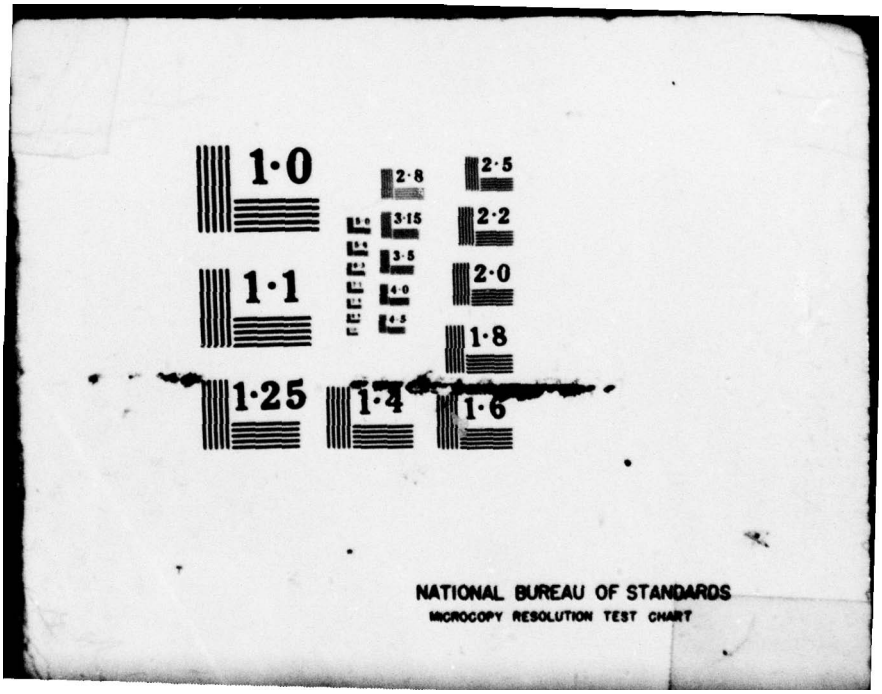
AD-A071 752 TELECOMMUNICATIONS ASSOCIATES FAIRFAX VA F/G 8/2
INVESTIGATION OF LINEAR TRANSFORMATIONS FOR AUTOMATIC CARTOGRAP--ETC(U)
APR 79 R L PICKHOLTZ, M MOVAHED, S S MURTY DAAK70-78-C-0045
UNCLASSIFIED TA-79-1-1 ETL-0181 NL

1 OF 2
AD
A071752


1·0
1·1
1·25
2·8
3·15
3·5
4·0
4·5
1·4
2·5
2·2
2·0
1·8
1·6
NATIONAL BUREAU OF STANDARDS
MICROCOPY RESOLUTION TEST CHART

ETL - 0181

LEVEL 10

Investigation of Linear Transformations for Automatic Cartographic Analysis

Telecommunications Associates

1 April 1979

Final report for period April 1978 - April 1979

Approved for public release; distribution unlimited.

Prepared for

US Army Engineer Topographic Laboratories

ETL - R1

Fort Belvoir, Virginia 22060

(9) Final rept. Apr 78 - Apr 79,

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

(19) (14) REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER ETL - 0181	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
(6) TITLE (and Subtitle) Investigation of Linear Transformations for Automatic Cartographic Analysis,		5. TYPE OF REPORT & PERIOD COVERED Final 4/78 - 4/79
		6. PERFORMING ORG. REPORT NUMBER TA - 79 - 1 - 1
(10) AUTHOR(s) R.L. PICKHOLTZ M. MOVAHED S.S. MURTY		8. CONTRACT OR GRANT NUMBER(s) DAAK 70 - 78 - C - 0045
9. PERFORMING ORGANIZATION NAME AND ADDRESS Telecommunications Associates, 3613 Glenbrook Rd, Fairfax, Va 22031		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 6.11.02A (16) 4A16110 2B52C ~~1852C20012~~
11. CONTROLLING OFFICE NAME AND ADDRESS Engineer Topographic Laboratories Fort Belvoir, Va 22060		(11) 12. REPORT DATE 1 April 1979 13. NUMBER OF PAGES 182
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) Same (12) 189p		15. SECURITY CLASS. (of this report) Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

(15) DAAK70-78-C-0045

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

(14) TA-79-1-1

18. SUPPLEMENTARY NOTES

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

Transforms, Image Analysis, Cartography

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

This report contains a summary of the investigation in applying linear transforms such as Fourier, Bessel, Walsh-Hadamard, Slant and Discrete Cosine to cartographic analysis. Recommendations are made and hardware and software implementations are proposed.

411301

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

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

ABSTRACT

This report contains a summary of the investigation in applying linear transforms such as Fourier, Bessel, Walsh-Hadamard, Slant and Discrete Cosine to cartographic analysis. Recommendations are made and hardware and software implementations are proposed.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

CHAPTER I

INTRODUCTION

The field of image processing is very vast and has grown considerably during the past decade. The development of ever faster and smaller processors, increasing resolution and on-chip processing sensors, larger capacity mass memories, wider bandwidth communication channels and more sophisticated digital processing stations have provided equipment devices, and techniques that five years ago were beyond expectation. Image processing has found a significant role in scientific, industrial, biomedical, space and government applications. Being a vast field, image processing can be divided into different subfields relating to different aspects. Image transforms, Image coding, Image enhancement, Image restoration, feature extraction and image understanding are all different aspects of image processing.

This report is devoted to one of the aspects of image processing, namely the investigation of linear transformations for automatic cartographic analysis and feature extraction. Aerial photographs are pictures taken from some distance above the earth. They usually contain images of a variety of objects. These objects have different characteristics and so can be classified according to some specific properties. For example roads and rivers can be put into one group, since on a picture they usually appear as a line. After this categorization has been done, an attempt can be made to enhance some of these features automatically by the aid of some mathematical transforms. The actual recognition of specific cartographically important objects requires a non-linear detection process. The object recognition process is not addressed in this picture.

A picture is a flat object whose appearance varies from one point to another point. In a " black and white " picture this variation can be described by a single parameter corresponding to the total amount of light reaching the

observer from the given point. In accordance with the above remarks, we can define a picture as a two-dimensional light intensity function f(x,y), where x and y denote spatial coordinates and the value of f at any point (x,y) is proportional to the brightness (gray level) of the picture at that point. The picture, therefore, is a non-coherent image of the object.

A digital picture is a picture f(x,y) which has been discretized both in spatial coordinates and in brightness. In this content a digital picture can be shown by a matrix whose row and column indices identify a point in the picture and the corresponding matrix element value identifies the gray level at that point. The element of such a digital array are called picture elements or pixels.

A picture transform takes a picture from the spatial domain into another functional domain. The recognition of some cartographic features may be facilitated by using such picture transforms. For this purpose instead of searching for an object in the original image, its transform is taken and it is hoped that certain salient features or patterns will be exhanced. The transform selected should have a strong response to the characteristic features of interest while being invariant to miscellaneous parameters (such as size, translation , etc.) and insensitive to the general background which is of little or no interest. Since different mathematical transforms have different properties, it will be interesting to investigate their properties and see how they act on different functions. For example, the much used Fourier transform, in which sines and cosines are the kernel functions, brings out periodic structures of the image quite well. However, the Fourier transform is not optimal or even suitable for all image features. Circular objects are best identified with the Bessel transform for example, while other features may be more efficiently

identified with the Walsh or Haar transforms. The task is to determine which transform, if any, is best suited for identifying a given terrain features. The paramount consideration in selecting a transform shall be its success in identifying a given object but other factors will be considered, such as execution speed with digital and analog processors.

To follow the pattern of the above discussion, this paper is comprised of six chapters.

Chapter one, the present chapter, contains some introductory discussions. At the end of this chapter, the objects usually present in an aerial imagery are classified.

Chapter two gives a review of continuous transformations. Continuous transformations apply to continuously dense picture elements and supply a rich repertoire of analytic techniques and results. Fourier transform, Hankel transform, Matched filtering process and some novel extrapolation techniques are discussed in section A of this chapter. In section B of chapter two some candidate features from the objects usually present in photographs are chosen and used to illustrate the application of the theory. The transformations discussed in section A are applied to those candidate features to enhance the characteristics defined by their analytical properties. Some examples of two-dimensional matched filters are also given in section B of chapter two.

One continuous transform, the Karhunen- Loéve transform, was studied in interim report # 4. This transform deals with the statistics of an image rather than its exact shape or geometry. This transformation has been mainly proposed

for other uses in image processing rather than feature extraction. Since the results of our study on the Karhunen - Loéve transform didn't look promising, it is not included in this report.

Chapter three follows the same pattern of chapter two but for discrete transformations. A review of the analytical properties of Walsh-Hadamard transform (WHT), Haar transform (HT), Slant transform (ST), and Discrete-cosine transform (DCT) is given in section A of this chapter. In section B some examples with potential cartographic applications are examined .

In looking for an object feature image in a given picture, there is not usually any a priori information about the orientation or the size of the object. For this reason, it is very desirable to see how the result of a transform is affected by translation, rotation or scaling of the original object image. In chapters two and three we have tried to consider and show those effects for different transforms. This is done both analytically and by example.

In chapter four, a software package is developed for the evaluation of two-dimensional (WHT), (HT), (ST), and (DCT) in processing the data pixels. The entire program is written in Fortran IV language. This package is used for various examples of cartographic features and the results are given in Appendix III.

In chapter five, a study of hardware feasibility of a signal processing system is made. In this chapter, a special purpose signal processor for cartographic studies is proposed. The processor makes use of the two-dimensional Hadamard/Haar/Slant/Discrete Cosine transforms. An assessment is made of the speed and of the cost.

In chapter six, some conclusions and recommendations have been made based upon our transforms study in the first five chapters. Those conclusions and recommendations are merely based on the analytical properties of the transforms studied or on the examples used to illustrate those properties. The conclusions are at best, estimates and should not be taken as the last word. It is very possible that a deeper look into the subject may contradict some of our conclusions. The above remarks should be taken as a caveat to the reader.

Thus our interpretations and value judgements, while given in order to share the insight we gained while carrying out these studies, must be tempered with the recognition that they are based on a limited analysis and a still incomplete understanding of the picture interpretive process.

CLASSIFICATION OF OBJECTS USUALLY PRESENT IN AERIAL IMAGERY

In order to maximize the likelihood of success in using linear transforms, it will be necessary to classify the principal attributes of particular cartographically interesting objects. This problem can be possibly approached from different ways. A good choice is to categorize the cartographic objects according to the dimensionality of their images. While the image of a road appears as a line, the image of a lake covers some area of the picture. So according to the feature dimension, the features that are usually present in an aerial cartograph can be divided into three different groups, point features, line features and area features.

The distinction between points, lines and areas depends on : (1) the distance from which the picture has been taken. (2) the degree of spatial discretization of the picture. For example, the image of a storage container for oil

will appear as a point if the picture is taken from a far distance. However, a picture of the same scene taken from a closer distance can have the image of the container shown as a disk. On the other hand, suppose we discretize an 8" X 8" picture into an 8 X 8 square array (see chapter three), then a circle of radius 0.5 " can appear only as a point or not appear at all. However, if the picture is discretized into an 80 X 80 square array, the same circle will cover some area of the digital picture. So we should have in mind that when we compare the features according to their dimensionalities, the comparison will be true only for features present in the same picture or for pictures taken from almost the same distance and with the same degree of discretization.

POINT FEATURES

Images of objects of small size (small length and small width) appear as points on an aerial photograph. Examples of these objects are isolated buildings, storage thanks, tunnel entrances, etc. Point features on an image can be different in brightness suggesting the presence of different objects. A storage container made of metal will surely appear brighter than one made of concrete. Point features may appear as a set on a picture and the set can have some special characteristic. For example, if a grounded aircraft is taken as a point feature, then one squadran of grounded aircrafts will exhibit point periodicities on the image. Another example can be the storage containers for oil and gas which usually have a clustering symmetry among themselves.

LINE FEATURES

Any object whose length is much longer than its width will appear as a line

on an aerial cartograph. Rivers, roads, transmission lines, bridges, etc. are all examples of the so-called line features. While some line features like a dam appear as a segment, others like roads or rivers may extend through the whole picture. The line features are usually different in characteristics, for example, while runways are invariably straight and frequently have intersections, roads and rivers usually are expected to have unbounded extent and will likely exhibit parallelism and meandering symmetries, respectively.

There is a quite nice distinction between the line features that are man-made and those which are natural. The former one usually shows some kind of regularity or straightness while the latter one is not so neat. For example, if a pipeline is to be extended between two points on a rather flat surface, it will certainly be built as a straight line, but a river between those two points can follow some other arbitrary direction. Another property which can be only related to those man-made line features is the possibility of having a set of them with some special properties among the members. For example, parallelism is a characteristic of some man-made line features like dual highways or transmission lines.

AREA FEATURES

The objects whose images cover a relatively large portion of a picture are considered as area features. These features can have any arbitrary shape depending on the nature of their construction. A lake, a cemetery, an urban area, etc., are all examples of area features. Shape, size, and convexity are the features that make a distinction between different area features. Neat geometrical shape or symmetry is usually the characteristic of man-made areas. For example, a field can show rectangular symmetry while a large container may show circular

symmetry. Most of man-made area features have some internal properties if examined in close detail. For example, towns will have internal straight line parallelism due to streets and point periodicities due to housing developments.

A list of the objects usually present on cartographs is given on the following page.

CARTOGRAPHIC FEATURES

POINT FEATURES

Isolated buildings

Storage tanks

Quarry or borrow pit

Tunnel entrance

LINE FEATURES

Rivers and streams with water

Drainage channels without water

Canals

Dual highways

Primary roads

Secondary roads

Unpaved roads and trails

Transmission lines

Pipe lines

Levees

Dam

Rapids and falls

Bridges

Shoreline (large water body)

Airport

AREA FEATURES

Large rivers (water on left and water on right)

Lakes

Wash

Forest

Scrub

Marsh and swamp

Mangrove

Orchard and Vineyard

Urban area

Suburban area

Industrial area

Railroad yard

Cemetery

CHAPTER II

CONTINUOUS TRANSFORMS

In this chapter, some two-dimensional continuous linear transformations will be studied. Continuous transformations are those transformations that apply to continuously dense picture elements. Since a picture is a function of two real variables, so all the transforms used in picture processing will be two dimensional. In general, a two-dimensional continuous transform is given by,

$$F(\xi,\eta) = \int_{-\infty}^{+\infty}\int K(x,y,\xi,\eta)\ f(x,y)\ dx\ dy$$

in which $f(x,y)$ is the input function in spatial domain, $F(\xi,\eta)$ is the function in the transformed domain and $K(x,y,\xi,\eta)$ is the kernel of the transform. The integral is taken over the whole domain in which the function exists. Fourier and Hankel transforms, discussed in this chapter, are examples of two-dimensional continuous transforms. In section 2A, the analytical properties of these transforms have been investigated and in section 2B some examples have been used to illustrate their properties.

In the second part of this chapter, the two-dimensioanl matched filtering process is discussed. Matched filters are used to detect object images in a picture. Although matched filtering can be considered as a linear transform, there is a significant difference between matched filtering and transforms. In a linear transform, the kernel of the transform is fixed and unique for any input function. In matched filtering, the kernel is matched to the object to be detected (which needs to be known), and so varies from one case to another. Analytical properties of the two-dimensional matched filter have been discussed in section 2A along with some examples in section 2B. Some novel ideas for allowing the matched filter to be used in cases when the object is not exactly known are also discussed.

In the last part of this chapter, some new extrapolation techniques are discussed. It has been shown that in the special case of bandlimited signals, one can extrapolate a two-dimensional function in terms of a known part. It is suggested that this technique may be used to exhance a hidden part of a picture using the information conveyed by the rest of that picture.

The main purpose of studying continuous transforms in this chapter is to see if they can be used in cartography. Some transforms act better on a particular feature than others. This behaviour of a transform is usually manifested by the characteristics of its kernel. In this chapter, we will try to show which transform is most suitable for extracting a particular feature.

In extracting a feature from aerial imagery, we usually look for the image of an object when there is no a priori information about its orientation or size. For this reason, it will be instructive to see how a particular transform behaves, as an object image is translated, rotated or scaled. In our study of different transforms in this chapter, a try has been made to answer the above question.

2A : THEORY

FOURIER TRANSFORM

In Fourier transform, sines and cosines are the kernel functions. The most suitable transform for extracting a particular feature, is a transform whose kernels are as closely matched to that feature as possible. Since sines and cosines are periodic functions, even before going into details of analytical properties of Fourier transform, we can predict that this transform will act quite well on periodic structures in a photograph. Parallel streets in an aerial pic-

ture of an urban area, is an example of such periodicity.

Let $f(x,y)$ be a function of two independent variables x and y, then its Fourier transform $F(u,v)$ is defined by,

$$F(u,v) = \int_{-\infty}^{+\infty}\int f(x,y)\ \exp[-j2\Pi(ux + vy)]\ dx\ dy \qquad (2A\text{-}1)$$

The inverse Fourier transform of $F(u,v)$ is given by,

$$f(x,y) = \int_{-\infty}^{+\infty}\int F(u,v)\ \exp[j2\Pi(ux + vy)]\ du\ dv \qquad (2A\text{-}2)$$

The pair of functions $f(x,y)$ and $F(u,v)$ related by equations (2A-1) and (2A-2) will be denoted by

$$f(x,y) \Longleftrightarrow F(u,v) \qquad (2A\text{-}3)$$

The Fourier transform of $f(x,y)$ may not exist unless f satisfies certain conditions [1] . The main condition for $f(x,y)$ is to be continuous and square integrable.

The Fourier transform of a real function is generally complex.

$$F(u,v) = R(u,v) + j\ I(u,v) \qquad (2A\text{-}4)$$

$$= \left|F(u,v)\right| \exp[j\Phi(u,v)] \qquad (2A\text{-}5)$$

where,

$$F(u,v) = [\ R^2(u,v) + I^2(u,v)]^{1/2} \qquad (2A\text{-}6)$$

and

$$\Phi(u,v) = \tan^{-1}\left[\frac{I(u,v)}{R(u,v)}\right] \qquad (2A\text{-}7)$$

The magnitude function $\left|F(u,v)\right|$ is called the Fourier spectrum of $f(x,y)$ and $\Phi(u,v)$ its phase angle. The square of the spectrum

$$E(u,v) = \left|F(u,v)\right|^2 \qquad (2A\text{-}8)$$

is referred to as the energy spectrum of $f(x,y)$

Some properties of the two-dimensional Fourier Transform :

Suppose we have,

$$f(x,y) \Longleftrightarrow F(u,v)$$

then

$$f(x - x_0,\ y - y_0) \Longleftrightarrow F(u,v)\exp[-j2\Pi(ux_0 + vy_0)] \qquad (2A\text{-}9)$$

It is interesting to note from equation (2A-9) that a shift in $f(x,y)$ does not affect the magnitude of its Fourier transform,

$$\left|F(u,v)\ \exp[-j2\Pi(ux_0 + vy_0)]\right| = \left|F(u,v)\right| \qquad (2A\text{-}10)$$

This is a very good result since visual examination of the transform is usually limited to a display of its magnitude and so Fourier spectrum possess translational invariance.

If we introduce polar coordinates,

$$x = r\,\mathrm{Cos}\theta \qquad u = w\,\mathrm{Cos}\phi$$
$$y = r\,\mathrm{Sin}\theta \qquad v = w\,\mathrm{Sin}\phi \tag{2A-11}$$

and if we have

$$f(x,y) \Longleftrightarrow F(u,v)$$

then

$$f(r,\theta) \Longleftrightarrow F(w,\phi) \tag{2A-12}$$

Now if the coordinates of $f(r,\theta)$ are rotated by the amount θ_o it can be shown that,

$$f(r,\theta + \theta_o) \Longleftrightarrow F(w,\phi + \theta_o) \tag{2A-13}$$

The above result is very interesting and important for cartography since it shows that if $f(x,y)$ is rotated by an angle θ_o, then $F(u,v)$ is rotated by the same angle.

Some other useful properties of the Fourier transform pair are,

$$a\,f(x,y) \Longleftrightarrow a\,F(u,v) \tag{2A-14}$$

$$\text{(linearity)}\quad f_1(x,y) + f_2(x,y) \Longleftrightarrow F_1(u,v) + F_2(u,v) \tag{2A-15}$$

and,

$$\text{(scaling)}\quad f(ax,by) \Longleftrightarrow \frac{1}{ab} F(u/a\ ,\ v/b) \tag{2A-16}$$

From equation (2A-16), one can find the Fourier transform of a scaled version of the function f(x,y), [see appendix I] as,

$$f(kx,ky) \Longleftrightarrow \frac{1}{k^2} F\left[\frac{u}{k}, \frac{v}{k}\right] \qquad (2A\text{-}17)$$

The above relation shows that the Fourier transform of a scaled version of a funtion is a scaled version of the Fourier transform of the function multiplied by the squared inverse of the scaling factor. As a result the shape is preserved.

For a more detailed study of Fourier transform, see (1), (2), and (3).

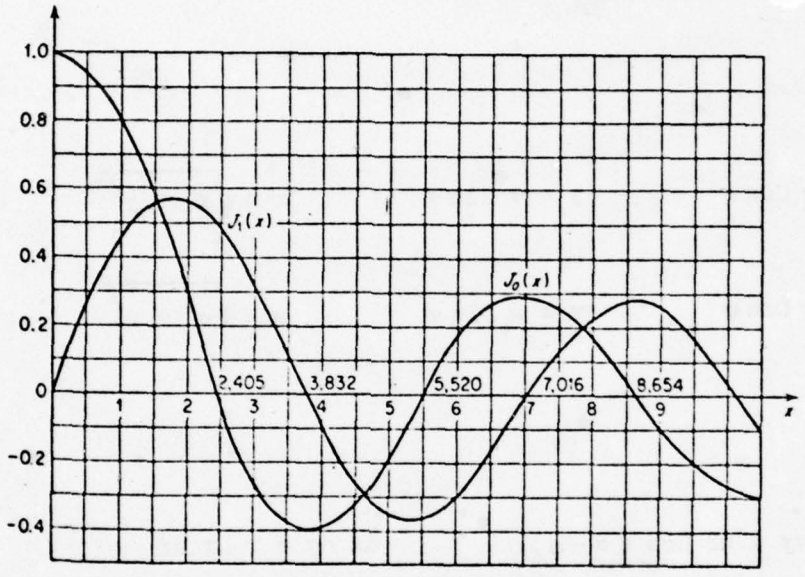
HANKEL TRANSFORM

One of the features usually present on an aerial photograph is the storage container. A single storage container appears as a centered circle or as a disk. Storage containers can be located as a cluster. Sometimes this cluster is circularly symmetric. The Hankel transform discussed in this section appears very useful in bringing out the object images with circular symmetry. This property is based on the presence of the Bessel function as the kernel of Hankel transform.

Given a function f(r) and a real constant w, we form the integral,

$$F(w) = \int_0^\infty r\, f(r)\, J_0(wr)\, dr \qquad (2A\text{-}18)$$

This integral defines the Hankel transform F(w) of f(r). In equation (2A-18), $J_0(x)$ is the Bessel function of order zero as ahown in Fig. (2A-1)



FIG(2A-1) Bessel functions of order 0 and 1

If a two-dimensional signal has circular symmetry as, e.g. a circle, then its Fourier transform reduces to a Hankel transform. So the theory of Hankel transforms can be studied as a special case of the corresponding theory of Fourier transforms.

Suppose,

$$f(x,y) = f(\sqrt{x^2 + y^2}) \tag{2A-19}$$

then with,

$$\begin{aligned} x &= r \cos\theta \qquad y = r \sin\theta \qquad r = \sqrt{x^2 + y^2} \\ u &= w \cos\phi \qquad v = w \sin\phi \qquad w = \sqrt{u^2 + v^2} \end{aligned} \tag{2A-20}$$

we have,

$$ux + vy = wr \cos(\theta - \phi) \qquad dx\,dy = r\,dr\,d\theta$$

Inserting the transformations into the integral form of Fourier transform which defines $F(u,v)$, we will have,

$$F(u,v) = \int_0^\infty rf(r) \int_{-\Pi}^{+\Pi} e^{-jwr \cos(\theta-\phi)} d\theta\, dr = 2\Pi \int_0^\infty rf(r)\, J_0(wr)\, dr \tag{2A-21}$$

$$= 2\Pi F(w)$$

It can be shown that the inversion formula for Hankel transform is,

$$f(r) = \int_0^\infty w\, F(w)\, J_0(rw)\, dw \tag{2A-22}$$

By looking at the Hankel transform as a special case of a two-dimensional

Fourier transform, all of its properties can be derived from the corresponding properties of Fourier transform. The most important observable fact is that $F(w)$ has circular symmetry. This means that the Fourier/Hankel transform of a circularly symmetric function will be circularly symmetric. Reference (2) gives a very detailed study of Hankel transforms.

MATCHED FILTERING

One of the major problems in communications engineering is to detect a signal which has been degraded by noise. The matched-filter implemented by electrical circuits, is widely used as a means of one-dimensional signal detection. This filter is optimum in minimizing the signal-to-noise ratio (SNR) and has been studied in detail (4-5).

The two-dimensional version of the matched filter which is a straight forward extension of the one-dimensional case can be used for detecting objects within images.

Let $f(x,y)$ be the desired object to be detected. Since practically any image is accompanied by some kind of noise, on observation the image containing the object $f(x,y)$ can be shown as a sum of two parts. The first part is the object itself, and the second part is some additive noise. If we show the observed image by $g(x,y)$ and the noise by $n(x,y)$ we have,

$$g(x,y) = f(x,y) + n(x,y) \qquad (2A\text{-}23)$$

The noise $n(x,y)$ can be of any kind, it may be the background of a picture containing the desired object, or it can be the noise of the communication

channel over which the image has been transmitted or both. The assumption about the additive nature of the noise is usually met in practice with good approximations since no detailed statistical assumption such as Gaussianess is necessary.

