3082,3050,3014,2998,2518,
     &2494,2494,1977,1977,1813,1813,1477,1477,1641,1641,1753,1753/
      DATA IY/1991,1991,2071,2071,2427,2427,2567,2811,
     &2815,2831,2875,2883,2879,2875,2855,2827,2599,2703,2703,2611,
     &2611,2435,2435,2079,2079,1991,1991,1991,408,404,492,492,
     &844,848,996,1204,1232,1268,1284,1288,1284,1272,1252,1020,
     &1020,1116,1116,1028,1028,848,848,488,488,408,408,408/
      CALL INITT(30)
      DO 6 I=1,16
      DO 6J=1,16
    6 IN(I,J)="B"
      DO 7 I=1,3
      IN(2+I,14-I)="G"
    7 IN(5,6+I)="C"
      DO 8 I=6,7
      DO 8 J=6,10
    8 IN(I,J)="T"
      DO 9 I=8,11
      DO 9 J=4,13
    9 IN(I,J)="H"
      DO 10 J=5,12
   10 IN(12,J)="W"
   51 CALL HOME
      CALL ANMODE
      DO 52 I=1,16
   52 PRINT53,(IN(I,J),J=1,16)
   53 FORMAT(1H ,17X,16A3)
      CALL MOVABS(IX(1),IY(1))
      DO 55 I = 2,28
   55 CALL DRWABS(IX(I),IY(I))
      CALL MOVABS(IX(29),IY(29))
      DO 56 I = 30,56
   56 CALL DRWABS(IX(I),IY(I))
      CALL ANMODE
      END
)E
```

DISTRIBUTION LIST

Director
US Army Research Office
ATTN: Library
P.O. Box 12211
Research Triangle Park, NC 27709

Director
Department of Defense
Advanced Research and Projects
  Agency
Washington, DC 20301

Commander
US Army Materiel Development &
  Readiness Command
ATTN: DRCSA-R
      DRCSA-PP
Washington, DC 20315

Commander
US Army Aviation Systems Command
12th & Spruce Streets
St. Louis, MO 63166

Commander
US Army Tank-Automotive Research &
  Development Command
ATTN: DRDTA, Technical Library
Warren, MI 48090

Commander
US Army Electronics Command &
  Devices Laboratory
Ft. Monmouth, NJ 07703

Commander
US Army Electronics Command
Atmospheric Science Laboratory
White Sands Missile Range
ATTN: DRSEL-BL-MS
      Mr. Richard Gomez
White Sands, NM 88002

Commander
US Army TECOM
ATTN: AMSTE-TA-A (2)
STEAP-TL
STEAP-DS-LP
Aberdeen Proving Ground, MD 21005

Commander
US Army Harry Diamond Laboratories
ATTN: AMXDO-TD/002
AMXDO-TIB
2800 Powder Mill Road
Adelphi, MD 20783

Commander
US Army ARRADCOM
ATTN: DRDAR-SC COL A. J. Larkins
DRDAR-SC Dr. D. A. Gyorog
DRDAR-SCF, L. Berman
DRDAR-SCF-I, J. Lehman
DRDAR-SCF-IO, J. Lester
DRDAR-SCF-IO, G. Sivak (12)
DRDAR-SCP, J. Glennon
DRDAR-MSA, R. Isakower
DRDAR-TSS (5)
Dover, NJ 07801

Defense Technical Information Center (12)
Cameron Station
Alexandria, VA 22314

Commander
US Army Armament Materiel and
Readiness Command
ATTN: DRDAR-LEP-L
Rock Island, IL 61299

Director
US Army TRADOC Systems
Analysis Activity
ATTN: ATAA-SL (Tech Lib)
White Sands Missile Range, NM 88002

US Army Materiel Systems
Analysis Activity
ATTN: DRXSY-MP
Aberdeen Proving Ground, MD 21005

Weapon System Concept Team/CSL
ATTN: DRDAR-ACW
Aberdeen Proving Ground, MD 21010

Technical Library
ATTN: DRDAR-CLJ-L
Aberdeen Proving Ground, MD 21010

Technical Library
ATTN: DRDAR-TSB-S
Aberdeen Proving Ground, MD 21005

Technical Library
ATTN: DRDAR-LCB-TL
Benet Weapons Laboratory
Watervliet, NY 12189

Dr. Thomas S. Huang
Institute for Image Technology
1000 North Western Ave.
West Lafayette, IN 47906