To detect $f(x,y)$ from $g(x,y)$, we pass $g(x,y)$ through a linear filter whose impulse response (point spread function in optics) is $h(x,y)$, and we try to find $h(x,y)$ in such a way that the signal-to-noise ratio would be maximum at the output of this filter.

If the signal $g(x,y)$ is given to the input of a filter whose inpulse response is $h(x,y)$, the output will be,

$$g(x,y) * h(x,y) = f(x,y) * h(x,y) + n(x,y) * h(x,y) \qquad (2A\text{-}24)$$

in which * denotes the convolution operation. This is shown graphically in Fig. (2A-2).

The instantaneous signal power at the filter output is given by,

$$S = \left| f(x,y) * h(x,y) \right|^2 \qquad (2A\text{-}25)$$

$$= \left| \int_{-\infty}^{+\infty} \int F(u,v)\, H(u,v)\, \exp[j(u\xi + v\eta)]\, du\, dv \right|^2 \qquad (2A\text{-}26)$$

$F(u,v)$ and $H(u,v)$ are respectively the Fourier transforms of $f(x,y)$ and $h(x,y)$. The noise power at the filter output is, (6)

$$N = \int_{-\infty}^{+\infty} \int S_n(u,v) \left| H(u,v) \right|^2 du\, dv \qquad (2A\text{-}27)$$

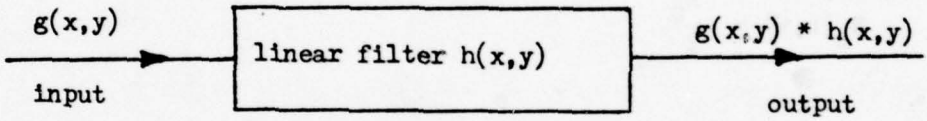


Fig. (2A-2) A linear filter.

in which $S_n(u,v)$ is the power spectrum of $n(x,y)$. The signal-to-noise ratio at the filter output is,

$$\frac{S}{N} = \frac{\left| \int_{-\infty}^{+\infty}\int F(u,v)\ H(u,v)\ \exp\left[j(u\xi + v\eta)\right] du\ dv \right|^2}{\int_{-\infty}^{+\infty}\int S_n(u,v)|H(u,v)|^2\ du\ dv} \tag{2A-28}$$

using the Schwartz inequality (5), we find the S/N is maximum for

$$H(u,v) = \frac{F^*(u,v)\ \exp\left[-j(u\xi + v\eta)\right]}{S_n(u,v)} \tag{2A-29}$$

If the noise is white,

$$S_n(u,v) = \frac{N_0}{2} \tag{2A-30}$$

and

$$H(U,V) = \frac{2}{N_0} F^*(u,v)\ \exp\left[-j(u\xi + v\eta)\right] \tag{2A-31}$$

or

$$h(x,y) = \frac{2}{N_0} f(\xi - x, \eta - y) \tag{2A-32}$$

As we see in the case of additive white noise, the optimum filter impulse response is an amplitude scaled version of the signal image field rotated by 180° and so the term <u>matched filter</u> is used.

For the case of colored noise, the impulse response of the matched filter was given by Eq. (2A-29). This is equivalent to the white noise matched filter preceded by a whitening filter of system response proportional to the inverse of the noise spectrum. For more on this subject see (7).

The matched filter output is given by,

$$g_0(x,y) = g(x,y) * h(x,y)$$

$$= \int_{-\infty}^{+\infty}\int g(\alpha,\beta)\ h(x-\alpha,\ y-\beta)\ d\alpha\ d\beta \qquad (2A\text{-}33)$$

$$= \frac{2}{N_0} \int_{-\infty}^{+\infty}\int g(\alpha,\beta)\ f(\xi+\alpha-x, \eta+\beta-y)\ d\alpha\ d\beta \qquad (2A\text{-}34)$$

Looking at Eq.(2A-26), we see that the matched filter output was maximized at the point (ξ,η). At this point the output is,

$$g_0(\xi,\eta) = \frac{2}{N_0} \int_{-\infty}^{+\infty}\int g(\alpha,\beta)\ f(\alpha,\beta)\ d\alpha\ d\beta \qquad (2A\text{-}35)$$

As we see at this point, the matched filter output is simply the correlation between the image and the desired object to be detected. The point (ξ,η) is called the matched filter <u>offset</u> . Ordinarily, the parameters (ξ,η) of the matched filter transfer function are set to zero so that the origin of the output plane becomes the point of no translational offset between $g(x,y)$ and $f(x,y)$.

In using matched filters for feature extraction, and specially if the output is going to be judged by visual examination, it is always advantageous to apply thresholding to the filter output. This action eliminates the possibility of " false detection " or " miss " by giving a bright spot in a dark background where the desired object is present on the input image.

Examples of candidate objects to be detected along with the matched filter transfer functions which are matched for detecting those objects and the corresponding outputs are shown in Fig.(2B-4).

SOME PROPERTIES OF THE TWO-DIMENSIONAL MATCHED FILTER :

Suppose that a matched filter is matched to the signal $f(x,y)$ and its offset has been chosen to be the origin of the output plane. Then,

$$h(x,y) = \frac{2}{N_0} f(-x,-y) \tag{2A-36}$$

and the matched filter output will be,

$$g_o(x,y) = \frac{2}{N_0} \int_{-\infty}^{+\infty}\int f(\alpha,\beta)\, f(\alpha - x, \beta - y)\, d\alpha\, d\beta + \frac{2}{N_0} \int_{-\infty}^{+\infty}\int n(\alpha,\beta)\, f(\alpha - x, \beta - y)\, d\alpha\, d\beta \tag{2A-37}$$

in which the SNR peak occurs at $x = 0$ and $y = 0$. Now if the matched filter input contains a translated version of the object image, we have,

$$g(x,y) = f(x + \Delta x,\ y + \Delta y) + n(x,y) \tag{2A-38}$$

and the matched filter output is,

$$g_0(x,y) = \frac{2}{N_0} \int_{-\infty}^{+\infty}\int f(\alpha + \Delta x,\ \beta + \Delta y)\, f(\alpha - x, \beta - y)\, d\alpha\, d\beta + \frac{2}{N_0} \int_{-\infty}^{+\infty}\int n(x,y)\, f(\alpha - x, \beta - y)\, d\alpha\, d\beta \tag{2A-39}$$

In this case, as seen from Eq. (2A-39), the SNR peak occurs at $x = \Delta x$ and $y = \Delta y$, thus indicating the translation of the input image relative to the reference image. Hence the matched filter is translation invariant and any point (ξ,η) at which the correlation peak occurs in the output plane describes

the position of the detected two-dimensional image.

Convolution of the input signal with the matched filter response searches the entire image plane for a possible match and so if there are multiple targets of the form,

$$\sum_i f(x - \Delta x_i , y - \Delta y_i) \qquad (2A\text{-}40)$$

all of them will be detected by use of the same matched filter. Fig.(2B-5) gives an illustration of the above fact.

The matched filter is not rotation invariant [Interim Rep. #3]. However there may be rotated versions of an object image in a photograph. We like to know how a rotated version of an object image can be detected by the same matched filter . Suppose that a matched filter is matched to the signal $f(\rho , \theta)$ (polar coordinates). A rotated version of this signal which is represented by $f(\rho, \theta + \phi)$ cannot be detected by the same matched filter. The only way to get a peak is by rotating the matched filter itself. The impulse response of the rotated matched filter will be,

$$h'(\rho , \theta , \phi) = h(\rho , \theta + \phi)$$

$$= f(-\rho , \theta + \phi) \qquad (2A\text{-}41)$$

Consequently, if it is also desired to detect rotated versions of $f(\rho , \theta)$, the impulse response of the filter should be rotatable. The value of ϕ in Eq.(2A-41) is varied until a maximum for SNR is obtained Fig.(2A-3)

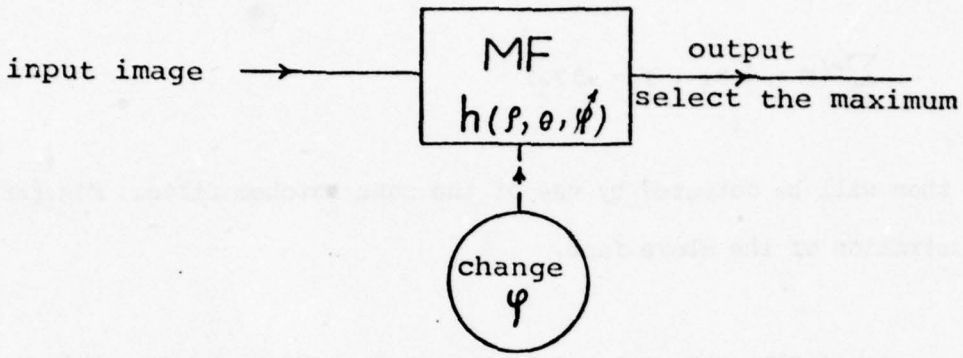


Fig (2A-3) A matched filter to detect the rotated version of an object.

At this point, let's investigate how a matched filter which is matched to a signal f(x,y) can be used to detect a scaled version of f(x,y). The size of an object image in a photograph depends on the distance from which the picture has been taken. Although the shape of an object may be preserved in pictures taken from different distances its size will be different in different pictures. For this reason we are interested to know how a matched filter can be modified to detect scaled versions of an object image.

In [Appendix 1], a scaled version of a function f(x,y) has been defined as f(kx , ky). A matched filter whose impulse response is given by,

$$h(x,y,k) = \frac{2}{N_0} f(\xi - kx,\ \eta - ky) \qquad (2A\text{-}42)$$

can detect scaled versions of the image f(x,y). The matched filter is built in a way that parameter k can be varied. The output is examined in a maximum likelihood sense and the largest output is selected. Fig. (2A-4).

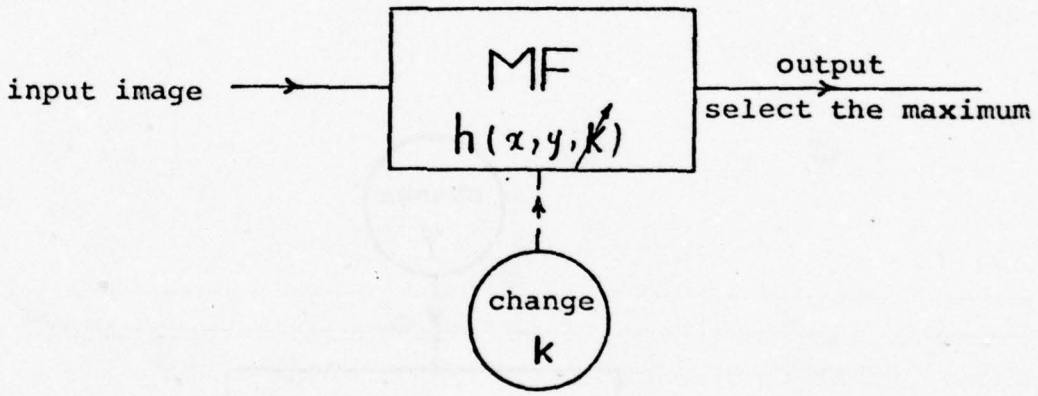


Fig (2A-4) A matched filter to detect the scaled version of an object.

The assumption discussed in Appendix 1 , that the origin of x - y plane, should be inside the curve, does not result in loss of generality. Since the matched filter is translation invariant, it detects the scaled version of f(x,y), even if it does not encircle the origin.

A matched filter with two variable parameters can detect any scaled, rotated or translated version of the image f(x,y). The transfer function of such a filter in polar coordinates will be,

$$h(\rho, \theta, k, \phi) = \frac{2}{N_0} f(-k\rho, \theta + \phi) \qquad (2A\text{-}43)$$

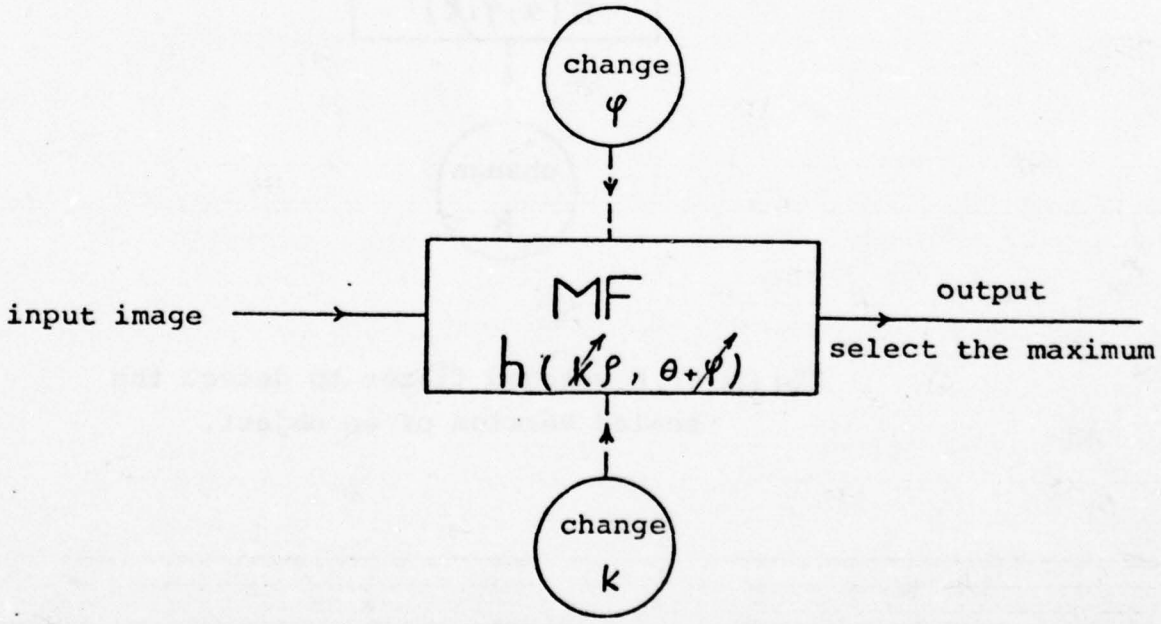


Fig.(2A-5) A matched filter to detect the translated, rotated or scaled version of an image.

The normal matched filter discussed in this section has a basic limitation which is especially important in feature extraction. The output of this matched filter is primarily dependent on the energy of the images rather than their spatial structure. A hexagonal object can be easily mistaken by a circle of the same brightness and the same size. In general normal matched filter provides relatively poor discrimination between objects of different shape, but of similar size or energy content.

Suppose that in a given picture, we are looking for a disk with radius A and constant gray level C. If the center of this disk is located at a point (ξ, η), we have

$$f(x,y) = C \quad \text{for} \quad \sqrt{(x-\xi)^2 + (y-\eta)^2} \leq A \tag{2A-44}$$

A normal matched filter designed to detect this signal, will have the same structure and will give us a peak of the amount $C^2 \Pi A^2$ at the point (ξ, η). As we have mentioned before, matched filtering is usually followed by thresholding, so that the positions of the peaks are detected more clearly. The above result would be excellent if in the output picture (processed by the matched filter) all other gray levels are smaller than $C^2 \Pi A^2$, namely

$$h(x,y) * f(x,y) < C^2 \Pi A^2 \qquad \text{for} \quad x \neq \xi, \; y \neq \eta \tag{2A-45}$$

then thresholding at a rate $C^2 \Pi A^2 - \epsilon$ (small ϵ) will locate the center of the circle. But in a real picture, there may be disks of different gray levels and different sizes and even other configuration which after being processed by the matched filter in (2A-44), would give larger peaks than $C^2 \Pi A^2$. How can we discriminate among these peaks and choose the one related to our desired circle ?

Certainly there is not a unique technique to avoid all these problems, and moreover some of them are related to the shortcoming of the normal matched filter itself.

To overcome the problem of getting some peaks larger than the one corresponding to our signal, a normalization technique on the original image will be useful. Suppose before using the matched filter in (2A-44), we normalize the picture as follows,

$$f(x,y) = \begin{cases} f(x,y) & \text{for} \quad f(x,y) \leq C \\ C & \text{for} \quad f(x,y) > C \end{cases} \tag{2A-46}$$

The above normalization will guarantee that, no part of the picture will have a gray level above C, and in the output the largest possible peak corresponding to our desired object. The value of C can be chosen as the average gray level value of the picture.

Even using the normalization technique in (2A-46) one cannot be sure that a peak of value $C^2 \Pi A^2$ will correspond to a disk with radius A and gray level C.

To overcome all the difficulties, related to the normal matched filter, one may think of applying an edge detection technique to a picture before it would be processed by the matched filter, In this manner, the edge structure of an object will be used for its detection.

Suppose f(x,y) is an object to be detected, its gradient is given by,

$$\nabla f(x,y) = \frac{\delta f}{\delta x} \hat{x} + \frac{\delta f}{\delta y} \hat{y} \qquad (2A\text{-}47)$$

the squared magnitude of this gradient, $(\frac{\delta f}{\delta x})^2 + (\frac{\delta f}{\delta y})^2$ can be used as an edge detector. This edge detector will work excellent if the gray level is constant across the object. If F(u,v) is the Fourier transform of f(x,y),

$$f(x,y) \Longleftrightarrow F(u,v) \qquad (2A\text{-}48)$$

then

$$\left.\begin{aligned} \frac{\delta}{\delta x} f(x,y) &\Longleftrightarrow -iu\, F(u,v) \\ \frac{\delta}{\delta y} f(x,y) &\Longleftrightarrow -iv\, F(u,v) \end{aligned}\right\} \qquad (2A\text{-}49)$$

and

$$(\frac{\delta f}{\delta x})^2 + (\frac{\delta f}{\delta y})^2 = -(u^2 + v^2)\, F(u,v) \qquad (2A\text{-}50)$$

and the desired matched filter will have an impulse response given by,

$$H_G(u,v) = -(u^2 + v^2)\, H_n(u,v) \qquad (2A\text{-}51)$$

where $H_N(u,v)$ is the impulse response of the corresponding normal matched filter. Since matched filtering is usually done on a digital computer, the implementation of an edge detecting matched filter will not be much more difficult than a normal matched filter.

The introduction of the gradient matched filter brings about another difficulty which was not much severe in using the normal matched filter. The problem is that since we do not usually have any a priori information about the orientation of the object to be detected, we have to build our matched filter with the capability of being rotated. This subject was discussed before. The whole

process is done on a digital computer and the angle of rotation is changed discretely until a large peak is detected.

Using a gradient matched filter, the above process becomes more difficult, because supposing that there is a rotated version of the target in a picture under consideration, it is very improbable that it will overlap our matched filter precisely, no matter how fine we choose the steps in rotating the matched filter. See Fig. (2B-7)

The above problem can be somehow avoided by producing a fuzzy version of the matched filter response Fig. (2B-8). As an example a fuzzy version of a ring with equation

$$f(x,y) = C \qquad \text{for} \qquad x^2 + y^2 = R^2$$

will be,

$$f(x,y) = C \qquad \text{for} \qquad (R-\epsilon)^2 < x^2 + y^2 < (R+\epsilon)^2$$

where ϵ is a small number.

The above technique can help to detect a rotated version of a target, even if it does not completely overlap our filter.

Throughout this section, the object to be detected $f(x,y)$ was always assumed deterministic. If the state of $f(x,y)$ is only known statistically, the matched filtering concept can be extended to the detection of a stochastic image field in the presence of noise. See [Interim Rep # 6] and (8).

A two-dimensional matched filtering process can be described in an all

digital system. Because the computer has a tremendous signal-to-noise advantage over physical systems and because there is no need to implement the filtering process on energy sensitive films, the matched filter can simply be constructed with two-dimensional Fourier transformation and conjugate operation of the signal to be detected. The actual filtering operation, then is obtained by Fourier transforming the input plane, multiplying by the filter plane, and then inverse transforming the product.

The above technique is not the only method for building a two-dimensional matched filter. In fact, the idea of building the matched filter optically is more exciting than using a digital computer . For optical data processing systems, implementation of a true matched filter can be extremely difficult. This is because the filter contains both modulus and phase information which somehow must be placed on some form of light sensitive material. For more on this subject see [Interim Rep # 6] and [9 - 10].

EXTRAPOLATION TECHNIQUES

Suppose a part of a photograph has been destroyed by some unwanted phenomenon (e.g.noise). Is there any way to reconstruct the damaged part and extract the features hidden in that portion ? This question may be answered by extrapolation of bandlimited signals discussed in this section.

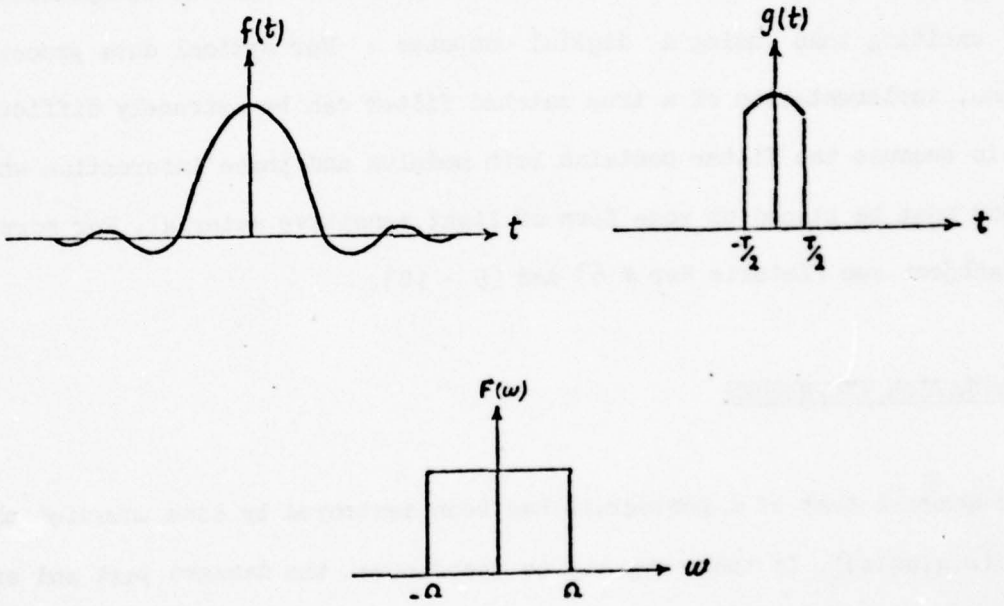
Very often it is desirable to extrapolate the unknown part of a signal f(t) in terms of a known finite segment g(t) where,

$$g(t) = \begin{cases} f(t) & |t| \leq T/2 \\ 0 & |t| > T/2 \end{cases} \qquad (2A\text{-}52)$$

In general this problem is not solvable, but if the function f(t) is constrained to be bandlimited, there are neat solutions to the problem.

Let $F(\omega)$ be the Fourier transform of f(t). If $F(\omega)$ vanishes for $|\omega| > \Omega$ then f(t) is said to be <u>bandlimited</u> and Ω is called the <u>cutoff angular frequency.</u>

An example of f(t) and g(t) is shown in Fig (2A-6)



Fig(2A-6) An example of a bandlimited signal.

As we mentioned earlier, in the case of bandlimited f(t), there are different ways to extrapolate the unknown part of f(t) in terms of g(t). One could, for example, calculate successive derivitives of f(t) at some point in the interval (- T/2 , T/2) and form a Taylor series representation which would cover everywhere. But since in practice, such a Taylor series has to be truncated somewhere along the line, the resultant approximation would give a very poor approximation of f(t) specially for large values of t, and moreover it will not be bandlimited.

Another approach is to try to find a complete set of bandlimited functions $\varphi_i(t)$ which are orthogonal on the interval (-T/2, T/2) as well as on the entire real line. Using these functions, one can expand g(t) in terms of $\varphi_i(t)$ in the interval (- T/2 , T/2), and then for any other t outside this interval, the expression gives the value of f(t) at that point.

Fortunately such functions have been found and are called Prolate Spheroidal functions. [Appendix II] or [11 - 12]

In terms of the above functions, f(t) can be written as,

$$f(t) = \sum_{0}^{\infty} a_n \varphi_n (t) \qquad (2A-53)$$

and the coefficients a_n can be expressed in terms of g(t) and the above functions,

$$g(t) = \sum_{0}^{\infty} a_n \varphi_n (t) \qquad (2A-54)$$

or

$$a_n = \frac{1}{\lambda_n} \int_{-T/2}^{+T/2} g(t)\, \varphi_n(t)\, dt \qquad (2A\text{-}55)$$

It follows that F(u), the Fourier transform of f(t) is given by [13]

$$F(u) = K \sum_{0}^{\infty} a_n \sqrt{\lambda_n}\, \varphi_n(ku) \qquad (2A\text{-}56)$$

in which

$$K = \sqrt{\frac{\Pi T}{\Omega}} \quad \text{and} \quad k = \frac{T}{2\Omega}$$

The idea of analytic continuation of a signal can be also used in image processing [14].

An image is a two-dimensional signal of limited extent,

$$f(x,y) = 0 \qquad \text{for} \qquad |x| > A \quad \text{or} \quad |y| > B \qquad (2A\text{-}57)$$

Let F(u,v) be the Fourier transform of f(x,y),

$$F(u,v) = \int_{-A}^{+A} \int_{-B}^{+B} f(x,y)\, \exp\left[-j2\Pi(ux + vy)\right] dx\, dy \qquad (2A\text{-}58)$$

If F(u,v) is only known over some passband

$$|u| \leq \frac{U}{2} \qquad (2A\text{-}59)$$

$$|v| \leq \frac{V}{2} \qquad (2A\text{-}60)$$

$$F(u,v) = \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} a_{ij}\, \varphi_{1i}(u)\, \varphi_{2j}(v) \qquad \forall\, u,v \tag{2A-61}$$

It is easy to show that the coefficients a_{ij} are given by,

$$a_{ij} = \frac{1}{\lambda_{1i}\, \lambda_{2j}} \int_{-\frac{U}{2}-\frac{V}{2}}^{+\frac{U}{2}+\frac{V}{2}} \varphi_{1i}(u)\, \varphi_{2j}(v)\, F(u,v)\, du\, dv \tag{2A-62}$$

the implicit dependence of $\varphi_1(u)$ on $\frac{UA}{2}$ and of $\varphi_2(v)$ on $\frac{VB}{2}$ has not been shown.

For many years, the classical Rayleigh diffraction limit of resolution was considered as a theoretical limit beyond which no higher spatial frequencies could be resolved (in optical system). But as we see from Eq.(2A-61), with some a priori knowledge (finite limit in space of the original object) it is possible to go beyond the diffraction limit.

While the technique of analytic continuation by the above method is valid theoretically, there are some limitations for practical purposes. First, in Eq (2A-53), the series should be truncated somewhere, and so the truncation error exists. However since the eigenvalues are in a decreasing order as a function of index i, [Appendix II] the mean square error is minimum in a Karhunen-Loéve expansion sense [Interim Rep # 3]. Secondly, since practically any image is accompanied by noise, F(u,v) in Eq. (2A-62) will not be the pure Fourier transform of f(x,y) and so the coefficients a_{ij} in the expansion will be degraded by noise. A very large image signal-to-noise ratio is needed to permit continuation to just a few spatial frequencies beyond the diffraction limit. The computational difficulty of Prolate Spheroidal functions is another problem in

the utilization of the discussed method.

Another method for extrapolation of bandlimited signals, known in a finite interval, has been proposed by Papoulis [13]. This method builds up the unknown part of the signal by successive iterations.

As before, let $g(t)$ be the known part of the signal $f(t)$ in the interval $(-T, T)$. $f(t)$ is bandlimited and we have,

$$F(\omega) = 0 \qquad \text{for} \quad |\omega| > \Omega \tag{2A-63}$$

The Fourier transform of $g(t)$ is called $G(\omega)$,

$$G(\omega) = \int_{-T}^{+T} g(t) \exp[-j\omega t]\, dt \tag{2A-64}$$

$G_0(\omega)$ is put equal to $G(\omega)$ and is put to zero for $|\omega| > \Omega$, the new function is called $F_1(\omega)$. The inverse Fourier transfor of $F_1(\omega)$ is taken and $f(t)$ is formed. For $|t| < T$, $f_1(t)$ is replaced by $g(t)$ and the new function is called $g_1(t)$ and so on. The n th. iteration step comprises the following

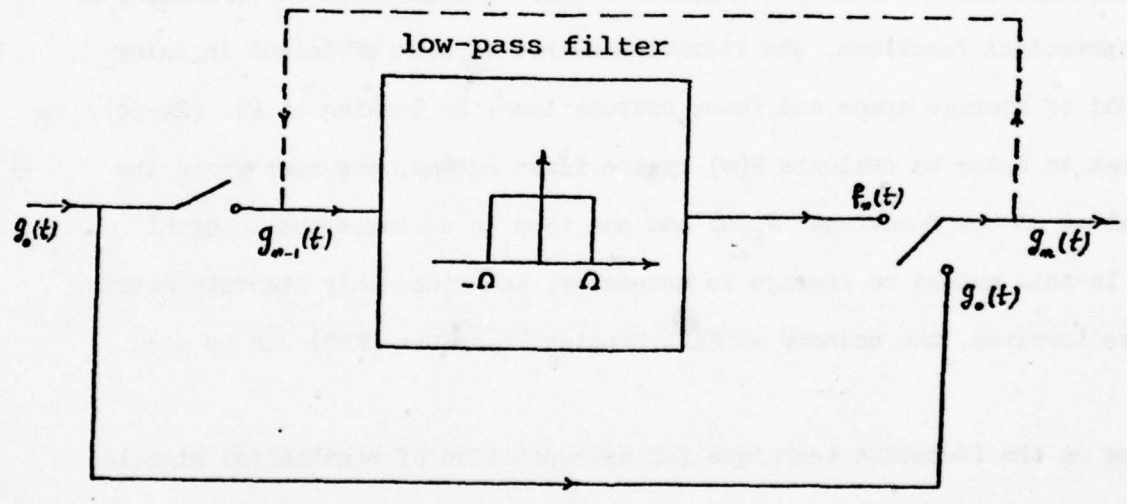
1 :
$$F_n(\omega) = \begin{cases} G_{n-1}(\omega) & |\omega| \leq \Omega \\ 0 & |\omega| > \Omega \end{cases} \tag{2A-65}$$

2 :
$$f_n(t) = \int_{-\Omega}^{+\Omega} F_n(\omega) \exp[j\omega t]\, d\omega \tag{2A-66}$$

3 :
$$g_n(t) = \begin{cases} g(t) & |t| \leqslant T \\ f_n(t) & |t| < T \end{cases} \qquad \text{(2A-67)}$$

4 :
$$G_n(\omega) = \int_{-\infty}^{+\infty} g_n(t) \exp[-j\omega t]\ dt \qquad \text{(2A-68)}$$

Fig. (2A-7) shows a realization of the above method using an ideal low-pass filter.



Fig(2A-7) The iterative method of signal extrapolation.

If we show the error at the n th iteration step by,

$$e_n(t) = f(t) - f_n(t) \tag{2A-69}$$

it can be shown that as $n \longrightarrow \infty$

$$e_n(t) \longrightarrow 0 \tag{2A-70}$$

$$f_n(t) \longrightarrow f(t) \tag{2A-71}$$

The number of necessary iterations for finding a good fit, depends on T and increases as ΩT decreases. This method as any other method is subject to some errors which are discussed in the original paper. Compared with the technique of Prolate Spheroidal functions, the iteration method is more efficient in using less amount of storage space and fewer computations. By looking at Eq. (2A-56), we see that in order to evaluate $F(\omega)$ by the first method, one must store the sample values of the functions $\varphi_k(u)$ and use them to evaluate the integral (2A-55). In this method no storage is necessary, and since only discrete Fourier series are involved, the economy of Fast Fourier Transform (FFT) can be used.

Based on the iteration technique for extrapolation of bandlimited signals, another technique has been proposed in a paper by Sabri and Steenaart [15]. In this technique, the total extrapolation process is achieved by a single matrix operation. The matrix is called the Extrapolation Matrix. The proposed technique and its implementation has some advantages over the other extrapolation techniques in terms of the number of computations and accuracy of results. Moreover it can be implemented on a real-time basis.

The idea of extrapolation of bandlimited signals in terms of a known segment, has been mainly used in image processing for super resolution. However since this paper is devoted to cartography and the associated problem of feature extraction, it will be nice to see how the extrapolation technique can be used to enhance hidden details of an image. Suppose we have a picture which is bandlimited (in practice any picture can be considered bandlimited). A part of this picture has been degraded by noise, and unfortunately this is the part in which we are most interested. We are looking for a scheme which can bring out and enhance the features hidden in that part. One plausible approach is to use the above discussed extrapolation technique. Since the picture is bandlimited, we can pick a non-degraded extent-limited portion of the picture and extrapolate the whole picture in terms of that section. In this way we find a description for the part of the picture which was hidden in noise.

Usually the noise is not limited to one part of a picture, but is present everywhere and will be spread by extrapolation. To avoid its effect, one can use the randomness nature of noise and perform the above technique many times using different portions of the picture as a starting point. The results are averaged to give a good approximation of the part of the picture to be enhanced.

2B - EXAMPLES

In section 2A, the analytical properties of Fourier transforms, Hankel transforms and the matched filter were investigated. To see how these transformations act on different patterns, some candidate features from the objects usually present in photographs will be chosen and we will try to find the corresponding transformed function under the application of different transforms. On this basis we will be able to compare different transforms with respect to their capability of bringing out specific features. For Fourier and Hankel transforms we can usually get a closed analytical form for the transformed function and the analysis will be based on this closed form. For matched filters, we like to see how the position or/and value of the output peak is affected by the input.

Aerial photographs usually contain a variety of object images. Some of these objects have some known geometrical shape (or at least can be approximated in that way), and can be expressed by some analytical functions. However, there are many others which do not possess such a property. Here we select some simple but common cartographical features and try to find how they are affected by the picture transforms.

The following patterns seem satisfactory :

(a) Slanted lines

(b) Shaded edges

(c) Centered circles

(d) Cluster of circles with circular symmetry

Slanted lines are very commonly seen on an aerial image. They can have

different extents and may appear parallel to each other. Images of roads, streets, highway, bridges, etc., appear as slanted lines on a photograph.

Any object which appears as a line on a picture taken from a far distance, may appear as a strip on a picture of the same scene taken from a closer distance. By a shaded edge we mean a strip whose brightness gradually changes along its width.

Circles on a photograph generally suggest the presence of storage containers in the scene. If there are more than one storage container on a spot, they usually show circular symmetry on the picture. This suggests to see how a particular transform affects the patterns of several circles with circular symmetry.

Fig (2B-1) shows the candidate features named above.

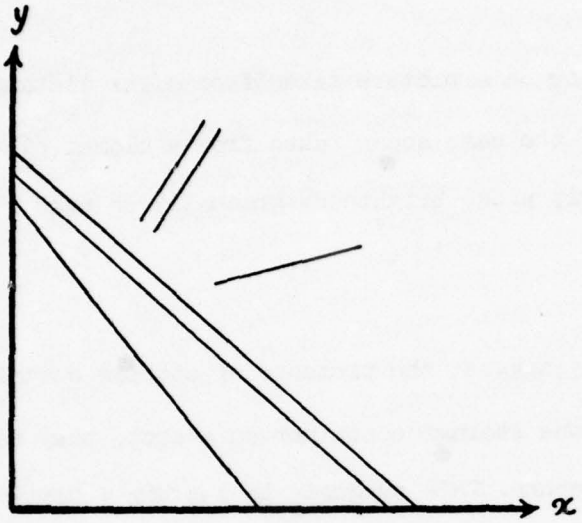
EXAMPLES FOR FOURIER & HANKEL TRANSFORMS

The visual examination of a transform is usually limited to a display of its magnitude. In the Fourier transform this magnitude is called the Fourier spectrum. The spectrum possess translational invariance and moreover is rotated by an angle θ_o if the original image is rotated by the same angle (see section 2A)

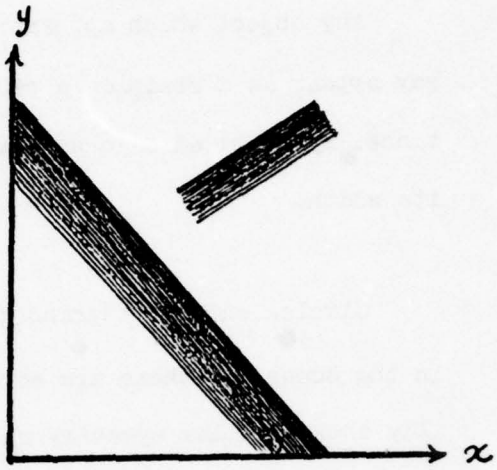
In accordance with the above remarks, we take the candidate features in their simplest form. It will be easy to visualize the transformation of their rotated or translated versions.

In Fig (2B-2) we have shown the Fourier transform of some candidate functions.

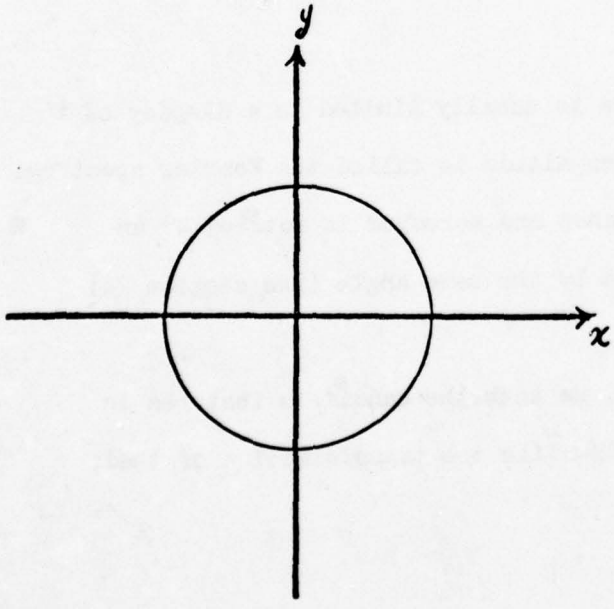
(a) Slanted Lines



(b) Shaded Edges



(c) Centered Circle



(d) Cluster of Circles with Circular Symmetry

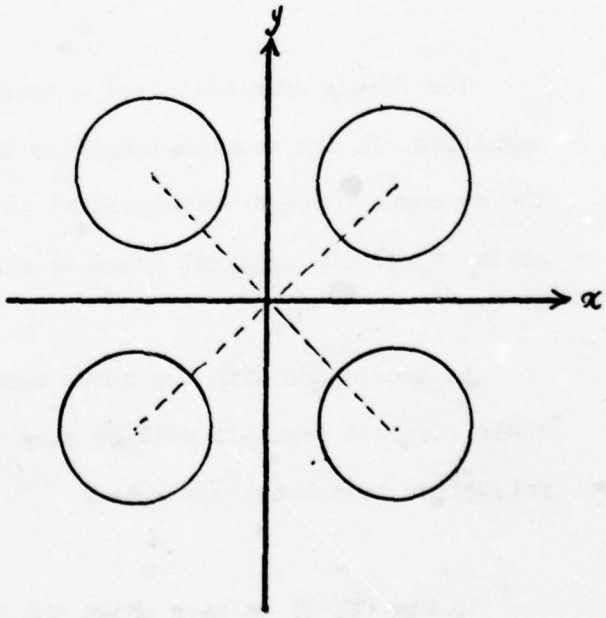
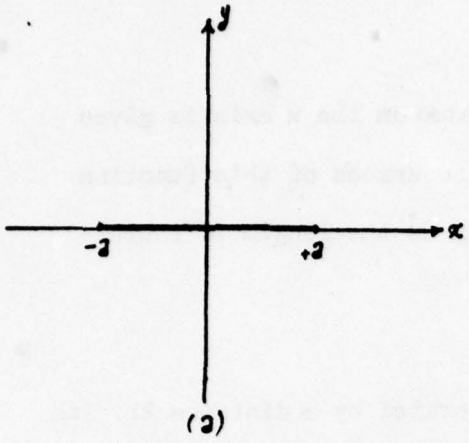


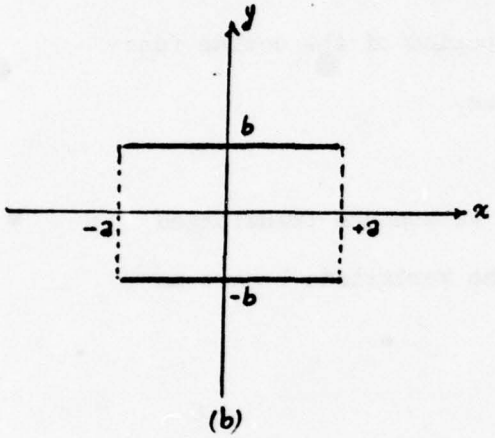
FIG.(2B-1) Candidate features for transforms study



(a)

$$f(x,y) = P_a(x)\,\delta(y) \qquad P_a(x) = \begin{cases} 1 & a > x > -a \\ 0 & \text{otherwise} \end{cases}$$

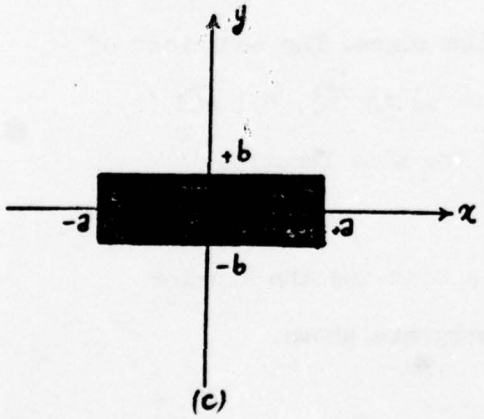
$$F(u,v) = \frac{2 \sin 2\pi a u}{2\pi u}$$



(b)

$$f(x,y) = P_a(x)\left[\delta(y-b) + \delta(y+b)\right]$$

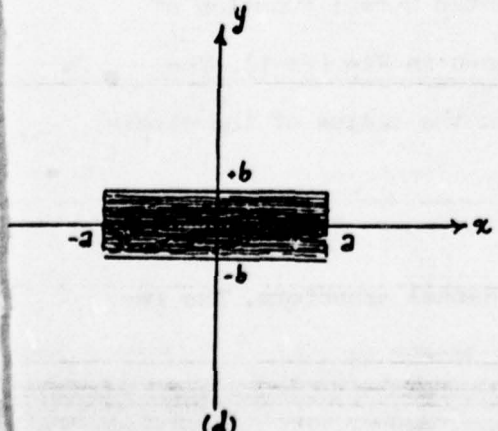
$$F(u,v) = \frac{4 \sin 2\pi a u}{2\pi u} \cos bv$$



(c)

$$f(x,y) = P_a(x)\,P_b(y)$$

$$F(u,v) = 4\,\frac{\sin 2\pi a u}{2\pi u} \cdot \frac{\sin 2\pi b v}{2\pi v}$$



(d)

$$f(x,y) = P_a(x)\left[1 - \frac{|y|}{b}\right]$$

$$F(u,v) = \frac{\sin^2\left(v\frac{2\pi b}{2}\right)}{b\pi^2 v^2} \cdot \frac{\sin 2\pi a u}{\pi u}$$

FIG. (2B-2) Candidate Functions and their Fourier Transform

(a) The Fourier transform of a line segment located on the x axis is given by a sinc function along the u axis. The period of the zeroes of this function depends on the length of the segment and gets smaller as the length is increased. No variation exists along the v axis.

(b) Here we have two parallel line segments seperated by a distance 2b. Its Fourier transform is a function as in part (a) multiplied by a cosine function that varies in a direction normal to the lines. The period of the cosine function is proportional to the distance between the lines.

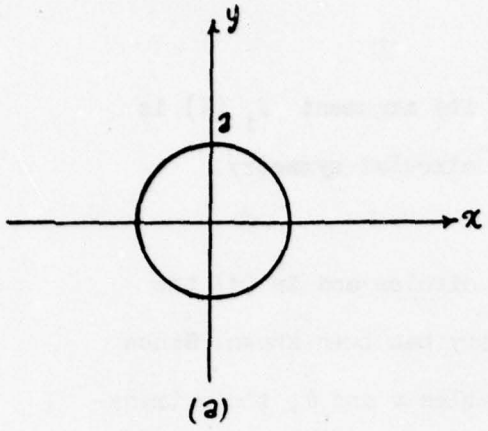
(c) A strip of light has been considered and as we see the transformed function has variations in both u and v direction. The variations behave as sinc functions.

(d) In this pattern the brightness intensity is maximum at the center of the strip and is gradually decreased as we approach the edges. The behaviour of the transformed function in the u direction is the same as in (a),(b), and (c). Along the v axis the function varies as the square of the sinc function.

In Fig (2B-3) the Hankel transform of a circle, a disk and the Fourier transform of a cluster of circles with circular symmetry are shown.

(a) The Hankel transform of a circle is given by the Bessel function of zero order. The behaviour of the function has been shown in Fig (2A-1). The argument of the Bessel function is (aw) and depends on the radius of the circle. The transform is circularly symmetric.

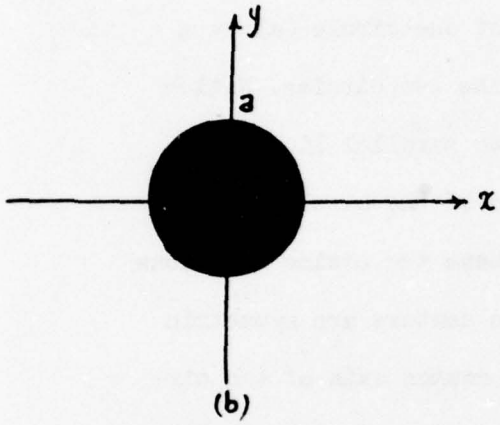
(b) A disk of radius a has been acted on by the Hankel transform. The re-



(a)

$$f(r) = \delta(r-a)$$

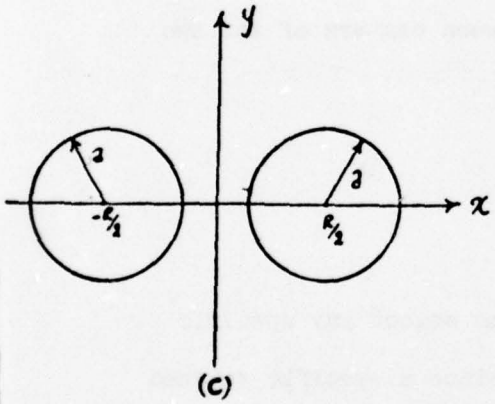
$$F(\omega) = a J_0(a\omega)$$



(b)

$$f(r) = \begin{cases} 1 & a > r > 0 \\ 0 & \text{otherwise} \end{cases}$$

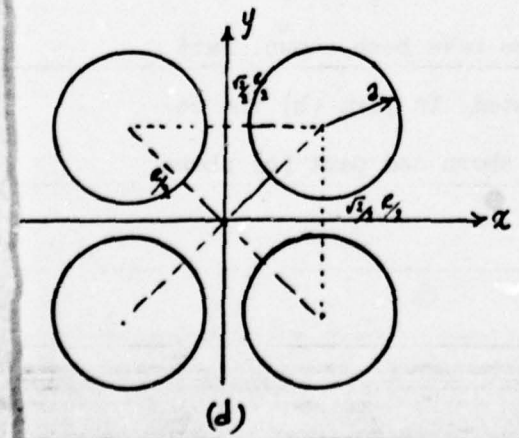
$$F(\omega) = \frac{a J_1(a\omega)}{\omega}$$



(c)

$$f(x,y) = \delta\left[(x+\tfrac{R}{2})^2 + y^2 = \rho^2\right] + \delta\left[(x-\tfrac{R}{2})^2 + y^2 = a^2\right]$$

$$F(u,v) = 4\pi a J_0(a\sqrt{u^2+v^2}) \cos uR/2$$



(d)

$$F(u,v) = 8\pi a J_0(a\sqrt{u^2+v^2}) \cos \frac{Ru}{2}\frac{\sqrt{2}}{2} \cos \frac{Rv}{2}\frac{\sqrt{2}}{2}$$

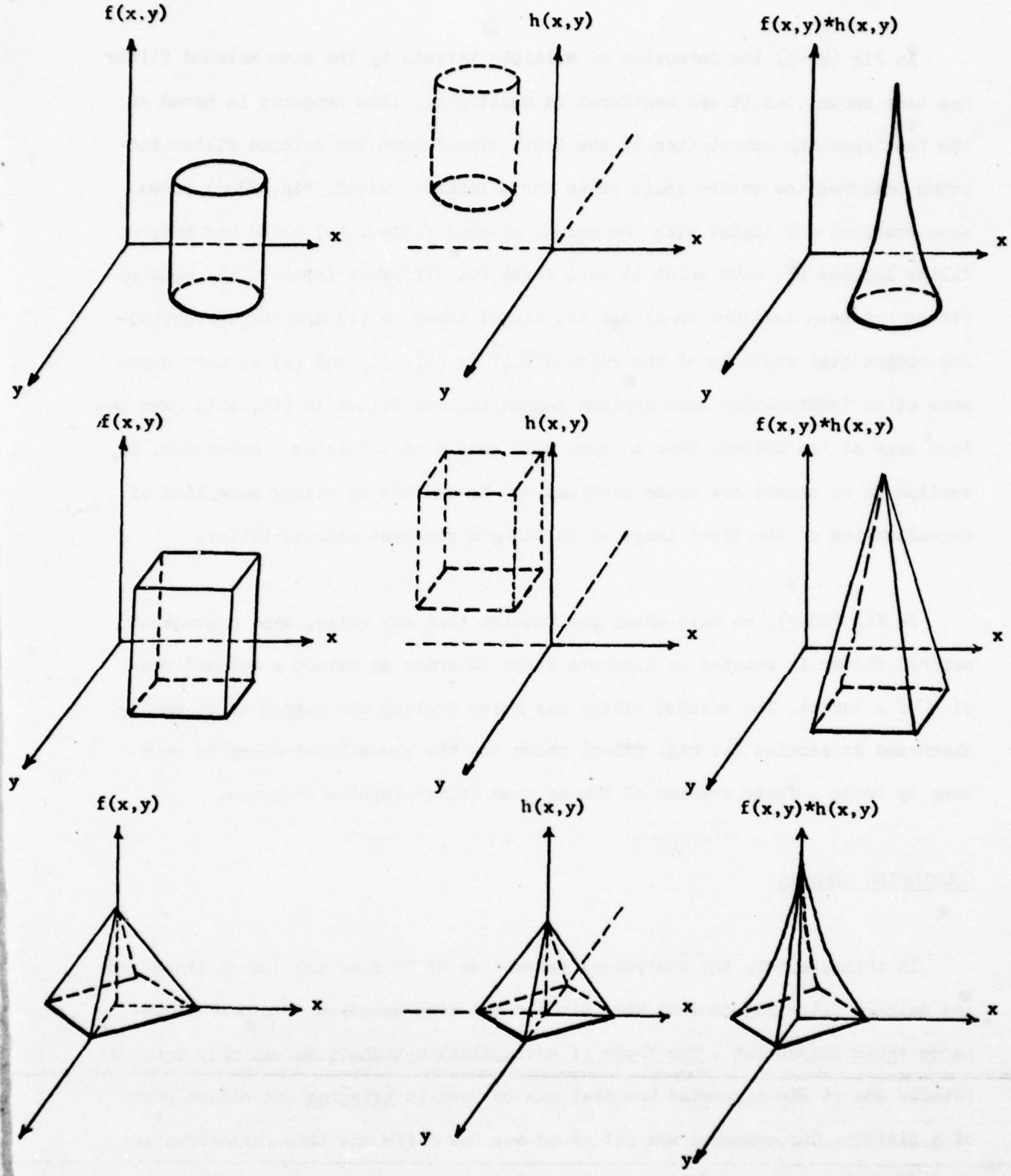
FIG (2B-3) Candidate functions and their Hankel/Fourier Transform

sult is the Bessel function of first order divided by its argument $J_1(X)$ is shown in Fig (2A-1). The transformed function possess circular symmetry.

(c) and (d): In (c) the Fourier transform of two circles and in (d) the Fourier transform of four circles with circular symmetry has been shown. Since the function in (c) and (d) are functions of two variables r and θ, their transforms won't be circularly symmetric. The Fourier transform of the two circles in (c) is the multiplication of the Fourier transform of one circle (a) by a cosine function which varies along the common axis of the two circles. Notice that this behaviour is analagous to the behaviour of two parallel lines in Fig (2B-2). In (d) we see that the transformed function of the circle in (a) has been multiplied by two cosine functions. Each of these two cosine functions varies along the common axis of those two circles whose centers are symmetric relative to the origin. It is easy to show that if the common axis of two circles makes an angle θ with the x axis, then the argument of the cosine function will appear as $\frac{Ru}{2}\cos\theta$ in which R is the distance between centers of the two circles.

EXAMPLES FOR MATCHED FILTERING

In matched filtering method, it is not necessary to select any specific features in order to show the power of the technique. Since a specific matched filter can be built to detect any object image in a picture, any example will be as useful as any other. In Fig (2B-4), some examples have been shown. Part (a) of the figure shows some object images to be detected. In part (b) the corresponding matched filter transfer function has been shown and part (c) shows the corresponding outputs.



(a) matched filter input

(b) matched filter transfer function

(c) matched filter output

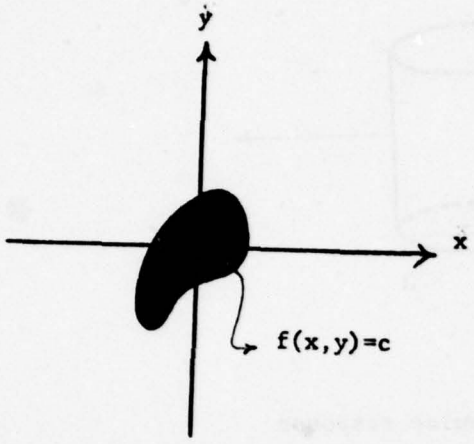
FIG(2B-4)Some examples illustrating matched filtering operation

In Fig (2B-5) the detection of multiple targets by the same matched filter has been shown . As it was mentioned in section 2A, this property is based on the fact that the convolution of the input signal with the matched filter response searches the entire image plane for a possible match. Fig.(2B-6) shows some problems associated with the normal matched filter. (a) shows our matched filter impulse response which is kept fixed for different inputs. This matched filter has been designed to detect the target shown in (b) and the corresponding output peak would be of the value $C^2 \Pi A^2$. In (c), (d) and (e) we have shown some other inputs which when applied to the matched filter in (a), will give the same peak at the output. This in turn will result in a " false " detection. In section 2A we showed how these problems can be avoided by either some kind of normalization of the input image or by using a gradient matched filter.

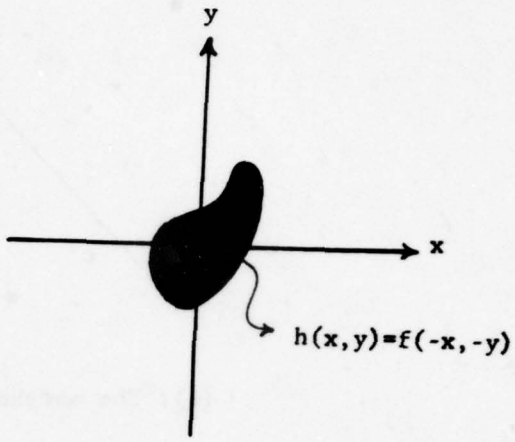
In Fig (2B-7), we have shown the problem that may arise, when a gradient matched filter is rotated in discrete steps in order to detect a rotated version of a target. The matched filter may never overlap the target as it was discussed in section 2A. Fig. (2B-8) shows how the above problem can be overcome by using a fuzzy version of the matched filter inpulse response.

CONCLUDING REMARKS

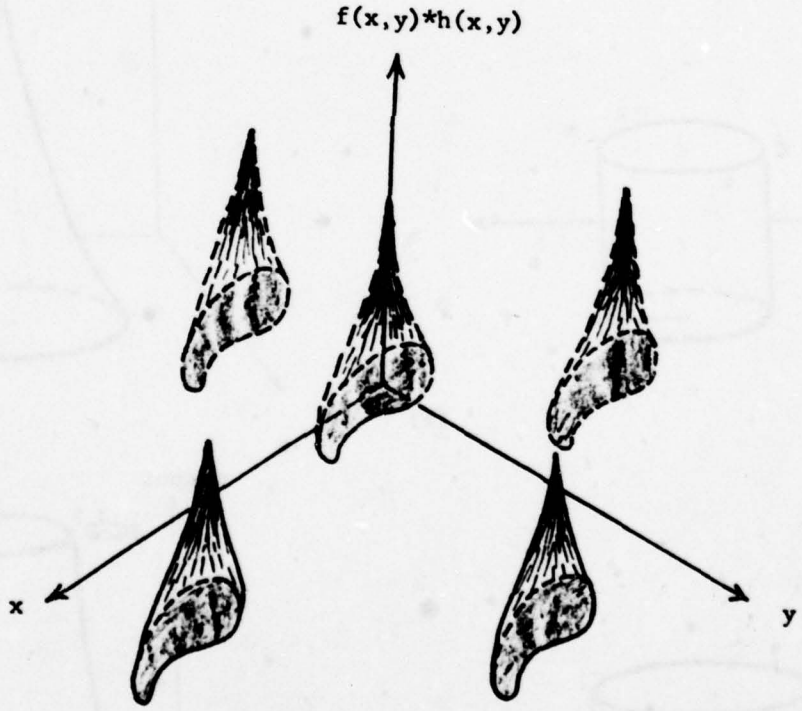
In this chapter, the analytical properties of Fourier and Hankel transforms and matched filtering process were studied and some examples were used to enhance those properties . The topic of extrapolation techniques was only touched briefly and it was suggested how that can be used in bringing out hidden parts of a picture. Our emphasis was put on to see how different transformations act on different patterns. Moreover, an important aspect of any transformation being used in feature extraction, is how that transform behaves when the input



(a) matched filter input

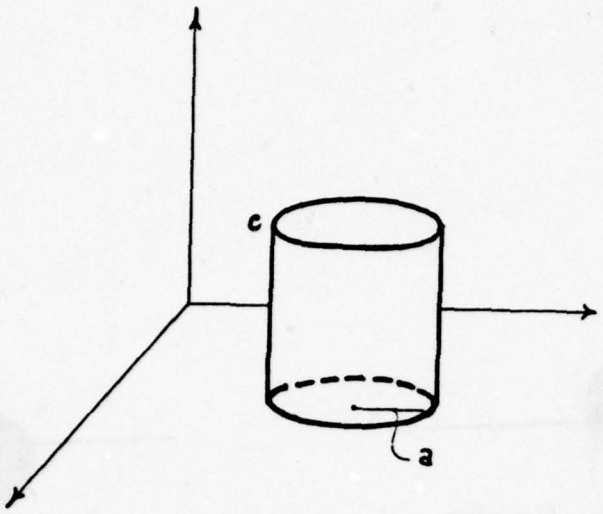


(b) matched filter transfer function

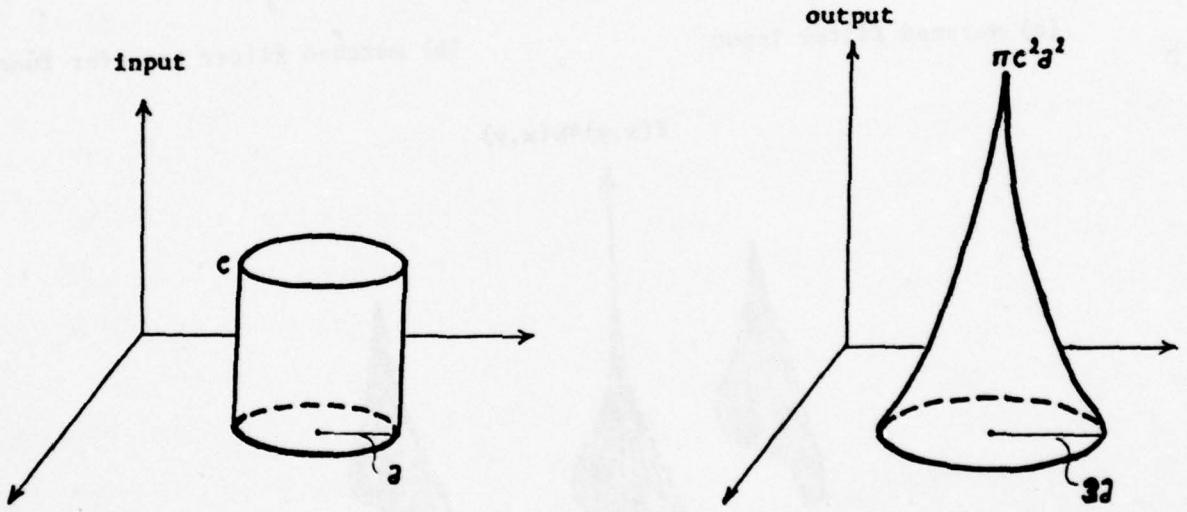


(c) matched filter output

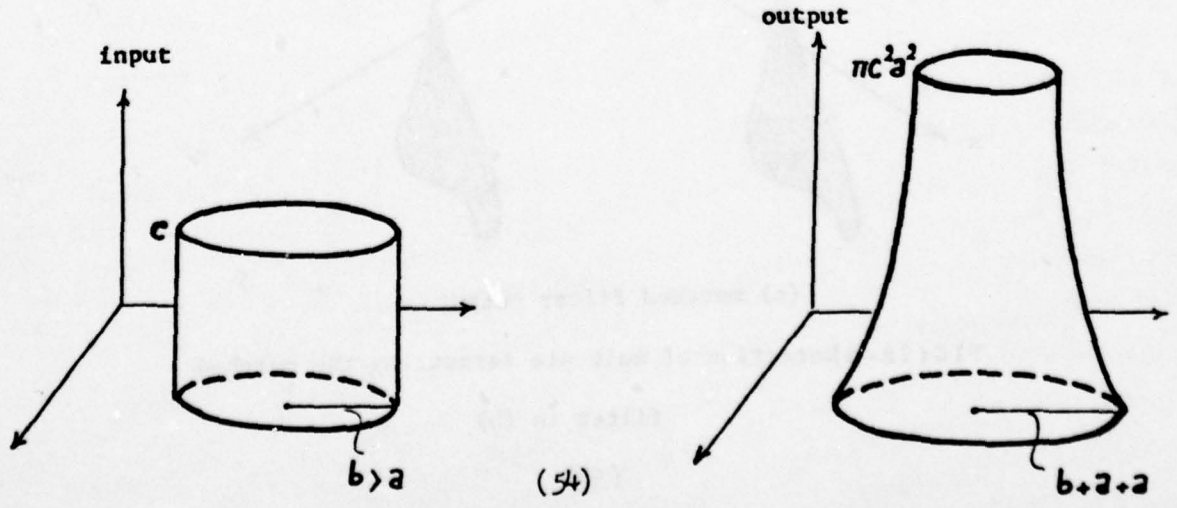
FIG(2B-5)Detection of multiple targets by the matched filter in (b)



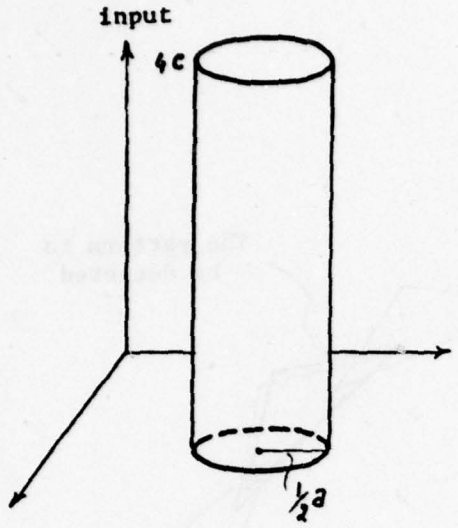
(a): The matched filter impulse response

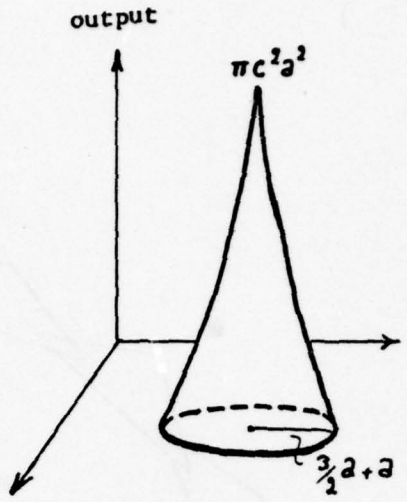


(b)

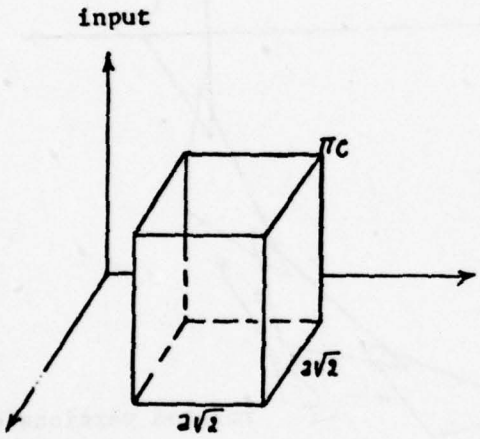


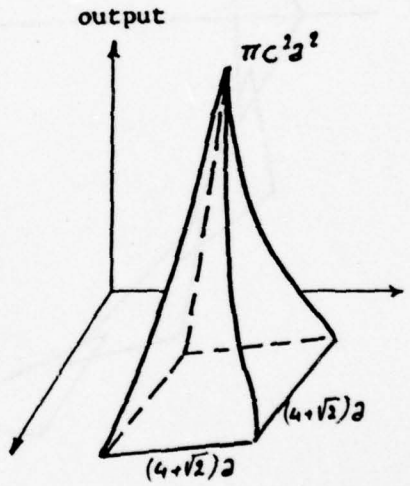
(c)



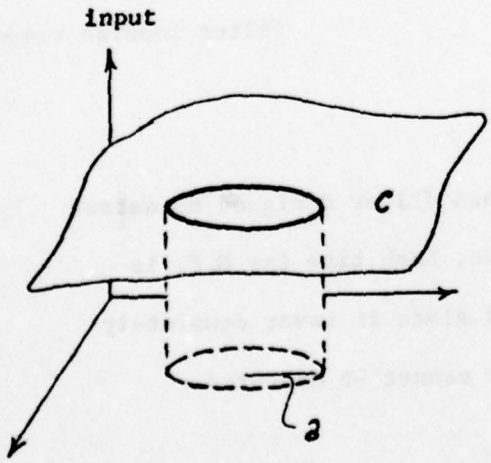


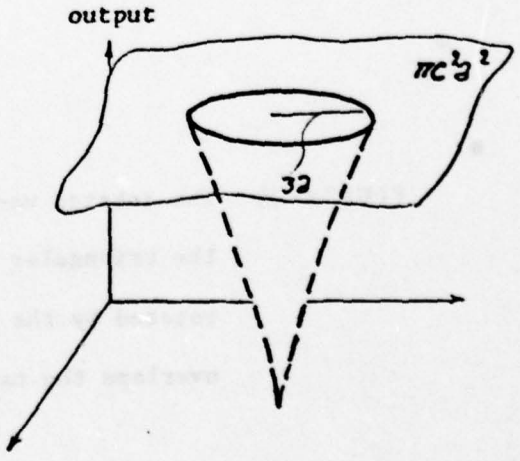
(d)





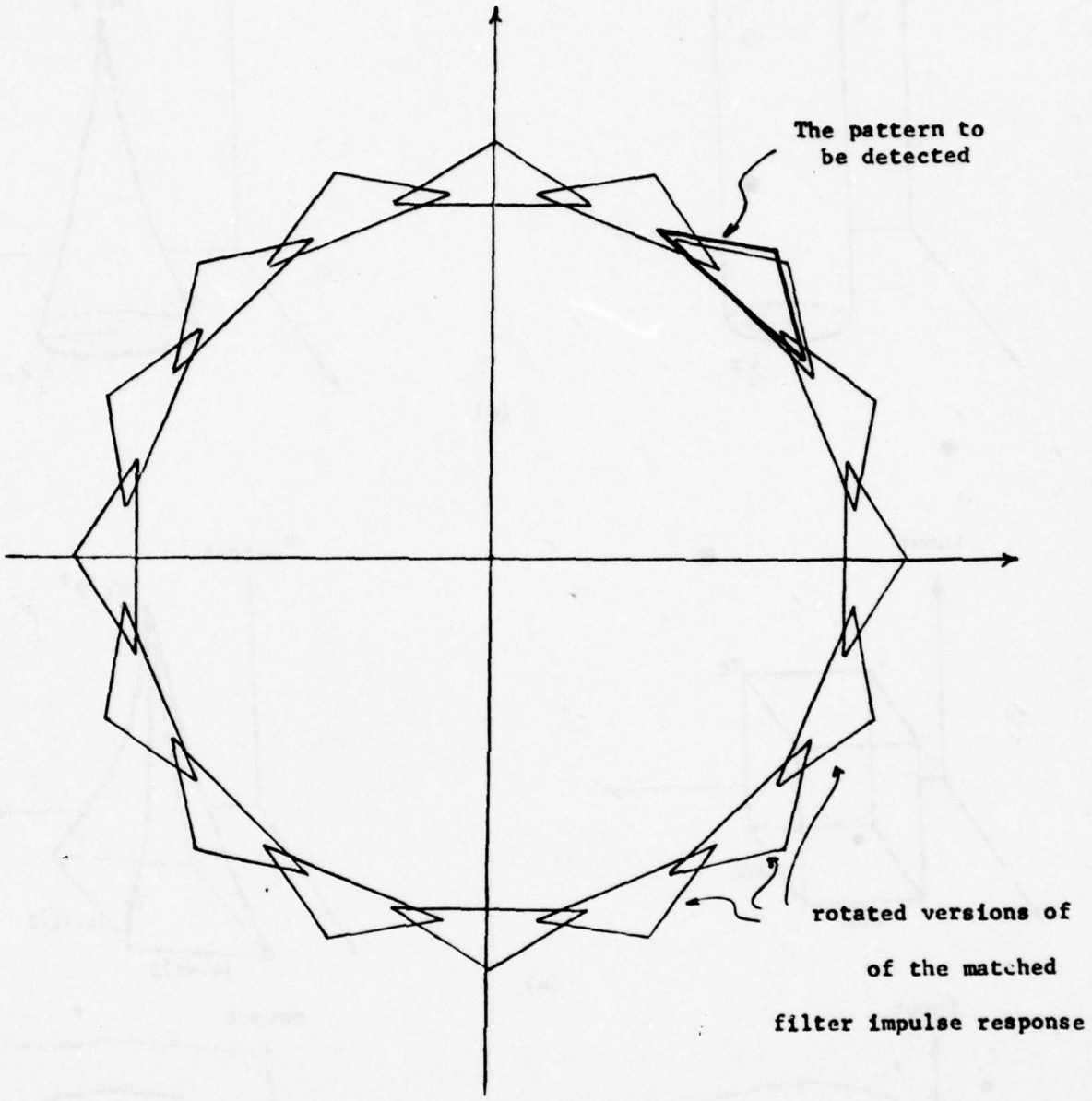
(e)



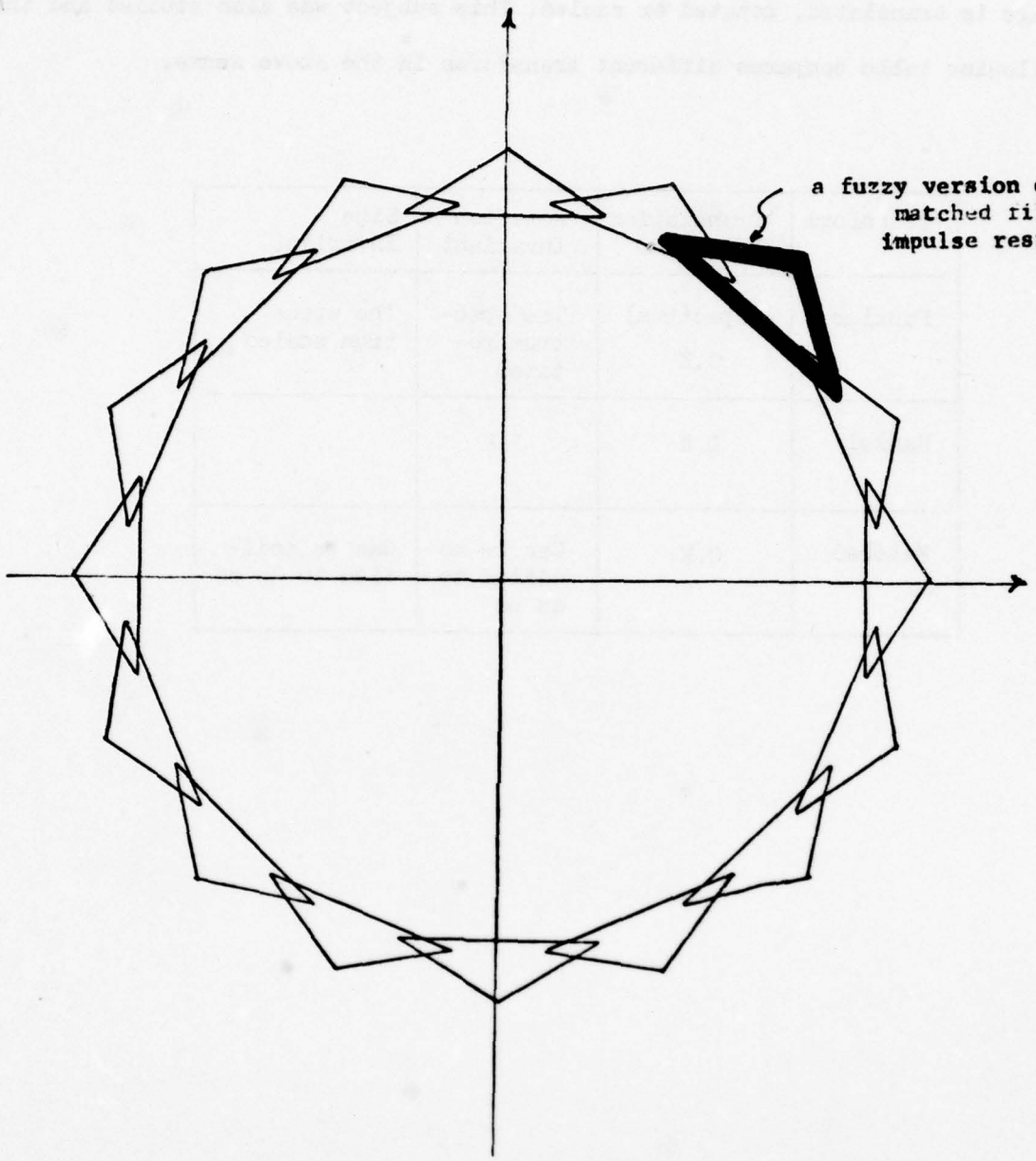


(f)

FIG(2B-6) The response of the Matched Filter in (a) to different input patterns



FIG(2B-7) The rotated versions of a matched filter designed to detect the triangular image shown above. Each time the M.F. is rotated by the amount 22.5, and since it never completely overlaps the target, the object cannot be detected.



FIG(2B-8) **The drawback of the matched filter in** FIG(2B-7) **has been overcome by using a fuzzy version of it . Here one of the rotated versions overlap the object image to be detected.**

image is translated, rotated or scaled. This subject was also studied and the following table compares different transforms in the above sense.

Transform	Translation Invariant	Rotation Invariant	Size Invariant
Fourier	(Spectrum) O.K	The spectrum rotates	The spectrum scaled
Hankel	O.K	O.K	,,
Matched	O.K	Can be modified to do so	Can be modified to do so

REFERENCES OF CHAPTER TWO

(1) Goodman, J.W., Introduction to Fourier Optics, Mc Graw Hill, New York, 1968.

(2) Papoulis, A , Systems and Transforms with Applications in Optics, Mc Graw Hill, New York, 1968.

(3) Gonzalez, Rafael, Digital Image Processing, Addison - Wesley, Massachusettes, 1977.

(4) Turin, G.L., " An Introduction to Matched Filters ", IRE Trans. Inf. Theory, June 1960.

(5) Wozencraft, John M. and Jacob, I.M., Principles of Communication Engineering , John Wiley and Sons Inc., New York, 1965.

(6) Papoulis, A, Probability, Random Variables and Stochastic Processes, Mc Graw Hill, New York, 1965.

(7) Van Trees, Harry L., Detection, Estimation, and Modulation Theory , John Wiley & Sons, Inc., New York, 1968.

(8) Arcese, A. et . al, " Image Detection through Bipolar Correlation, " IEEE Trans. Inf. Theory, IT - 16, 5, Sept. 1970.

(9)V Vander Lugt, A., " Signal Detection by Complex Spatial Filtering ", IEEE Trans, Inf. Theory, IT - 10, No.2, April 1964.

(10) Lohmann, A.W., " Signal Detection by Correlation of Fresnel Diffraction Patterns ", Appl. Opt. 6, No. 12, Dec. 1967.

(11) Slepian, D. and Pollak, H.O., " Prolate Spheroidal Wave Functions, Fourier Analysis and Uncertainty - I " , Bell Syst. Tech.J., 40, Jan 1961

(12) Landau, H.J. and Pollak, H.O., " Prolate Spheroidal Wave Functions, Fourier Analysis and Uncertainty - II ', Bell Syst. Tech. J., Jan 1961.

(13) Papoulis, A. " A New Algorithm in Spectral Analysis and Band-Limited Extrapolation ", IEEE Trans. Circuits and Systems, Vol. CAS -22, No. 9, Sept. 1975.

(14) Pratt, William K., Digital Image Processing , Wiley-Interscience, New York, 1978.

(15) Sabri, M.S. and Steenarrt, W., " An Approach to Band-Limited Sygnal Extrapolation : The Extrapolation Matrix ", IEEE Trans. Circuits and Systems, Vol. CAS - 25, No. 2, Feb. 1978.

CHAPTER III

DISCRETE TRANSFORMS

Cartographic objects have different characteristics and so can be categorized according to some specific properties. As an example, roads and rivers, on a cartographic picture look like a line. Some objects may look like circles, rectangular patches and some geometrical shapes. The discrete transforms when operated on any of those objects, sometimes give some particular patterns or salient features representative of the original objects of cartography in transform domain. Since the geometry and the object are usually represented in discrete numbers it is logical to attempt to resort to discrete transform techniques for feature extraction of these objects. Additionally, discrete transforms lend themselves readily to easy implementation. This chapter will concentrate on an exposition of discrete transforms and examples which may have some relevance to cartographic analysis.

To process a picture f(x,y), this picture is usually sampled on a grid of M X N points, each number in the array representing the gray level in the picture at the corresponding sampling point. Thus, a picture can be represented by a matrix. [f], the elements of which are f(x,y).

A general discrete transform on this picture is given by,

$$F(u,v) = \sum_{x=0}^{M} \sum_{y=0}^{N} f(x,y)\, g(x,y,u,v)$$

F(u,v) is the transform with u,v taking on integer values from M to N respectively and g(.) is the Kernel of the particular transform. The inverse transform if it exists will be,

$$f(x,y) = \sum_{u=0}^{M} \sum_{v=0}^{N} F(u,v)\ h(x,y,u,v)$$

where the $g(x,y,u,v)$ and $h(x,y,u,v)$ pair are called the forward and inverse transformation kernels. The forward kernel is said to be seperable if

$$g(x,y,u,v) = g_1(x,u)\ g_2(u,v)$$

and in addition is symmetric if g_1, and g_2 are functionally equal.

In this case,

$$g(x,y,u,v) = g_1(x,u)\ g_1(y,v)$$

The same argument holds for the inverse kernel. The seperable symmetric property is desirable when the implementation problem comes into consideration. Furthermore, since statistical intensity variations of most images are nearly the same in the vertical and horizontal directions, only seperable symmetric kernels usually need to be considered, of course, as we will see later, some transforms like Slant transform are not symmetric and they are usually used to process some special patterns.

It is often useful to express two-dimensional transforms in matrix form. For simplicity let us take M = N, so that we will have a square array. Then for a transform kernel that is seperable symmetric let,

$[f]$ = image matrix

$[F]$ = transformed image matrix

$[A]$ = transform matrix

then by matrix multiplication

$$[F] = [A][f][A]$$

It can be easily shown that the inverse transform is given by,

$$[f] = [A]^{-1}[F][A]^{-1}$$

Thus $f(x,y)$ and $F(u,v)$ can be expressed as a two-dimensional transform if $[A]$ has an inverse

If $[A]$ is a symmetric orthogonal matrix then,

$$[A]^{-1} = [A]$$

If a transform kernel is seperable but not symmetric, then we have,

$$[F] = [A][f][A]^{T}$$

On the other hand if it is to be a one dimensional transform then $[F] = [A][f]$ and the inverse transform is $[f] = [A]^{T}[F]$ if $[A]$ is not symmetric or else $[f] = [A][F]$ if $[A]$ is symmetric.

One can easily indentify $1/N\ W_n$, $1/N\ H_n$, $1/N\ S_n$ and D_n as the Kernel A

of N X N dimension in the case of Walsh Hadamard, Haar, Slant and Discrete Cosine transform in the following pages.The image matrix [f] is represented as a vector f_N where

$f_N = [f_0 \quad f_1 \quad f_2 \ldots\ldots\ldots f_{N-1}]$ for a one dimensional data pixel system and as f_{MN} where

$$f_{MN} = \begin{bmatrix} f_{00} & f_{01} & \ldots\ldots\ldots\ldots & f_{0\,N-1} \\ f_{10} & f_{11} & \ldots\ldots\ldots\ldots & f_{1\,N-1} \\ & & & \\ f_{M-10} & f_{M-11} & \ldots\ldots\ldots\ldots & f_{M-1\,N-1} \end{bmatrix}$$

for a two dimensional data pixel system.

To make this chapter self contained, in section (A) of this chapter, a general review of the two-dimensional discrete transforms is discussed along with a brief summary of their properties. The Walsh Hadamard transforms (WHT), the Haar transforms (HT) the Slant transforms are covered in this chapter. The discrete transforms seem to have a decided advantage over the continuous transforms from the point of view of the Hardware/Software implementation with a Microprocessor based signal processing system . In section B of this chapter, the two-dimensional transforms of the candidate features of the aerial photographic patterns, are compared from the point of view of the characteristic properties of feature extraction.

3A: THEORY

TWO-DIMENSIONAL WHT (1):

Walsh-Hadamard transforms are used for the Walsh representation of data sequences. Their basis functions are sampled Walsh functions which can be expressed in terms of the Hadamard matrices W_n . These matrices can be generated using the following recurrence relation

$$W_k = \begin{bmatrix} W_{k-1} & W_{k-1} \\ W_{k-1} & -W_{k-1} \end{bmatrix} \qquad k = 1,2, \ldots n \tag{3A-1}$$

where $W_0 = 1$ and $n = \log_2 N$.

For example, with k = 1, and k = 2, Eq. (3A-1) yields

$$W_1 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \qquad W_2 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

The order of a Hadamard matrix need not be a power of two. Existence of the Hadamard transforms for values of N other than integer power of two has been shown up to N = 200. However, most of the applications of transforms in image processing are based on $N = 2^n$ samples per row or column of an image.

A frequency interpretation can be given to the Hadamard matrix generated by the core matrix of Eq. (3A-1). Along each row of the Hadamard matrix the frequency is called the number of changes in sign. The word " sequency " is primarily used to designate the number of sign changes. It is possible to construct a Hadamard matrix of $N = 2^n$ that has sequency components at every integer from 0 to N - 1. Fig. (3A-1) shows the Hadamard matrix of order 8 along with the sequency interpretation.

MATRIX								SEQUENCY
+	+	+	+	+	+	+	+	0
+	-	+	-	+	-	+	-	7
+	+	-	-	+	+	-	-	3
+	-	-	+	+	-	-	+	4
+	+	+	+	-	-	-	-	1
+	-	+	-	-	+	-	+	6
+	+	-	-	-	-	+	+	2
+	-	-	+	-	+	+	-	5

Fig. (3A-1) : Hadamard matrix of order 8

A very closely related transform to the Hadamard transform is the Walsh transform. For $N = 2^n$, the Walsh transformation matrix is found by interchanging the orders of the rows and columns of the corresponding Hadamard matrix. Since this is the only difference between Walsh and Hadamard matrices when $N = 2^n$, the use of the Walsh and Hadamard transforms is intermixed in the picture processing literature and hence the term Walsh-Hadamard transform is used to denote either of them.

It is straightforward to show that the matrices W_k of 2^k rows and 2^k columns have the following properties:

(a) W_k is a symmetric matrix, i.e.

$$W_k' = W_k \tag{3A-2}$$

prime denoting transpose.

(b) W_k is orthogonal, i.e.

$$W_k' \; X \; W_k = 2^k \, I_k \tag{3A-3}$$

where I_k is the $(2^k \; X \; 2^k)$ identity matrix.

(c) The inverse of W_k is proportional to itself, i.e.

$$W_k^{-1} = 2^{-k} \, W_k \tag{3A-4}$$

where W_k^{-1} is the inverse of W.

Let f_N denote an N-periodic Data sequence,

or $$f_N = [f(0) \; f(1) \; ..... \; f(N - 1) \tag{3A-5}$$

where $$N = 2^n \tag{3A-6}$$

The (WHT) of f_N is defined as

$$F_N = \frac{1}{N} W_n f_N \qquad (3A\text{-}7)$$

Define F(k) as the kth WHT coefficient and let $F_N = [F(0), F(1).... F(N - 1)]$.

From (3A-4) and (3A-7) it follows that the inverse (WHT) transform (IWHT) is defined as

$$f_N = W_n F_N \qquad (3A\text{-}8)$$

Since (3A-7) and (3A-8) constitute a transform pair, the (WHT) representation of f_n is unique. The two-dimensional (WHT) and its inverse can be equivalently defined in matrix form as follows: (the details are covered in the interim report 4).

For a data sequence f_{MN} of M rows N columns

$$F_{MN} = \frac{1}{MN} W_m f_{MN} W_n$$

and (3A-9)

$$f_{MN} = W_m F_{MN} W_N$$

where $M = 2^m$ and $N = 2^n$

$$\text{or } F_{MN} = \begin{bmatrix} F(0,0) & F(0,1) & F(0,N-1) \\ F(1,0) & F(1,1) & F(1,N-1) \\ \cdots & \cdots & \cdots \\ F(M-1,0) & F(M-1,1) & F(M-1,N-1) \end{bmatrix}$$

where $F(i,j)$ is the (i,j)th transform coefficient.

PROPERTIES OF WHTs

(1) The sequency spectrum of a function is a measure of the correlation of this function with the Walsh basic-patterns. These basic-patterns were used to analyze the Walsh spectra of one and two-dimensional figures and relations between image samples and Walsh spectrum coefficients were established [2] (see the Fig. 3A-2, 3A-3, 3A-4).

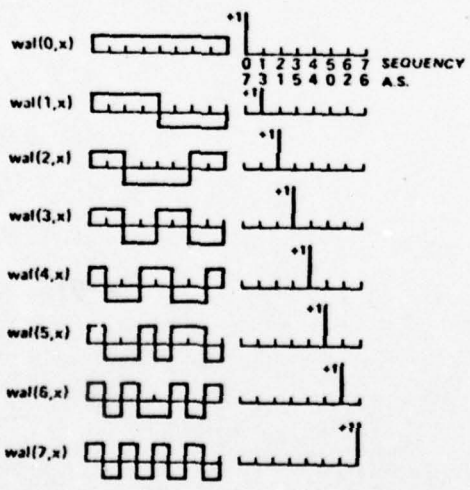


Fig. (3A-2) Sequency Spectra of Walsh Function

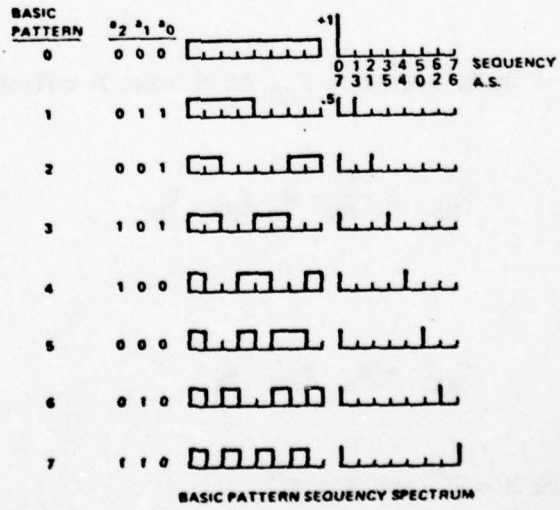


Fig. (3A-3) One-Dimensional " Basic Patterns " and their Sequency Spectra

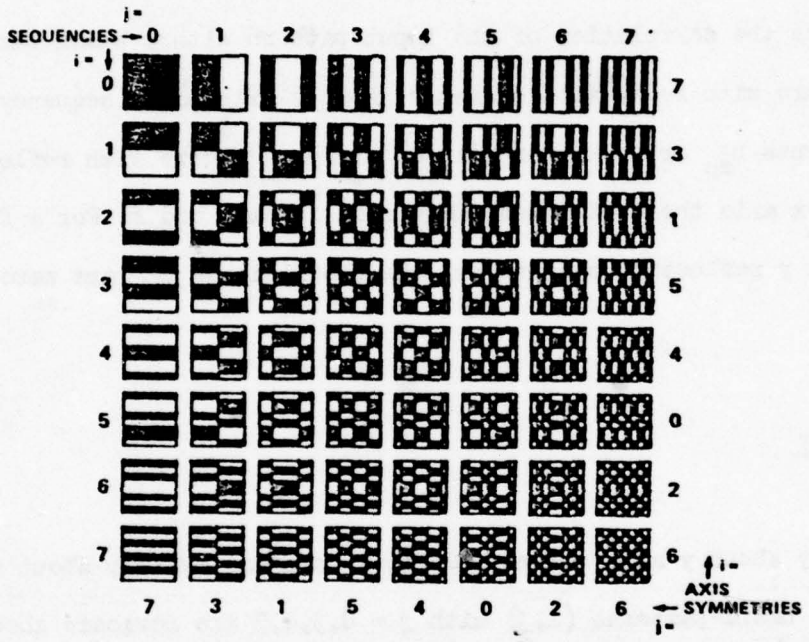


Fig. (3A-4) Walsh Basic Patterns. Basic Pattern (i, j) is Located at the Intersection of Row i and Column j. -1's have been Blanked

(a) REFLECTION (OR IMAGE) SYMMETRY

Walsh basic patterns (i,j) with even j's possess reflection symmetry about y axis. When the frame is folded about its central vertical axis a_2, the right half of the pattern matches the left half exactly. Basic patterns with even i's possess reflection symmetry about x axis. Since every sequency coefficient represents the correlation of the input pattern with a Walsh basic-pattern, a figure with reflection symmetry about y axis has a sequency spectrum whose coefficients b_{mn} are zero for all odd n. For a figure with reflection symmetry about x axis the coefficients are zero for all odd m. For a figure with both x and y reflection symmetries, the coefficients b_{mn} are zero for all odd m and n.

(b) PERIODICITY

Periodicity about y axis corresponds to axis-symmetry only about vertical axis a_2 . Walsh basic-patterns (i,j) with j = 0,3,4,7 are periodic about y axis. If a general binary two-dimensional pattern is periodic about y axis, the coefficients b_{mn} of its sequency spectrum are zero for n = 1,2,5,6 and for patterns periodic about x axis the coefficients are zero for m = 1,2,5,6, when a pattern is periodic about both x and y axes, coefficients b_{mn} with m,n = 1,2,5,6.

(c) REFLECTION SYMMETRY ABOUT MAIN DIAGONAL

Since basic-patterns (i,j) with i = j are symmetric about their main diagonal, the diagonal coefficients of a spectrum give a measure of unique axis-

symmetries and reflection symmetries about the main diagonal of a figure.

(d) GENERAL SYMMETRY

A pattern possesses general symmetry when it is symmetric about both x and y axes and about its main diagonal. When a symmetric figure is centered in a frame so that the x and y axes of the figure are the x and y axes of the frame, it yields a sequency spectrum which is also symmetric. The transpose of the spectrum is the spectrum itself except for a scale factor. If the image is markedly unsymmetric, the transform matrix will be too.

(e) If all the elements of the image are equal and nonzero, then the sequency spectrum has only one nonzero term, b_{00} . Conversely, a single nonzero results in a sequency spectrum having constant magnitude. Symmetry features may, therefore, be extracted from a pattern by operating on its Walsh sequency spectrum.

(2) Fast WHT algorithms involving $N\log_2 N$ additions and subtractions, do exist. For details see the interim report # 4.

(3) Let the data sequence $f_N = [\, f_0 \quad f \, , \ldots\ldots f_{N-1}]$ be subjected to dyadic shift and the new shifted data sequence be designated as $\underline{f}_d = \underline{f}_0 \quad \underline{f}_1 \ldots\ldots \underline{f}_{N-1}]$. Where $\underline{f}_k = f_N(k + s)$, $k = 0,1 \ldots\ldots N - 1$ and $+$ denotes modulo 2 addition. The following table illustrates the modulo 2 addition for N = 8

Making use of the above table one can evaluate $\underline{f}_k$ for $k = 0,1, \ldots\ldots 7$. For example

$$\underline{f}_1 = [f_1 \quad f_0 \quad f_3 \quad f_2 \quad f_5 \quad f_4 \quad f_7 \quad f_6]$$

k →

s ↓

k ⊕ s	0	1	2	3	4	5	6	7
0	0	1	2	3	4	5	6	7
1	1	0	3	2	5	4	7	6
2	2	3	0	1	6	7	4	5
3	3	2	1	0	7	6	5	4
4	4	5	6	7	0	1	2	3
5	5	4	7	6	1	0	3	2
6	6	7	4	5	2	3	0	1
7	7	6	5	4	3	2	1	0

It is easy to show that the dyadically shifted WHTs $\underline{F}_N$ are related to F_N in the following manner :

$\underline{F}_N^2 = F_n^2$, where $F_N = [F_0 \quad F_1 \ldots\ldots F_{N-1}]$ a set of one dimensional WHTs.

Therefore the WHTs are dyadic invariant. For details see (1).

(4) Since the WHTs are dyadic invariant, one can develop WHT power spectrum analogous to the Fourier representation. For example when N = 8 the power spectrum P_n for n = 0,1 ..... N/2 is given by

$$P_0 = F_0^2$$

$$P_1 = F_1^2 + F_2^2$$

$$P_2 = F_3^2 + F_4^2$$

$$P_3 = F_5^2 + F_6^2$$

$$P_4 = F_7^2$$

Thus the general form of the above spectrum is as :

$$P_0 = F_0^2$$

$$P_k = F^2(2k - 1) + F^2(2k) \qquad k = 1,2 \ldots\ldots N/2 - 1$$

$$P_{N/2} = F^2(N - 1)$$

Where P_n is the WHT power spectral point associated with the order of sequency n. Thus this spectrum has $(N/2 + 1)$ points.

The WHT phase spectrum is defined as

$$\phi_0 = 0, \Pi$$

$$\phi_{N/2} = 2m\,\Pi \pm \Pi/2 \qquad m = 0,1,2$$

and

$$\phi_k = \tan^{-1}\left[\frac{F_{(2k - 1)}}{F_{2k}}\right]$$

$$\phi_k = 1,2, \ldots\ldots\ldots N/2 - 1$$

Since the power spectrum is dyadic invariant one can define a correlation.

If $f_k \triangleq 1/N \sum_{h=0}^{N-1} f_k \; g_{k \oplus h}$

where $k = 0,1, \ldots\ldots N - 1$

then

$\hat{F}_k = F_k \; G_k$ where G_k is the WHT for the sequence $g_{k \oplus h}$

$k = 0, 1, \ldots\ldots\ldots\ldots N - 1$

Similarly the auto correlation function is defined as :

If $f_k \triangleq 1/N \sum_{h=0}^{N-1} f_k \; f_{k \oplus h}$

$\hat{F}_k = F_k^2 \qquad k = 0,1, \ldots\ldots N - 1$

The convolution is defined as :

If $f_k = 1/N \sum_{h=0}^{N-1} f_k \; g_{k \ominus h}$

$k = 0,1 \ldots\ldots N - 1$ where $\ominus$ denotes the modulo 2 subtraction.

$\tilde{F}_k = F_k \; G_k \qquad k = 0,1, \ldots\ldots N - 1$

Since the modulo 2 addition and subtraction are identical, there is no difference between dyadic correlation and convolution. Hence the linear filtering and matched filtering techniques can be applied in the domain of Walsh Hadamard functions for signal processing. For details see (4) & (5).

TWO-DIMENSIONAL HAAR TRANSFORM (HT) (1) :

The Haar transform (HT) coefficients F(k), k = 0,1, ..., N - 1 corresponding to a data sequence f_N = [f(0) f(1) ..... f(N - 1)], are obtained by computing the transformation

$$F_N = \frac{1}{N} H_n f_N \qquad (3A\text{-}10)$$

where $N = 2^n$

where H_n is the (N X N) Haar matrix. H_n is obtained by sampling the set of Haar functions [har(r,m,t)] defined in Eq. (3A-11a). For example, the (8 X 8) Haar matrix is given by Eq. (3A-11b).

$$\text{har}(0,0,t) = 1 \qquad t \in [0,1)$$

$$\text{har}(r, m, t) = \begin{cases} 2^{r/2}, & \frac{m - 1}{2^r} \leq t > \frac{m - 1/2}{2^r} \\ -2^{r/2}, & \frac{m - 1/2}{2^r} \leq t > \frac{m}{2^r} \\ 0, & \text{elsewhere for } t \in [0,1) \end{cases} \qquad (3A\text{-}11a)$$

where $0 \leqslant r < \log_2 N$ and $1 \leqslant m \leqslant 2^r$.

Sampling of the set of Haar functions in (3A-5a) results in the array shown in Fig. (3A-5), each row of which is a discrete Haar function Haar(r,m,t). Arrays so obtained are used in connection with the Haar transform and may be denoted by H_n, where $n = \log_2 N$.

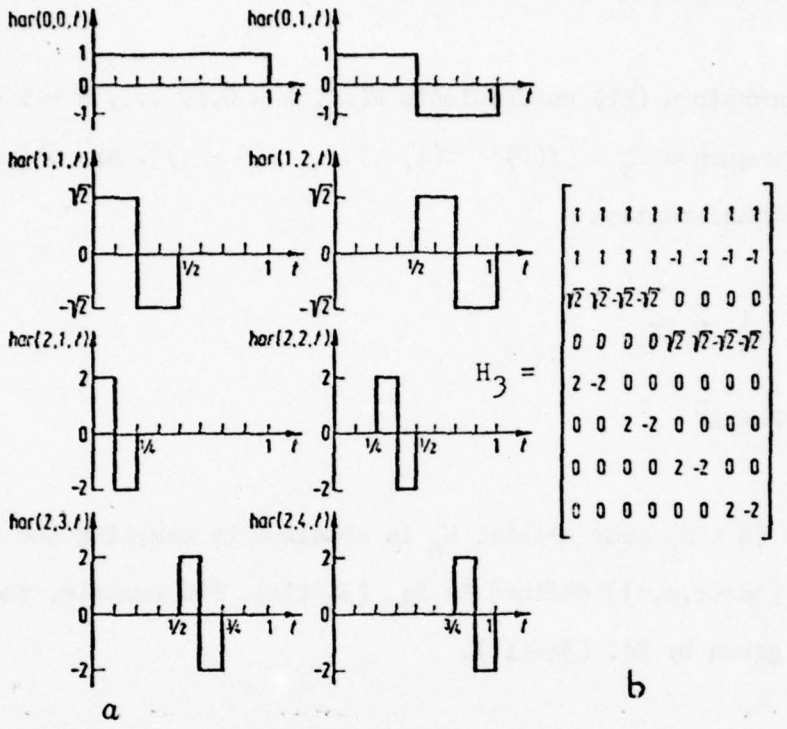


Fig.(3A-5a) a Continuous Haar Functions, N = 8; b Discrete Haar functions, N = 8

$$H_3 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} \\ 2 & -2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & -2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & -2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & -2 \end{bmatrix} \begin{matrix} \} N/N \\ \} N/N \\ \} \\ \} N/4 \\ \} \\ \} \\ \} \\ \} \end{matrix} N/2 \qquad (3A\text{-}11b)$$

Examining H_3 we observe that N/2 coefficients in the Haar domain measure the adjacent correlation of coordinates in the data space taken two at a time, N/4 measure corrdinates taken 4 at a time, etc. up to N/N coefficients measuring all the N coordinates of the data space.

The two-dimensional Haar transform of a data sequence f_{MN} of M rows and N columns is :

$$F_{MN} = 1/MN \; H_m \; f_{MN} \; H_N' \qquad (3A\text{-}12)$$

and $f_{MN} = H_n' \, F_{MN} \, H_n$ where $M = 2^m$ and $N = 2^n$

As in the case of WHT transforms F_{MN} can be computed using the one-dimensional Haar transform, a total of MN times as :

$$F_{MN} = \begin{bmatrix} F(0, 0) & F(0, 1) & F(0, N-1) \\ F(1, 0) & F(1, 1) & F(1, N-1) \\ & & \\ F(M-1, 0) & F(M-1, 1) & F(m-1, N-1) \end{bmatrix}$$

where F(i,j) is the (i , j)th transform coefficient. (3A-13)

PROPERTIES OF THE HTs

(1) The HT provides a domain that is both locally sensitive as well as globally sensitive. In the case of the discrete Fourier and Walsh-Hadamard transforms, each transform coefficient is a function of all coordinates in the ori-

ginal data space (global), where as this is true only for the first two Haar coefficients.

(2) The Cooley-Tukey type of algorithm to compute Fast HT requires $\log_2 N$ bit reversals and is implementable in $2(N - 1)$ additions and subtractions and N multiplications. For details see the interim report # 4.

(3) The HTs are dyadic invariant and hence possible to build a matched filter and not invariant to size and translation.

TWO-DIMENSIONAL SLANT TRANSFORM (1) :

A desirable property for an image coding transform is that the transform compacts the image energy to as few of the transform domain samples as possible. Qualitatively speaking, an efficient energy compaction will result if the basis vectors of the transformation matrix " resemble " typical horizontal or vertical lines of an image. In the case of a typical monochrome image, a large number of the lines are of nearly constant gray level over a considerable length. The Fourier, Hadamard, and Haar transforms possess a constant valued basis vector that provides an efficient representation for constant gray level image lines, while the Karhunen-Loeve transform has a nearly constant basis vector suitable for this representation. Another type of typical image line is the line that linearly increases or decreases in brightness over the length. None of the data transforms previously mentioned possess a basis vector that efficiently represents such image lines.

Orthogonal transformations containing a " Slant " basis vector are introduced. The slant vector is a discrete sawtooth waveform decreasing in uniform

steps over its length, and is suitable for efficiently representing gradual brightness changes in an image line. As illustrated in Fig. (3A-6). The slant transform is based on the slant vector which is an orthogonal matrix defined by Slant Matrix.

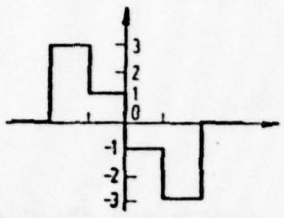


Fig. (3A-6) A Slant Vector for N = 4 and step size of 2 Units.

Slant Matrix Construction : If S_n denotes the (N X N) slant matrix for $N = 2^n$, than we define n matrices S_1, S_2, ... S_n as:

$$S_1 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \qquad (3A\text{-}14)$$

The slant matrix for N = 4 can be written as

$$S_2 = \frac{1}{\sqrt{4}} \begin{bmatrix} 1 & 1 & 1 & 1 \\ a+b & a-b & a+b & -a-b \\ 1 & 1 & 1 & 1 \\ a-b & -a-b & a+b & -a+b \end{bmatrix} \qquad (3A\text{-}15)$$

where a and b are real constants to be determined subject to the following conditions :

(1) Step size must be uniform, and

(2) S_2 must be orthogonal

The step size between the first two elements of the slant vector [see second row of S_2] is

$$(a + b) - (a - b) = 2b, \tag{3A-16}$$

and the step size between the second and third elements is

$$(a - b) - (-a + b) = 2a - 2b \tag{3A-17}$$

which leads to a = 2b

Hence $$S_2 = \frac{1}{\sqrt{4}} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 3b & b & -b & -3b \\ 1 & -1 & -1 & 1 \\ b & -3b & 3b & -b \end{bmatrix} \tag{3A-18}$$

Using the orthogonal condition

$$\frac{1}{\sqrt{4}}[3b\ b - b - 3b]\ \frac{1}{\sqrt{4}}\ [3b\ b - b - 3b]' = 1$$

we obtain

$$b = \frac{1}{\sqrt{5}}, \qquad a = \frac{2}{\sqrt{5}}$$

Thus the slant matrix in Eq. (3A-20) becomes

$$S_2 = \frac{1}{\sqrt{4}} \begin{bmatrix} 1 & 1 & 1 & 1 \\ \frac{3}{\sqrt{5}} & \frac{1}{\sqrt{5}} & \frac{-1}{\sqrt{5}} & \frac{-3}{\sqrt{5}} \\ 1 & -1 & -1 & 1 \\ \frac{1}{\sqrt{5}} & \frac{-3}{\sqrt{5}} & \frac{3}{\sqrt{5}} & \frac{-1}{\sqrt{5}} \end{bmatrix} \qquad (3A\text{-}19)$$

We observe that S_2 possesses the sequency property. It is easily seen that the sequencies of the rows of S_2 are 0,1, 1 and 2, which equal the sequencies of the corresponding rows of the Walsh-Hadamard matrix.

$$W_2 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix}.$$

Now S_2 can be expressed in terms of S_1 such that

$$S_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 0 & 1 & 0 \\ a_4 & b_4 & -a_4 & b_4 \\ 0 & 1 & 0 & -1 \\ -b_4 & a_4 & b_4 & a_4 \end{bmatrix} \begin{bmatrix} S_1 & 0 \\ 0 & S_1 \end{bmatrix} \qquad (3A\text{-}20)$$

where $a_4 = \frac{2}{\sqrt{5}}$, and $b_4 = \frac{1}{\sqrt{5}}$

Eq. (3A-20) can be generalized to give the slant matrix of order N/2 by the construction as shown in the Fig. (3A-7)

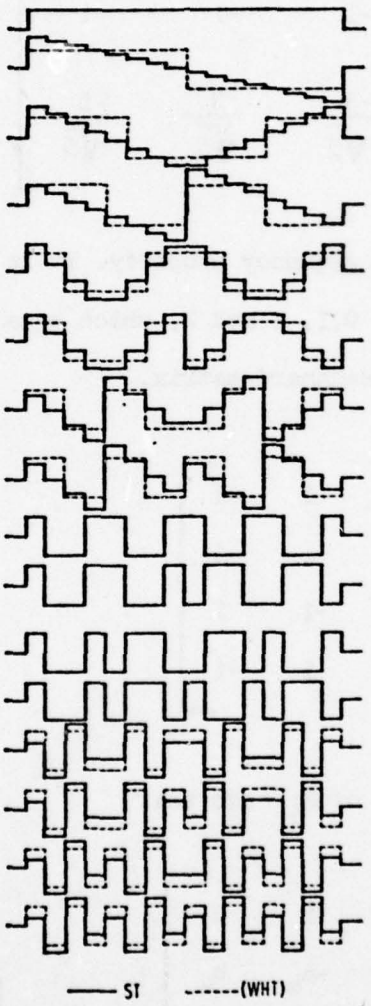


Fig.(3A-7) Comparison of WHT and ST basis Vectors for N = 16

Similarly the relation between S_2 and S_3 is given by (1)

$$S_3 = \frac{1}{\sqrt{8}} \left[\begin{array}{cccc:cccc} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ a_8 & b_8 & 0 & 0 & -a_8 & b_8 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ \hdashline 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 \\ -b_8 & a_8 & 0 & 0 & b_8 & a_8 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 \end{array}\right] \left[\begin{array}{cccc:cccc} 1 & 1 & 1 & 1 & & & & \\ \frac{3}{\sqrt{5}} & \frac{1}{\sqrt{5}} & \frac{-1}{\sqrt{5}} & \frac{-3}{\sqrt{5}} & & & & \\ 1 & -1 & -1 & 1 & & & & \\ \frac{1}{\sqrt{5}} & \frac{-3}{\sqrt{5}} & \frac{3}{\sqrt{5}} & \frac{-1}{\sqrt{5}} & & & & \\ \hdashline & & & & 1 & 1 & 1 & 1 \\ & & & & \frac{3}{\sqrt{5}} & \frac{1}{\sqrt{5}} & \frac{-1}{\sqrt{5}} & \frac{-3}{\sqrt{5}} \\ & & & & 1 & -1 & -1 & 1 \\ & & & & \frac{1}{\sqrt{5}} & \frac{-3}{\sqrt{5}} & \frac{3}{\sqrt{5}} & \frac{-1}{\sqrt{5}} \end{array}\right]$$

(3A-21a)

Similarly $S_n = \frac{1}{\sqrt{2}}$

$$\left[\begin{array}{cccc:cccc} 1 & 0 & & & 1 & 0 & & \\ a_N & b_N & & & -a_N & b_N & & \\ & & I_2 & & & & I_2 & \\ & & & I_2 & & & & I_2 \\ \hdashline 0 & 1 & & & 0 & -1 & & \\ -b_N & a_N & & & b_N & a_N & & \\ & & I_2 & & & & -I_2 & \\ & & & I_2 & & & & -I_2 \end{array}\right] \left[\begin{array}{c:c} S_{n-1} & 0 \\ \hdashline 0 & S_{n-1} \end{array}\right]$$

where I_2 denotes the (2 X 2) Indentity Matrix.

Where a_n and b_n are constants. In S_3 the slant vector is obtained by a simple scaling operation of S_2, while the remaining terms serve to obtain the sequency and orthogonal properties.

The coefficients (a_N , b_N) can be computed using the following relations

$$a_2 = 1$$

$$b_N = \frac{1}{(1 + 4a_{N/2}^2)^{1/2}}$$

$$N = 4,8,16,.....$$

$$a_N = 2b_N a_{N/2} \quad , \qquad (3A-22)$$

The two-dimensional Slant transform of f_{MN} data of M rows and N column is :

$$F_{MN} = S_m \; f_{MN} \; S_n'$$

and $$f_{MN} = S_m' \; F_{MN} \; S_n \qquad (3A-23)$$

where $M = 2^m$ and $N = 2^n$

As is the case with WHT transforms the two-dimensional transforms and the inverse can be computed using the one dimensional transform, a total of M X N times as :

$$F_{MN} = \begin{bmatrix} F(0, 0) & F(0, 1) \ldots\ldots\ldots\ldots\ldots F(0, N-1) \\ F(1, 0) & F(1, 1) \ldots\ldots\ldots\ldots\ldots F(1, N-1) \\ & \\ F(M-1, 0) & F(M-1, 1) \ldots\ldots\ldots\ldots F(M-1, N-1) \end{bmatrix}$$

where $F(i,j)$ is the (i,j)th transform coefficient.

PROPERTIES OF STs

(1) The spatial redundancy of color images and the limitations of human color vision can be exploited to achieve band width reduction for color image transmission (3).

(2) For detecting the gradual brightness changes in an image line, the STs are good.

(3) The transforms are not dyadic invariant and hence not useful for matched filtering techniques.

(4) Fast computational capability is poor and so far in the literature available no Fast Slant transform algorithms are reported.

DISCRETE COSINE TRANSFORMS (DCT) (1) :

It is known that the Fourier series representation of any continuous real and symmetric function contains only real coefficients corresponding to the

cosine terms of the series. This result can be extended to the discrete Fourier transform of images under proper interpretation. There are two ways in which an image field can be made symmetric, as shown in Fig. (3A-8). By the first technique the images are folded about an edge, and in the second method the images are folded and overlapped by one pixel. Hence for an N X N pixel image the Discrete Cosine Transforms can be defined as follows.

The DCT of a data sequence $f_N = [f(0)\ f(1)\ \ldots\ldots\ f(N-1)]$ is defined as

$$F(0) = \sqrt{\frac{1}{N}} \sum_{m=0}^{N-1} f(m)$$

and

$$F(k) = \sqrt{\frac{2}{N}} \sum_{m=0}^{N-1} f(m) \cos \frac{(2m+1)\Pi k}{2N} \qquad (3A\text{-}15)$$

$$k = 1,2, \ldots\ldots\ldots N-1$$

Where F(k) is the k-th DCT coefficient.

It is worthwhile noting that the set of basis vector elements

$$\left[\frac{1}{\sqrt{N}}, \qquad \sqrt{\frac{2}{N}} \cos \frac{(2m+1)\Pi k}{2N}\right]$$

is actually a class of discrete Chebyshev polynomials. This is easily seen by examining the following definition of Chebyshev polynomials

$$T_0(p) = \frac{1}{\sqrt{N}}$$

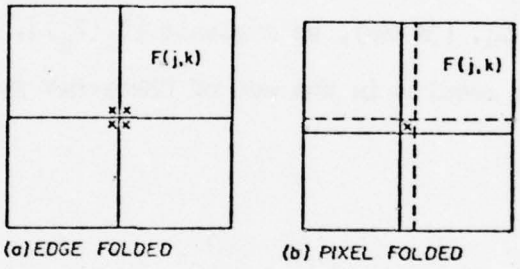
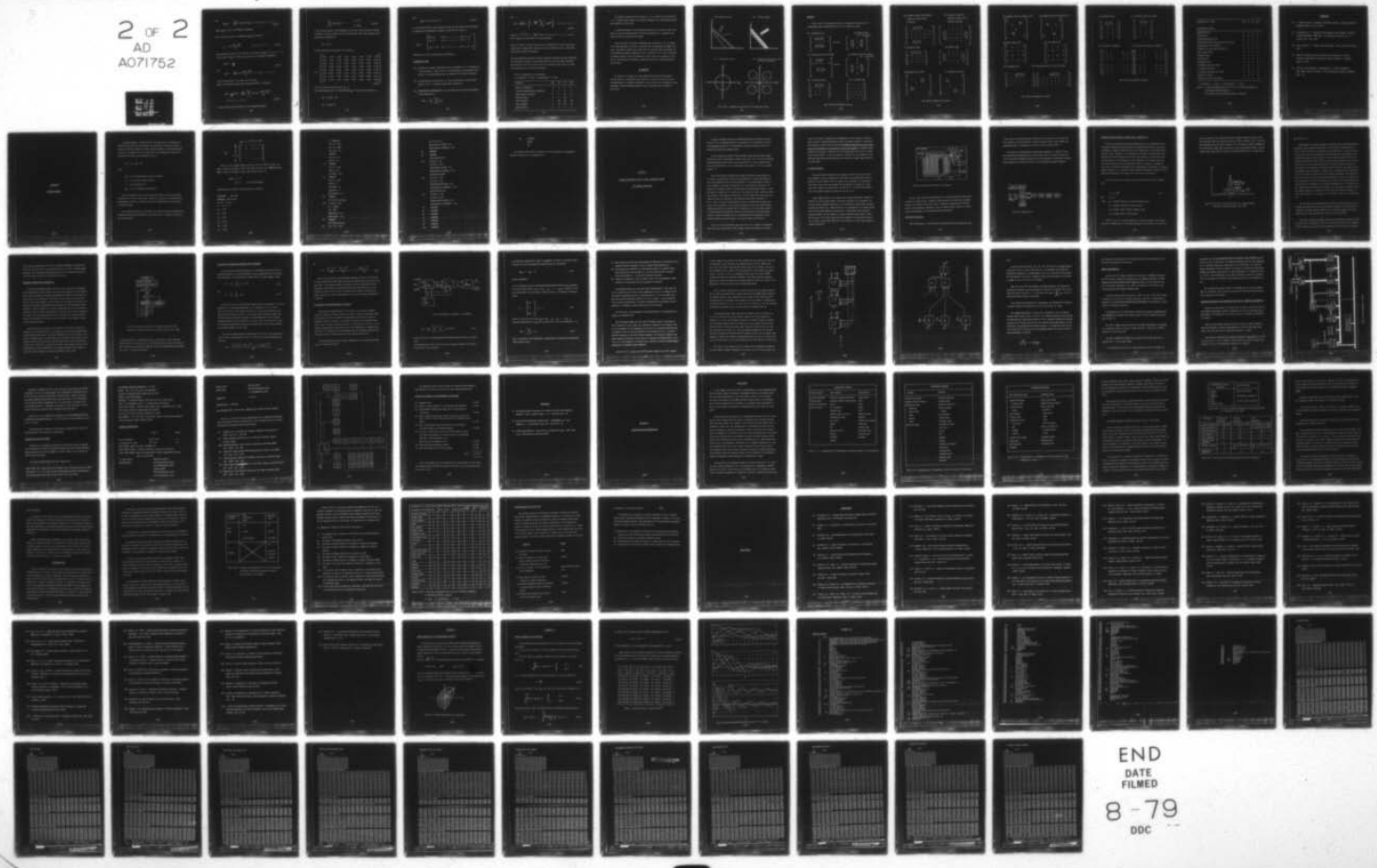


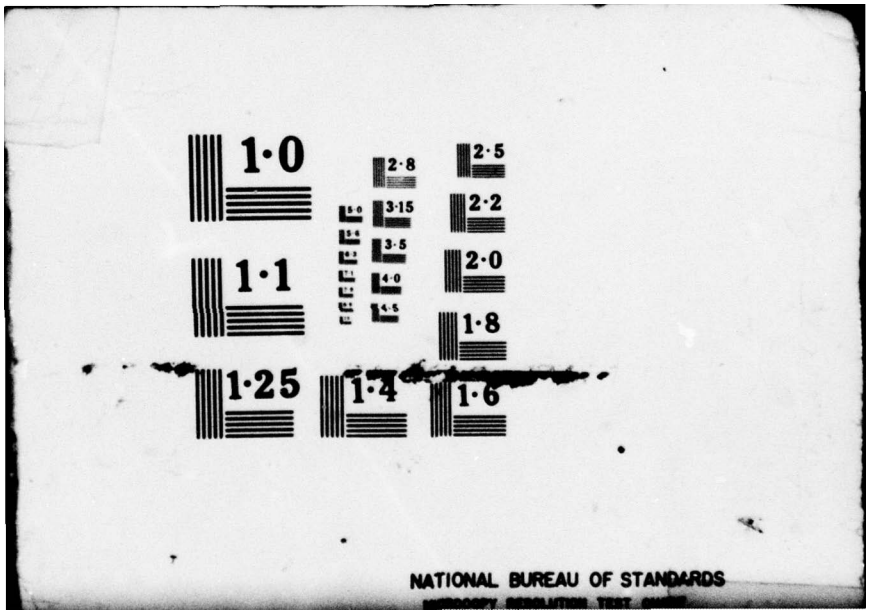
Fig. (3A-8) Cosine transform symmetry.

AD-A071 752 TELECOMMUNICATIONS ASSOCIATES FAIRFAX VA F/G 8/2
INVESTIGATION OF LINEAR TRANSFORMATIONS FOR AUTOMATIC CARTOGRAP--ETC(U)
APR 79 R L PICKHOLTZ, M MOVAHED, S S MURTY DAAK70-78-C-0045
UNCLASSIFIED TA-79-1-1 ETL-0181 NL

2 OF 2
AD
A071752



END
DATE FILMED
8-79
DDC


1·0
1·1
1·25
2·8
3·15
3·5
4·0
4·5
2·5
2·2
2·0
1·8
1·4
1·6
NATIONAL BUREAU OF STANDARDS

and

$$T_k(Z_m) = \sqrt{\frac{2}{N}} \cos[k \cos^{-1}(Z_m)] \quad k, m = 1,2,\ldots\ldots N - 1 \tag{3A-26}$$

where $T_k(Z_m)$ is the kth Chebyshev polynomial.

Now, the zeros of the Nth polynomial $T_N(Z_m)$ are given by

$$Z_m = \cos \frac{(2m + 1)\Pi}{2N} \qquad m = 0, 1, \ldots, N - 1 \tag{3A-27}$$

Substituting Eq. (3A-27) in Eq. (3A-26), we evaluate $[T_l(Z_m)]$, $l = 0,1,\ldots. N - 1$ at the zeros of $T_N(Z_m)$. This results in the set of Chebyshev polynomials

$$T_0(m) = \frac{1}{\sqrt{N}} \tag{3A-28}$$

and

$$T_k(m) = \sqrt{\frac{2}{N}} \cos \frac{(2m + 1)k\Pi}{2N}, \quad m = 0, 1, \ldots\ldots, N - 1$$

which are equivalent to the basis set of the DCT.

Again, the inverse discrete cosine transform (IDCT) is defined as

$$f(m) = \frac{1}{\sqrt{N}} F(0) + \sqrt{\frac{2}{N}} \sum_{k=1}^{N-1} F(k) \cos \frac{(2m + 1)k\Pi}{2N},$$

$$m = 0, 1, \ldots\ldots, N - 1 \tag{3A-29}$$

It can be shown that application of the orthogonal property

$$\sum_{m=0}^{N-1} T_p(m)\, T_q(m) = \begin{cases} 1, & p = q \\ 0, & p \neq q \end{cases} \qquad (3A\text{-}30)$$

to Eq. (3A-29) results in the definition of the DCT. If Eq. (3A-25) is written in matrix form and D_n denotes the (N X N) DCT matrix, then this orthogonal property can be expressed as

$$D_n' \quad D_n = I_n$$

For the purposes of illustration, we evaluate D_3

$$D_3 = \begin{bmatrix} 0.354 & 0.354 & 0.354 & 0.354 & 0.354 & 0.354 & 0.354 & 0.354 \\ 0.490 & 0.416 & 0.278 & 0.098 & -0.098 & -0.278 & -0.416 & -0.490 \\ 0.462 & 0.191 & -0\ 191 & -0.462 & -0.462 & -0.191 & -0.191 & 0.462 \\ 0.416 & -0.098 & -0.490 & -0.278 & 0.278 & 0.490 & 0.098 & -0.416 \\ 0.354 & -0.354 & -0.354 & 0.354 & 0.354 & -0.354 & -0.354 & 0.354 \\ 0.278 & -0.490 & 0.098 & 0.416 & -0.416 & -0.098 & 0.490 & -0.278 \\ 0.191 & -0.462 & 0.462 & -0.191 & -0.191 & 0.462 & -0.462 & 0.191 \\ 0.098 & -0.278 & 0.416 & -0.490 & 0.490 & -0.416 & 0.278 & -0.098 \end{bmatrix} \qquad (3A\text{-}32)$$

It can be easily verified that $D_3'\ D_3 = I_3$

The two-Dimensional DCT of a data sequence f_{MN} of M rows and N columns is :

$$F_{MN} = D_m\, f_{MN}\ D_3' \,.$$

$$f_{MN} = D_m'\, F_{MN}\ D_3$$

and

$$\text{where } M = 2^m \text{ and } N = 2^n \qquad (3A\text{-}33)$$

As in the case of WHT, the two-dimensional DCT and the inverse can be computed by using the one dimensional transform a total of M X N times as

$$F_{MN} = \begin{bmatrix} F(0, 0) & F(0, 1) \ldots\ldots\ldots\ldots F(0, N-1) \\ F(1, 0) & F(1, 1) \ldots\ldots\ldots\ldots F(1, N-1) \\ & \\ F(M-1, 0) & F(M-1, 1) \ldots\ldots\ldots F(M-1, N-1) \end{bmatrix}$$

where F(i,j) is the (i,j)th transform coefficient.

PROPERTIES OF DCTs

(1) The DCTs give superior performance and is comparable to K - L transform in ' rate distortion ' applications while maintaining the computational simplicity of a transform which does not depend on the picture statistics.

(2) By simulation one can show that the DCT is equivalant in a mean square sense to the K - L Transform under basis restriction.

(3) COMPUTATIONAL CONSIDERATIONS. It can be shown that the DCT can be equivalently expressed as

$$F(0) = \frac{1}{N} \sum_{m=0}^{N-1} f(m)$$

and

$$F(k) = \sqrt{\frac{2}{N}}\, \mathrm{Re} \left\{ e^{-\frac{ik\Pi}{2N}} \sum_{m=0}^{2N-1} f(m) W^{km} \right\}, \quad k = 1,2, \ldots, N-1 \tag{3A-34}$$

where $W = e^{-i2\Pi/2N}$, $i = \sqrt{-1}$, $f(m) = 0$, $m = N, N+1, \ldots, N-1$, and Re [.] denotes the real part of the term enclosed.

From Eq. (3A-34) it follows that all the N coefficients of the DCT can be computed using a 2N-point FFT. Similarly it can be shown that a 2N - point IFFT yields all the IDCT coefficients.

The fast transform algorithms for Walsh Hadamard transforms and Haar transforms are discussed in detail in the interim report # 4. The fast slant transform algorithm does not seem to exist so far. The following table shows the important properties of the Discrete transforms.

Table 1: Properties of the transforms.

A = Excellent B = Good C = Satisfactory D = Fair

Property	WHT	HT	ST	DCT
Speed of computation	B	A	D	C
Hardware implementation feasibility	B	A	D	C
Dyadic shift Invariance	X	X	NO	NO
Matched filter	X	X	NO	X
Power spectrum	X	NO	NO	NO
Phase spectrum	X	NO	NO	NO
Edge detection	B	C	A	D

The candidate features will be shown on a N X N grid and will be given by a N X N matrix. Each element of the matrix represents the corresponding brightness intensity at that point.

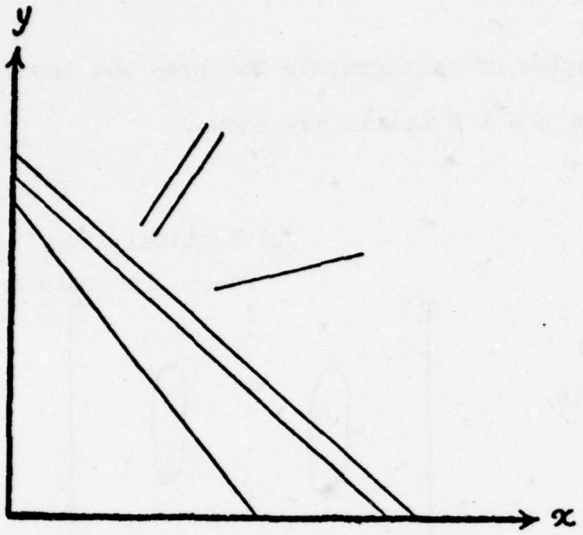
A transformed matrix can have negative elements as its entries. Since here again we are only concerned about the visual examination of a transform, the absolute value of entries will be taken.

Since the result of discrete transformations is given by a set of points, it is very difficult to come to a conclusion just by observing one sample. However, some general remarks can be made about the results. After evaluating the two-Dimensional Discrete transforms we set a threshold so that only the entries of greatest absolute value are preserved. Table 2 gives the comparative statement in recommending the different transform techniques in extracting different patterns.

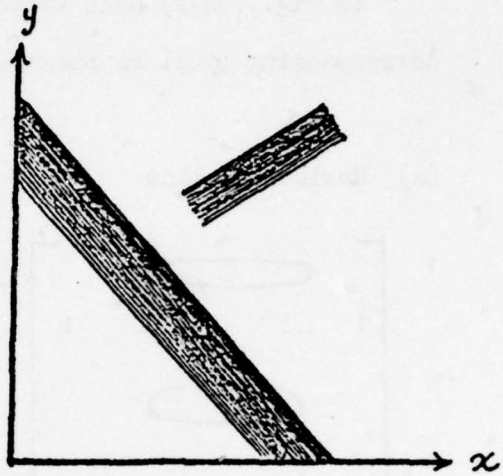
3B: EXAMPLES

In section B of chapter two, some candidate features from the objects usually present in photographs were selected. We used those features to show the application of continuous transforms. In this section we go through the same procedure. The same candidate features will be used and they are shown in Fig. (3B-1).

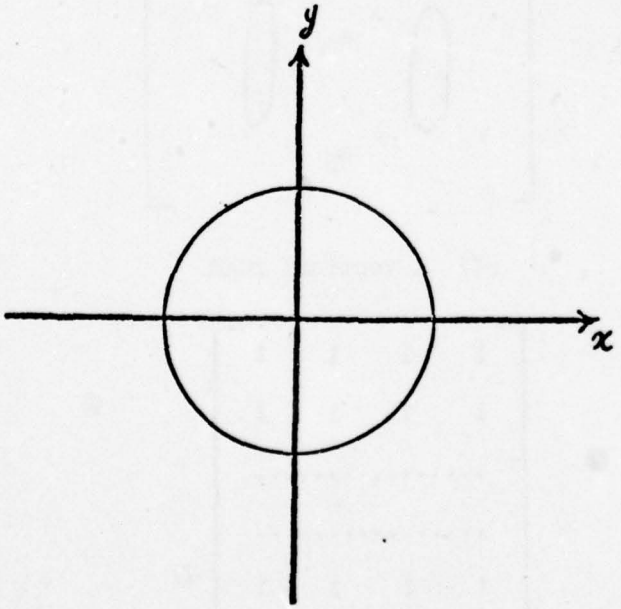
(a) Slanted Lines



(b) Shaded Edges



(c) Centered Circle



(d) Cluster of Circles with Circular Symmetry

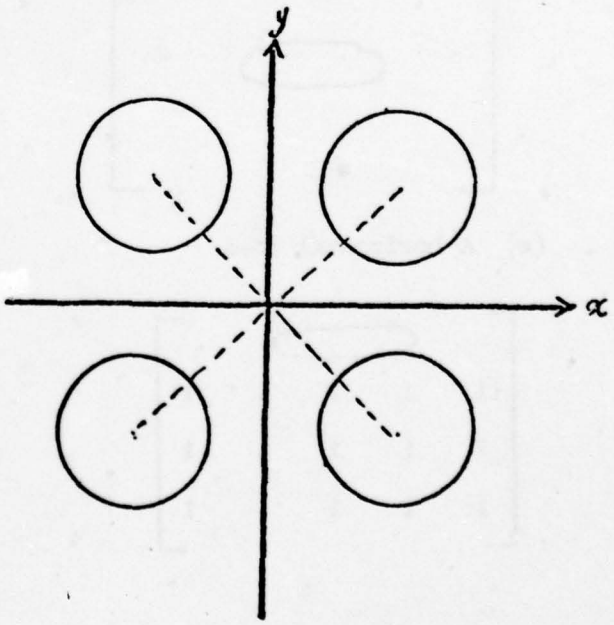
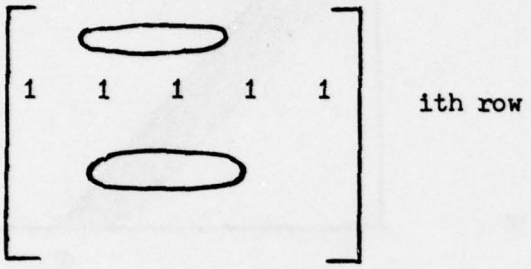


FIG. (3B-1) Candidate features for transforms study

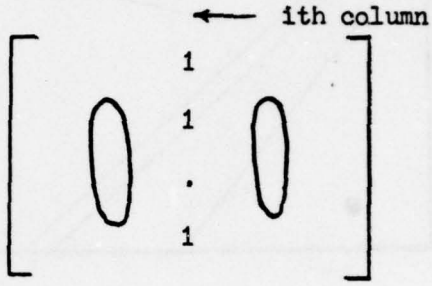
EXAMPLES

In Fig. (3B-2) some typical examples of cartographic features and the corresponding pixel representations by a N X N matrix are shown.

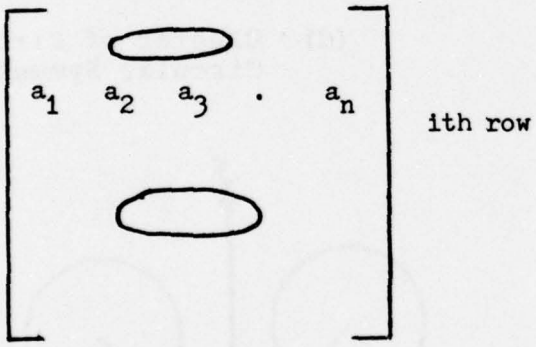
(a) Horizontal line



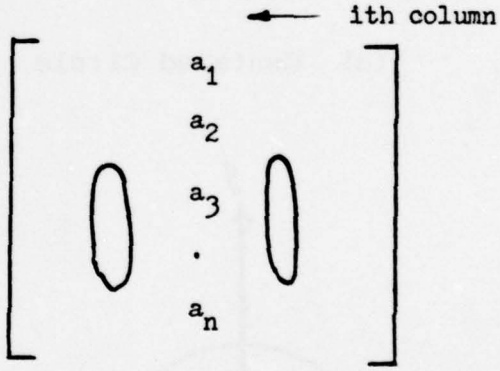
(b) Vertical line



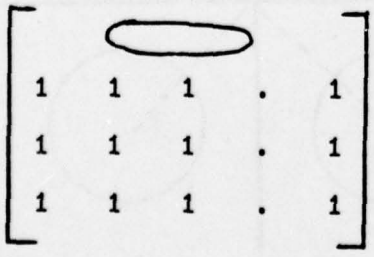
(c) Horizontal line with shades



(d) Vertical line with shades



(e) A horizontal road



(f) A vertical road

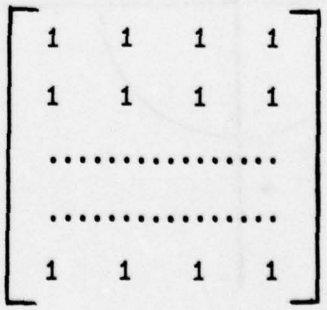
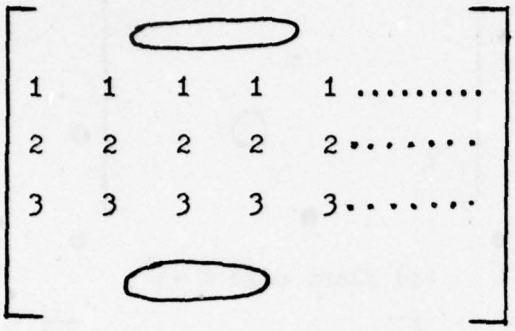
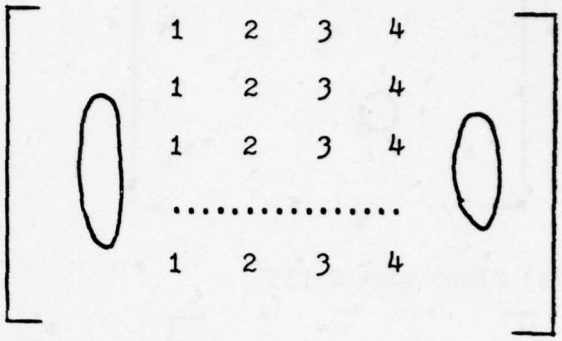


Fig. (3B-2a) Cartographic features

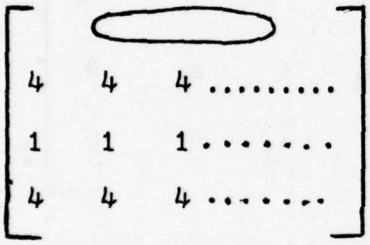
(g) A patch of land with different shades in the Horizontal direction.



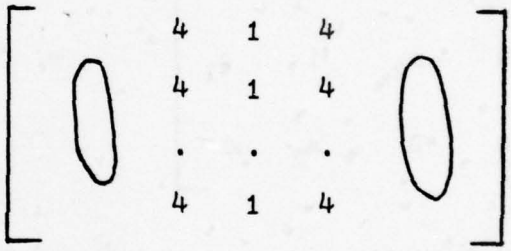
(h) A patch of land with different shades in the Vertical direction



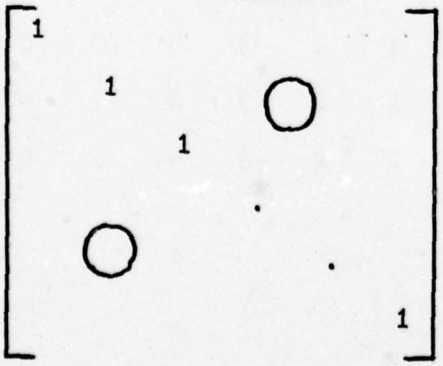
(i) Horizontal edge



(j) Vertical edge



(k) Diagonal line @ 135°



(l) Diagonal line @ 45°

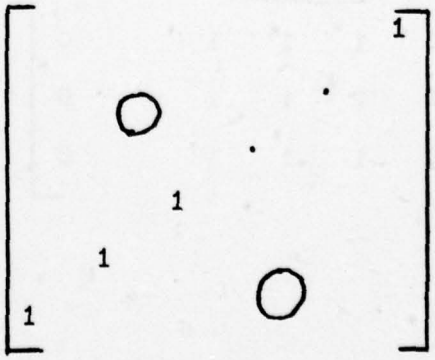
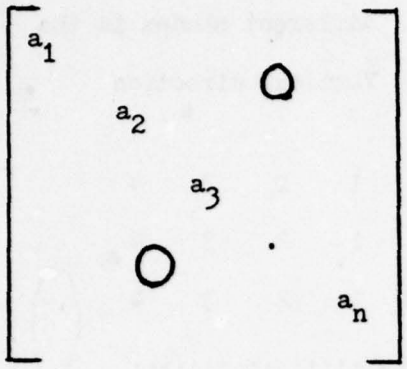
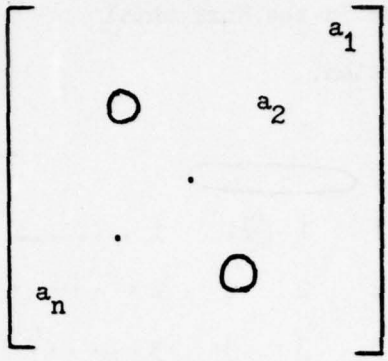


Fig. (3B-2b) Cartographic features

(m) Diagonal line with shades @ 135°



(n) Diagonal line with shades @ 45°



(o) Slant edge @ 135°

$$\begin{bmatrix} 1 & a_1 & 0 & 0 \\ a_2 & 1 & a_1 & 0 \\ 0 & a_2 & 1 & a_1 \\ 0 & 0 & a_2 & 1 \end{bmatrix}$$

(p) Slant edge @ 45°

$$\begin{bmatrix} 0 & 0 & a_1 & 1 \\ 0 & a_1 & 1 & a_2 \\ a_1 & 1 & a_2 & 0 \\ 1 & a_2 & 0 & 0 \end{bmatrix}$$

(q) Rectangular box

$$\begin{bmatrix} & & & & & \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix}$$

(r) Rectangular patch of land with shades

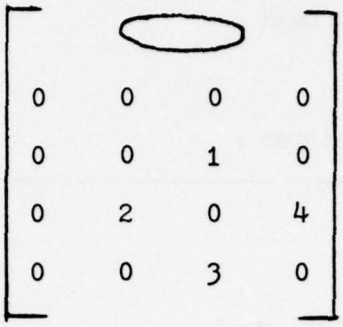
$$\begin{bmatrix} & & & & & & & \\ 0 & 0 & 1 & 2 & 3 & 4 & . & 0 \\ 0 & 0 & 1 & 2 & 3 & 4 & . & 0 \\ 0 & 0 & 1 & 2 & 3 & 4 & . & 0 \end{bmatrix}$$

Fig. (3B-2c) Cartographic features

(s) Centered circle

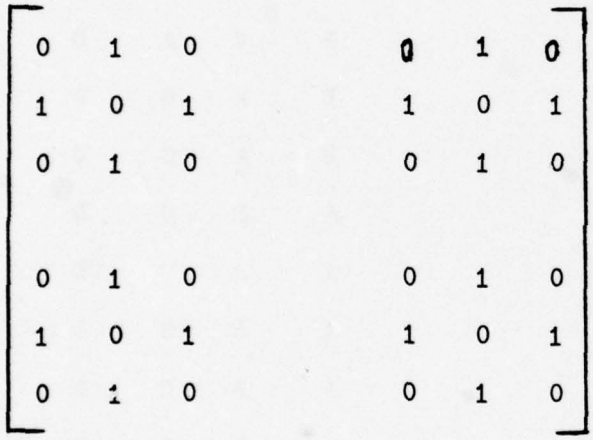
0	0	0	0
0	0	1	0
0	1	0	1
0	0	1	0

(t) Centered circle with shades



0	0	0	0
0	0	1	0
0	2	0	4
0	0	3	0

(u) 4 Circles of symmetry



0	1	0		0	1	0	0	1	0
1	0	1		1	0	1	2	0	4
0	1	0		0	1	0	0	3	0
0	1	0		0	1	0	0	1	0
1	0	1		1	0	1	2	0	4
0	1	0		0	1	0	0	3	0

(v) 4 Circles with shades of symmetry

0	1	0		0	1	0	0	1	0
2	0	4		2	0	4	2	0	4
0	3	0		0	3	0	0	3	0
0	1	0		0	1	0	0	1	0
2	0	4		2	0	4	2	0	4
0	3	0		0	3	0	0	3	0

Fig. (3B-2d) Cartographic features

Techniques to be used Cartographic features	WHT	HT	ST	DCT
Horizontal line	A	B	C	D
Vertical line	A	B	C	D
Horizontal/Vertical road	A	B	C	D
Horizontal/Vertical patch of land with colors	A	B	C	D
Diagonal line with @ $45^{o}/135^{o}$	B	A	C	D
Diagonal line with shades @ $45^{o}/135^{o}$	A	B	C	D
Slant edge $45^{o}/135^{o}$	B	C	A	D
Horizontal edge	B	A	C	D
Vertical edge	B	A	C	D
Rectangular area	A	B	C	D
Rectangular area with shades	A	B	C	D
A circular enclosure	A	B	C	D
A circular enclosure with shades	A	B	C	D
Cluster of circles	A	B	C	D
Cluster of circles with shades	A	B	C	D

A = Excellent B = Good C = Satisfactory D = Fair

Table 2 : A detailed recommendation for the use of different techniques in extracting different patterns.

For details see the computer printout in chapter 4.

REFERENCES

(1) N. Ahmed, K.R.Rao, " Orthogonal transforms for digital signal processing " Springer Verlag - New York 1975.

(2) Alexandridis N. A. " Relations among sequency, axis symmetry, and period of WALSH FUNCTIONS " IEEE Trans. on Information Theory Vol. 1T - 17 pp 495 - 497 , July 1971.

(3) Pratt, William, K " Digital Image Processing ", Willey Science, New York, 1978.

(4) Harmuth H: Sequency Theory, Foundations and Applications, (1977), Advances in Electronics and Electron Physics Supplement 9, Academic Press, New York.

(5) Pichler F and Quatember B: Implementation of a Dyadic Correlator, Proc. IEEE - EMC Sym. Montreux, (1975) Supplement on Sequency Techniques pp22 - 25.

CHAPTER IV

COMPUTER PROGRAM

A software package was developed for the evaluation of two-dimensional WHTs, HTs, STs, and DCTs in processing the data pixels. The entire program is written in Fortran IV WATFIV. For fast compiling process, the software package can be run in Fortran IV H Level. Right now the two-dimensional transform is computed by the brute force technique, such as:

$$F_{MN} = A_m \cdot f_{MN} \cdot A'_n$$

where

F_{MN} : The two-dimensional discrete transform.

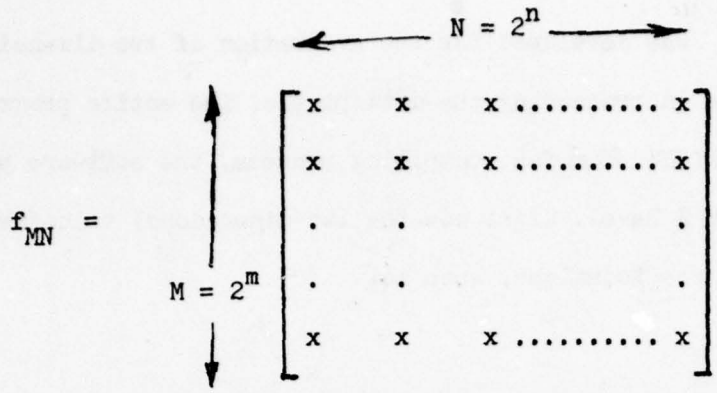
A : The Discrete transform matrix

A' : The transpose of A

F_{MN} : The two-dimensional data pixels

The Walsh Hadamard, Haar, Slant and Discrete Cosine matrices are readily available in the form of subroutines. Whenever the F_{MN} has to be evaluated the corresponding subroutine is called for and the multiplication of the matrices is carried through.

This program enables one to calculate all the four Discrete two-dimensional transforms, given the data pixels in the discrete form. It accepts the data in the following form:



Right now this package accepts the data of 32 X 32 pixels. In case one wants to extend the domain to 256 X 256 pixels of data, the " DIMENSION STATEMENTS " have to be changed to 256 X 256, instead of 32 X 32.

Change $M_1 = 8$

$M_2 = 8$ in the main program.

The modified sub routine for Haar matrix is as follows :

```
SUBROUTINE      HAR (E,M)
DIMENSION   E(256, 256)
DO 100    M = 2,8
N     = 2 * * M
N1    = N/2
N4    = N/4
N5    = N/8
N6    = N/16
N7    = N/32
N8    = N/64
```

```
      S = SQRT(2,0)
      DO 5 I1 = 1,N)
      DO 5 J1 = 1,N)
      F(I1,J1) = 0,0
5     CONTINUE
      DO 10 I = 1,N
      E(1,I) = 1,0
10    CONTINUE
      DO 20 J = 1,N1
      E(2,J) = 1,0
      E(2,J+N1) = -1,0
20    CONTINUE
      DO 30 K = 1,N4
      E(3,K) = S
      E(3,K+N4) = -S
      E(4,K+N1) = S
      E(4,K+N1+N4) = -S
30    CONTINUE
      IF (M-2) 120,120,150
150   DO 40 KK = 1,N1
      KL = (2*KK)-1
      KM = 2*KK
      E(N1+KK,KL) = 2,0
      E(N1+KK,KM) = -2,0
40    CONTINUE
      IF (M-4)120,140,140
140   DO 50  K1 = 1,N5
```

```
      DO  70 L1 = 1,4
      E(4+L1,K1+(L1-1)*N4) = 2,0
      E(4+L1,K1+(L1-1)*N4+N5) = -2,0
70    CONTINUE
50    CONTINUE
      IF(M-5)120,125,125
125   DO 80 L2 = 1,8
      DO 60 K2 = 1,N6
      E(8+L2,K2+(L2-1)*N5) - 2,0
      E(8+L2,K2+(L2-1)*N5+N6) = -2,0
      IF(M-6)120,126,126
126   DO 81 L3 = 1,16
      DO 61 K3 = 1,N7
      E(16+L3,K3+(L3-1)*N6) = 2.0
      E(16+L3,K3+(L3-1)*N6+N7) = -2.0
      IF(M-7)120,127,127
127   DO 82 L4 = 1,32
      DO 62 K4 = 1,N8
      E(32+L4,K4+(L4-1)*N7) = 2.0
      E(32+L4,K4+(L4-1)*N7+N8) = -2.0
62    CONTINUE
82    CONTINUE
61    CONTINUE
81    CONTINUE
60    CONTINUE
80    CONTINUE
120   CONTINUE
```

```
100     CONTINUE
        RETURN
        END
```

The Computer Program and the results for various examples of Cartographic features are enclosed in the Appendix III.

CHAPTER V

HARDWARE FEASIBILITY STUDY OF SIGNAL PROCESSING SYSTEM

FOR DISCRETE TRANSFORMS

A signal processing system for implementing Discrete transform functions consists of (1) an Imaging system for the pattern to be identified (2) a Data base management system (3) a Two-dimensional transform processor and (4) an Interactive Raster scan display system.

In this chapter we propose a Microprocessor based special purpose signal processor for use in cartographic studies which makes use of the two-dimensional Hadamard/Haar/Slant/and Discrete Cosine transforms. All these discrete Transforms are evaluated for an N X N dimensional array where $N = 2^n$ for n being an integer 0.

Some of the factors influencing the signal processing system design are (1) speed of computations (2) Ease of implementation by hardware/software in real time applications (3) cost effictiveness. In the cartographic studies of Aerial Imagery, the speed of computation is a very important criterion, for example the strategic Bombers, the high altitude reconnaissance planes,....etc. The speed of computation is a function of the number of pixels chosen for imaging a pattern. The bigger the dimension of the two-dimensional pixel array the higher the time of computation. One reasonable pixel array dimension is 256 X 256 . The type of Microprocessor chosen for realizing the signal processor must be capable of high speed acquisition of Data, high speed Memory addressing capability, hardware built in capability to do high speed Arithmetic processing such as Multiply, Divide, Add and Subtract in floating point notation, simple instruction set for communicating with peripherals in full duplex operation and handling high volume data storage and retrieval.

Only a 16 bit Microprocessor can meet such a set of demands as mentioned above. The most recent Intel's 8086, Zilog's Z 8000 and Motorola's MC 68000,

stand in the order of superiority of performance. In this chapter we propose a system based on Z-8000 Microprocessor which is already available in the market even though we strongly feel the Motorola MC 68000 Microprocess is (not yet released in the market) the outstanding one for this purpose. The general design approach of the system discussed in the following pages can as well be implemented with MC 68000 in case one decides to proceed at a later stage for high speed and high volume data applications for the same or larger image arrays as the need arises.

AN IMAGING SYSTEM :

The system under discussion is for Imaging a pattern into 256 X 256 picture elements. A special purpose 256 X 256 Element area Image sensor (CCD) is required for this purpose. With the present day technology it is possible to build such a device from the standard LSI techniques. The sensor is a solid state self scanned area image sensor suitable for use in camera type applications. The general schematic is shown in the fig. (5-1).

Light energy incident on the image sensor elements generates a packet of electrons at each sensing element. Electrical clocking of the photogate, the vertical analog transport registers, and the horizontal analog output register sequentially delivers the charge packets to the preamplifier. The charge packets from the horizontal register are sensed by a floating gate whose potential changes linearly with the quantity of signal charge and which drives a first MOS transistor. The output signal from the transistor in turn drives the gate of an output n-channel MOS transistor which produces the video output signal at terminal VIDEO OUT.

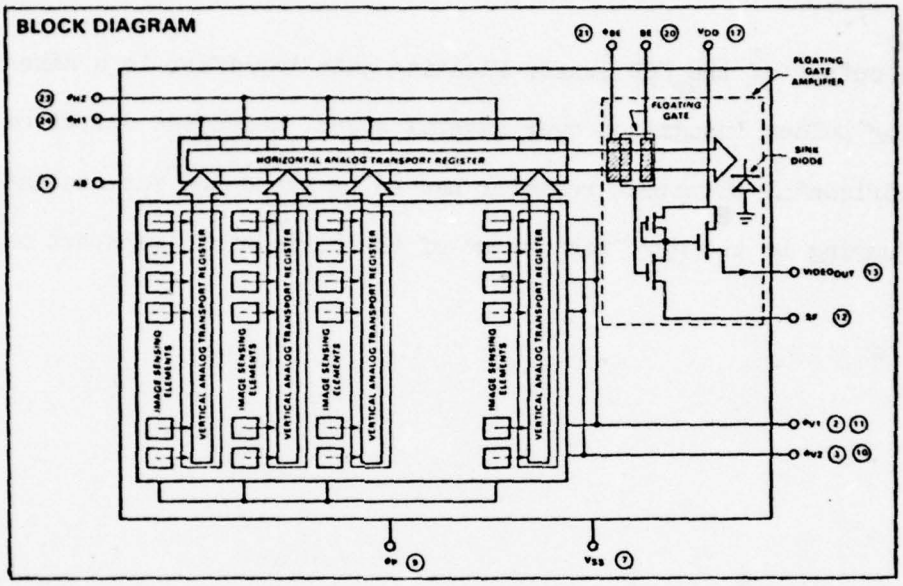


Fig.(5-1) Image sensor (Courtesy of Fairchild)

The fig. (5-2) shows the block diagram of an Imaging system. The CCD sensor is driven by clock drivers ordered by logic generator for sequential scanning purposes. The output of this CCD sensor is properly processed by the clamper and the video Amplifier combination. Thus the Imaging system output is a video output corresponding to the pattern scanned sequentially.

DATA BASE MANAGEMENT:

Each analog pulse is the output of a particular detector during one scan

of the array. The desired signals contained in the intensity of the light are now contained in the photodetector outputs. Normally, these output signals are contaminated with a fluctuating, as well as a fixed, pattern noise.

The output of the CCD Sensor Floating gate Amplifier is a stream of several analog pulses integrated over several scans. Hence the output of each row of the horizontal transport register has to be processed for statistical average before jumping to the next row output of the horizontal transport register.

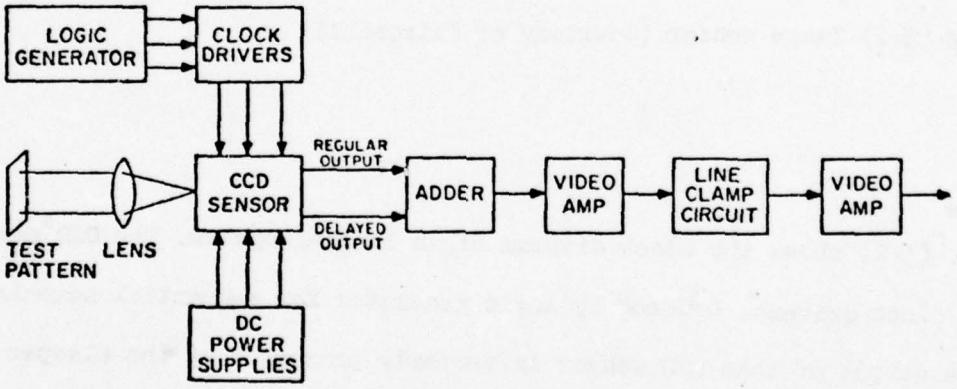


Fig.(5-2) Imaging set up

UNCORRELATED NOISE REDUCTION THROUGH SIGNAL AVERAGING (1)

The first type of fluctuating photodetector noise to be minimized is the inevitable thermal noise that arises in all semiconductor devices. This noise contributes to random fluctuations about the average value of each array photodetector output. Referred to as uncorrelated, the noise on one detector is unrelated with the fluctuating noise on any other detector. In some optical processing systems, it is necessary to detect changes in the average output signal that are smaller than the fluctuations about the average. A straightforward method to maintain the agerage while minimizing the fluctuations is through signal averaging. It is logical to apply central limit theorm to derive a set of statistical averages of the signals as in our case as shown below.

It can be shown when n samples from the same distributions are averaged then

$$\sigma = \sigma_1 \sqrt{n} \qquad (5\text{-}1)$$

and $a_0 = a_i$

where σ_0 = Standard deviation of signal averager output.

σ_i = Standard deviation of input samples.

a_0 = Average value of signal averager output

a_i = Average value of input samples.

Values σ_0 and σ_1 can also represent the rms noise levels of the output and input, respectively, of the signal averager. Note from Eqs. (5-1) and (5-2)

that the average of the output values from a signal averager is equal to the input average, while the variations about the output average (σ_0) is less than the variations about the input average (σ_1). The result of signal averaging is that the desired signal (a_0) is preserved, while the undesired signal (σ_0) decreases. Fig. (5-3) shows the input and output distributions of a signal aver-

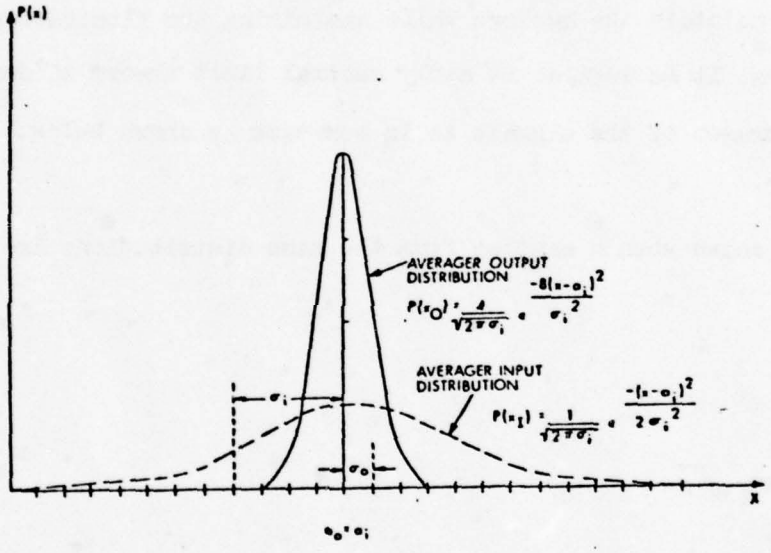


Fig.(5-3) Input and output distributions of a signal averager
(Courtesy of Computer Design, July, 1978)

ager with n = 16.

Implementation of signal averaging is simplified with the general-purpose laboratory signal processor making use of the new Z8000 microprocessor. An array of 24-bit locations is established and initialized to 0 as data accmulates in memory. One data accumulator must be reserved for each detector in the photodetector array. An equal number of 24-bit locations (input block) is reserved in memory, which will contain the input values obtained via the A-D converter each time the photodetector array is scanned. After each array scan, the content of each location in the input block is added to the content of its corresponding data accumulator. A register must be established and initialized to count the number of sums taken in the averaging algorithm. This counter register can be one of the 24-bit registers internal to the CPU or a memory location, assuming that not more than 255 sums are taken. This counter register is initialized at the beginning of the program decremented by 1, and checked for 0 each time the data accumulators are updated. When the register content reaches 0, the summing operation is completed. The CPU must then divide the content of each data accumulator by the number of sums taken to arrive at the final average.

The new Z8000 has integral hardware multiply and divide facility. Thus a software algotithm which was earlier required to carry out this operation, is replaced by a single hardware instruction. In signal averaging, the CPU can easily access to the data with more ease than an 8-bit machine.

Note that the signal averaging technique will minimize the effect of thermal noise from a photodetector only if the standard deviation of the detector noise is greater than the rms of the quantizing noise from the A-D converter.

If the least significant bit of the A-D converter represents a voltage level of K volts, then the rms of its quantizing noise is $K/\sqrt{12}$. Therefore signal averaging will minimize any random analog noise if the standard deviation of the noise signal is equal to or greater than K volts.

CORRELATED DETECTOR NOISE CANCELLATION

In optical processing systems, there is a noise type that is correlated among photodetector outputs because it causes output signals of all photodetectors to either increase or decrease simultaneously. Sources of this noise are fluctuations in the intensity of the light or laser source and in the photodetector array power supply. A simple method used to minimize this noise is to establish a reference photodetector that monitors only the unmodulated laser intensity. The input signal to the optical processing device can be limited in bandwidth so that the first photodetector of the array receives only unmodulated laser light. This photodetector can be considered the reference detector because large changes in its output will be due only to laser light or array power supply fluctuations.

After each scan of the array, the value from the reference detector is compared to the reference value of the previous scan. With no correlated noise present between array scans, the present reference value will be equal to the previous reference value. If correlated noise is present between scans, the reference signals will not be equal, and the present values in the input block of data must be modified to eliminate the effect of the correlated noise. This modification is easily implemented in the signal processor either before or after any signal averaging by using the flowchart of fig. (5-4).

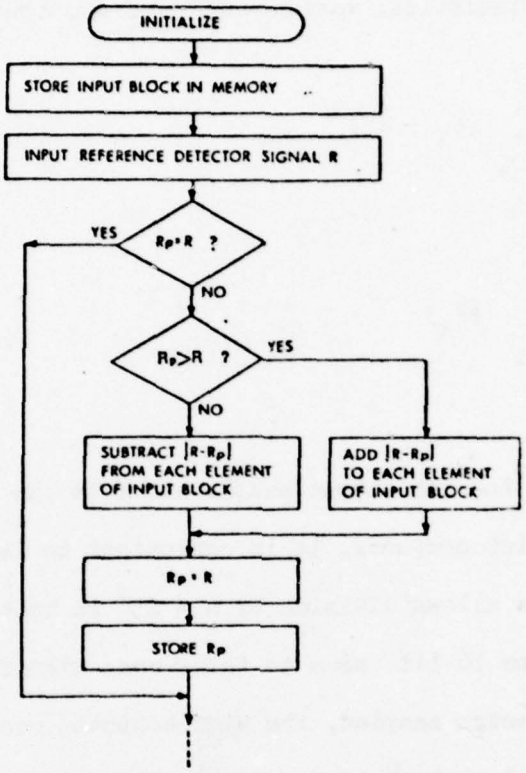


Fig. (5-4) Functional flowchart for correlated detector noise cancellation algorithm,(Courtesy of Computer Design, July, 1978)

A signal detector of a photodetector array can be used to obtain reference detector signal (R) during scan. R_p is reference detector signal from previous scan. Addition and subtraction operations are easily implemented because Z8000 CPU uses 2's complement arithmetic.

CALCULATION OF STATISTICAL VARIANCE USING ARITHMETIC

In some optical processing systems, it is necessary to measure the fluctuating component of single photodetector output quantitatively. This requires the calculation of the statistical variance (S^2) of n output signals, where

$$S^2 = \frac{1}{n} \sum_{i=1}^{n} (X_i - \bar{X})^2 \qquad (5\text{-}3)$$

and

$$\bar{X} = \frac{1}{n} \sum_{i=1}^{n} (X_i) \qquad (5\text{-}4)$$

In Eq. (5-3), X_i is the ith output sample and $\bar{X}$ is the average of n output samples. For a 16-bit microcomputer, it is convenient to let n = 256 in Eq. (5-3) and (5-4). This allows division by n = 256 in both equations to be realized by truncating the 16-bit sums to their most significant eight bits. After receiving 256 detector samples, the microcomputer can use the signal averaging technique to calculate $\bar{X}$ in Eq. (5-4). Then, $\bar{X}$ is used in Eq. (5-3). The standard multiplication technique of add-and-shift can be used to perform the squaring operation in Eq. (5-3).

Care must be taken during the calculation of S^2 in Eq. (5-3) when truncation is used to effect division. With n = 256, Eq. (5-3) can be rewrritten in the form

$$S^2 = \frac{(X_1 - \bar{X})^2 + (X_2 - X)^2 + \ldots + (X_{256} - \bar{X})^2}{256} \qquad (5\text{-}5)$$

or

$$S^2 = \frac{(X_i - \bar{X})^2}{256} + \frac{(X_2 - \bar{X})^2}{256} + \cdots + \frac{(X_{256} - \bar{X})^2}{256} \qquad (5\text{-}6)$$

Eqs. (5-5) and (5-6) are mathematically equivalent, but Eq. (5-6) cannot be used explicitly for small signals because the squared terms may be less than eight bits. If the squared terms are divided by 256 by truncating the least significant eight bits, the resultant variance will either be too low or 0. Therefore, Eq.(5-5) must be used to calculate the variance. After doing the signal averaging, the data corresponding to one is to be stored temporarily in a register.

TWO-DIMENSIONAL DISCRETE TRANSFORM PROCESSOR

Two-dimensional transformation for the data of the pattern processed earlier, operates on N^2 sample values. In order to achieve a reasonable transmission speed some form of parallel transformation of the vector series is desirable. Where N is small, it is possible to duplicate the transform hardware so that N separate transform systems, each operating on N samples, can be operated simultaneously. This gives rise to some redundancy in hardware components and, furthermore has the disadvantage that the transform operation cannot begin until the entire picture has been scanned and digitized.

The two-dimensional finite discrete transform of a two-dimensional array X_{ij} of N^2 points is given by

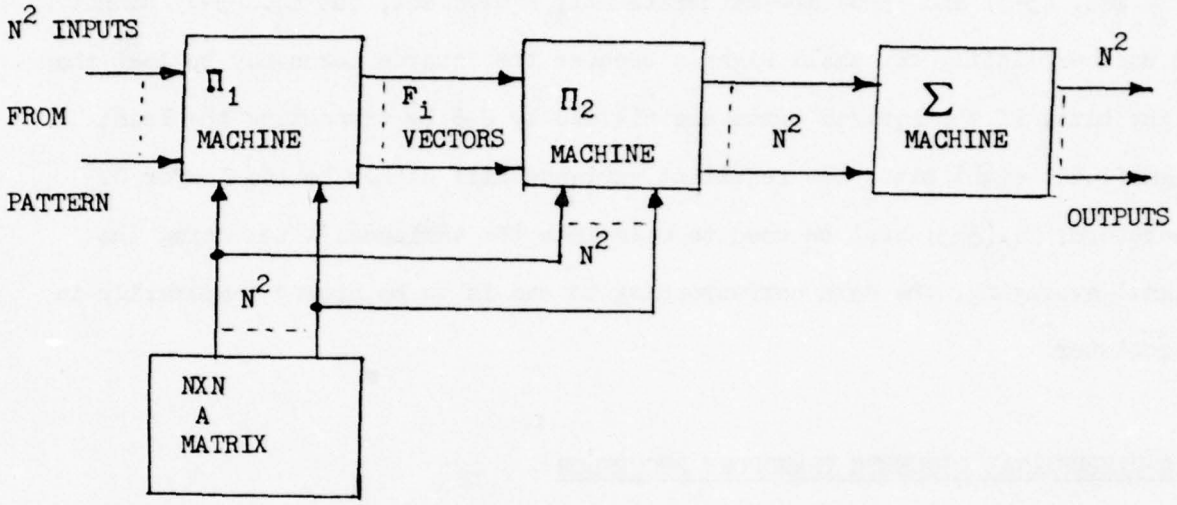


Fig. (5-5) Machine for Parallel A Transform

$$F_{NN} = \frac{1}{N^2} \sum_{u=0}^{N-1} \sum_{v=0}^{1\,N-1} f(u,v)\, A(N,V) \qquad (5\text{-}7)$$

where A(N,N) may be a WALSH/HAAR/SLANT/DISCRETE COSINE function in Discrete form.

The decomposition of Eq. (5-7) into rows and columns requires the complete matrix f(u,v) to be available.

An alternative decomposition theorm is suggested (2) based on the matrix representation for the two-dimensional transform given by the equation

$$F_{NN} = A. \quad f_{NN} \cdot A' \tag{5-8}$$

where A represents

an N X N discrete functions WALSH/HAAR/SLANT/DISCRETE COSINE and f_{NN} represents N X N set of data values. If $a_i = (a_{i1}, a_{i2}, \ldots\ldots a_{iN})$ is the ith row vector of the matrix A and we define a column vector, F_i to represent the product vector

$$F_i = \begin{bmatrix} F_{i0} \\ F_{i1} \\ F_{iN-1} \end{bmatrix} = f_i \cdot A' \tag{5-9}$$

where f_i is the ith row of data f_{NN} as $(f_{i0} \quad f_{i1} \quad f_{i2} \ldots\ldots f_{iN-1})$ then the transformation of f_{NN} will be given by the sum of the products $A \cdot F_i$.

$$F_{NN} = \sum_{i=0}^{N-1} A \cdot F_i \tag{5-10}$$

Thus, the process of two-dimensional transformation can proceed by implementing three distinct steps.

(1) Each pattern line which has been scanned and digitized is multiplied by the discrete function component to form a partial transformation, F_i .

(2) The products are retained in an accumulating array of N elements until N such products have been stored (i.e. one scan line has been processed). The vector products A . F_i are thus obtained and stored.

(3) Finally the elements of A.F_i are added to give the two-dimensional transformation of the pattern when the scanning is completed.

The important feature to note about this transformation is that only one line is operated on, at a time so that immediately step (1) is carried out and the partial transformed vector F_i, becomes available, it can be copied into a store for step (2) leaving the store for step (1) vacant to hold the results of the next line calculation. The delay in processing is thus reduced from N^2 elements to N elements, constituting one line of the pattern.

The following machine elaborates the implementation of the transform processing the digitized data.

The N^2 pattern samples are read into N^2 holding circuits. The values of the N^2 elements of the A matrix are permanently connected to the machine. Each row of the pattern is used separately and all rows are processed simultaneously in parallel. The N X N matrix generated from processing one row is a partial discrete transform of the input pattern analyzed. The N partial transforms are added together in parallel to give the N X N discrete transformed matrix. Fig.-(5-5) is a block diagram of the machine that implements parallel WALSH/HADAMARD/ HAAR/SLANT/ or DISCRETE COSINE TRANSFORM.

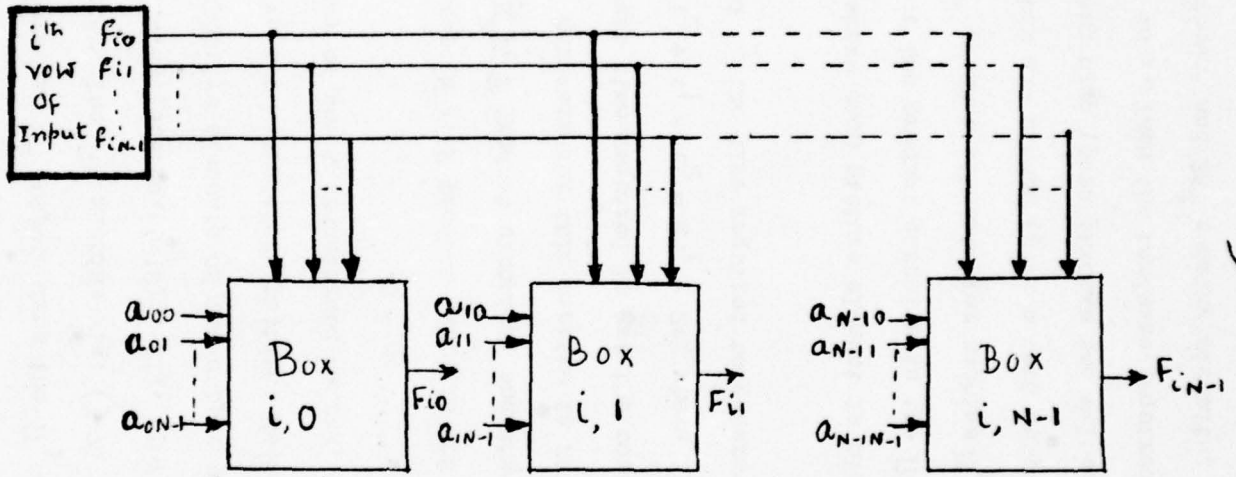
Input to the Π_1 machine are the N^2 pattern elements and the N^2 elements

of the A matrix. The outputs of the Π_1 machine are the N vectors F_i (each has N components). Fig. (5-6) shows the implementation of the Π_1 machine [see Eqs. (5-9) and (5-10)]. There are N^2 outputs from the Π_1 machine, each one is one component of an F_i vector. The Π_2 machine Fig. (5-7) performs the operations $A \cdot F_i$ for $i = 1,2,\ldots, N$ [see Eq.(5-10)]. Fig. (5-7) shows the Σ machine which adds the respective elements of the output partial transforms from the Π_2 machine to give the A transform pattern. Each F(i,j) output of the Σ machine is the ijth entry of the A transformed matrix.

The Π_1 machine requires (N^2). N product circuits and N^2 summing circuits. The Π_2 machine requires (N^2). N product circuits, and the Σ machine requires N^2 summing circuits. Total hardware for this machine is $2N^2$ product and $2N^2$ summing circuits. The longest time required to get the complete A transform of any pattern would be $t_1 + t_2 + t_1 + t_2 = 2t_1 + 2t_2$ where t_1 is the time required to generate the product and t_2 the time required to generate the sum.

For binary pictures (where each picture element is black or white), one of the two inputs to each of the product circuits in the Π_1 machine is a binary 0 or 1. Thus, the N^2 product circuits for machine Π_1 would be inexpensive electronic switches that are closed (give a zero output) when the input from the pattern is zero, and open (pass through the values of ± 1) only when the input from the picture is 1. Then the hardware required for this machine would be N^3 switches, N^3 product and $2N^2$ summing circuits, and the longest time required to obtain the complete A transform would be $t_0 + 2t_2 + t_1$ where t_1 and t_2 are as before and t_0 is the time required to set the switches of machine Π_1.

The inputs to the machine from the A matrix can be permanently connected and each pattern element corresponds to one input of each of the N product cir-



Where Box i,k : $\sum_{l=0}^{N-1} a_{kl} f_{il}$

Fig. (5-6) The Π_1 Machine

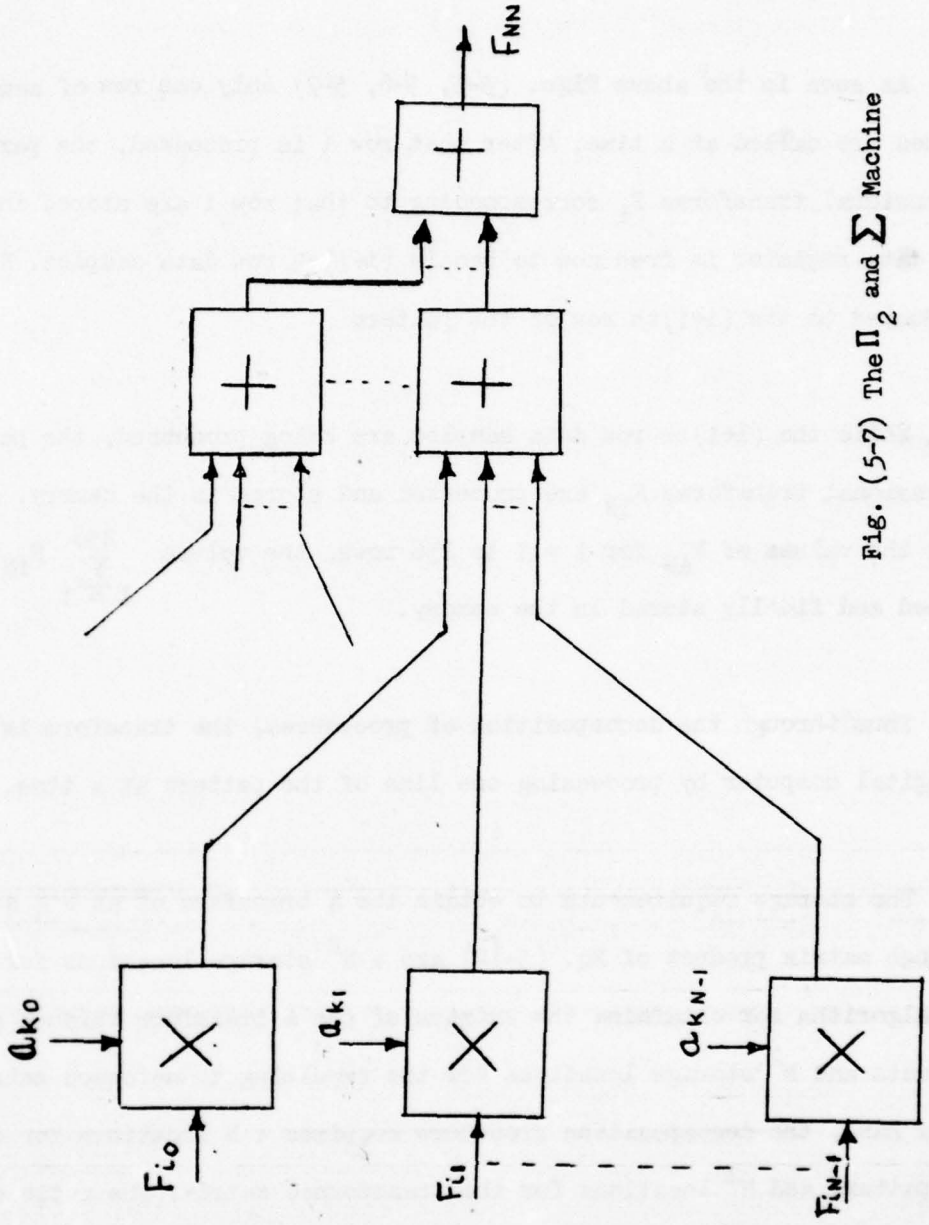


Fig. (5-7) The Π_2 and $\sum$ Machine

cuits.

As seen in the above Figs. (5-5, 5-6, 5-7) only one row of sampled digital values are called at a time. After that row i is processed, the partial one dimensional transforms F_i corresponding to that row i are stored in the memory. The data register is free now to handle (i+1)th row data samples. Now the scan is jumped to the (i+1)th row of the pattern

While the (i+1)th row data samples are being processed, the partial two-dimensional transforms F_{iN} are processed and stored in the memory. After computing the values of F_{iN} for i = 1 to 256 rows, the values $\sum_{i=1}^{256} F_{iN}$ are to be summed and finally stored in the memory.

Thus through the decomposition of procedures, the transform is obtained in a digital computer by processing one line of the pattern at a time.

The storage requirements to obtain the A transform of an N X N pattern through matrix product of Eq. (5-10) are : N^2 storage locations for the pattern, the algorithm for obtaining the entries of the A transform through sub matrices products and N^2 storage locations for the resulting transformed matrix. On the other hand, the decomposition procedure requires : N locations for one line of the pattern and N^2 locations for the transformed matrix. The ratio of the storage requirements is given by

$$\frac{2N^2}{N + N^2} = \frac{2}{1 + 1/N}$$

which results in approximately 2:1 storage reduction as the dimensionality of the picture increases as in our case.

MEMORY REQUIREMENT(3):

A total of $N + N^2$ memory locations are required for transform operations and N^2 for output storage. For carrying out the transform operation in WALSH/HADAMARD/HAAR/SLANT/DISCRETE COSINE TRANSFORM Domains for the collected data, the total memory requirement is $(N + 8N^2)$ in addition to the N^2 memory locations required for the signal averaging alone.

The total minimum memory requirement is $9N^2 + N$. For N = 256, this demands a minimum of 591 kilobits of storage locations. About 64k X 16 ROM memory is required for storing WALSH/HADAMARD/HAAR/SLANT/DISCRETE cosine Matrices coefficients for transform operations.

An additional 32k X 16 ROM memory is required for general programming data base management and general I/O control options. The total ROM memory requirement is 96k X 16.

The 32k X 8 EPROM chips are available from TEXAS Instruments in the market at a competetive price. A total of 6 chips will give a total 96k X 16 EPROM capacity at a price of $ 800.

The 32k X 8 EPROM chip TM2532 has an access time of 450n second and is operable with + 5 volts power supply.

The transformed data has to be stored properly in RAM. The storage re-

quirements for the WALSH/HADAMARD/HAAR/SLANT/DISCRETE COSINE TRANSFORM coefficients is 48k X 16 RAM memory. The general programming and data base management for I/O terminals requires an additional 16k X 16 RAM memory. The total RAM memory requirement is 64k X 16 memory. This memory is available by Bulk Bubble memories by Rockwell International or Texas Instruments at a competetive price of < $ 1000. The following Fig (5-8) shows the general structure of origanization of the memory, I/O control, Direct Memory Access and output terminal system.

The communication between various I/O terminals can be achieved through 8288 bus controller, 8259A interrupt controller and 8089 Synchronous/Asynchronous controller as shown below.

APPROXIMATE ESTIMATE OF THE PROCESSING TIME OF THE 2 DIMENSIONAL TRANSFORMS :

There are 32 Multipliers of type MPY 16HJ (16 bits by 16 bits) by TRW for the parallel processing of partial transform of the ith row. Each Multiplier takes 80nsec. for computation. The total time that is required for Multiplier is 2.56μsec.

There are 128 Adders of Type Am 25 LS 15 (4 bits by 4 bits) of AMD for parellel processing of partial transformation of ith row. Each Adder takes 25nsec. The total time that is required for adding is 3.2μsec.

This means for processing the partial transforms corresponding to the ith row alone is 3.2 + 2.56 = 5.76μsec. There are 256 rows. Hence the entire computation time for a two-dimensional transform is 1.475msec.

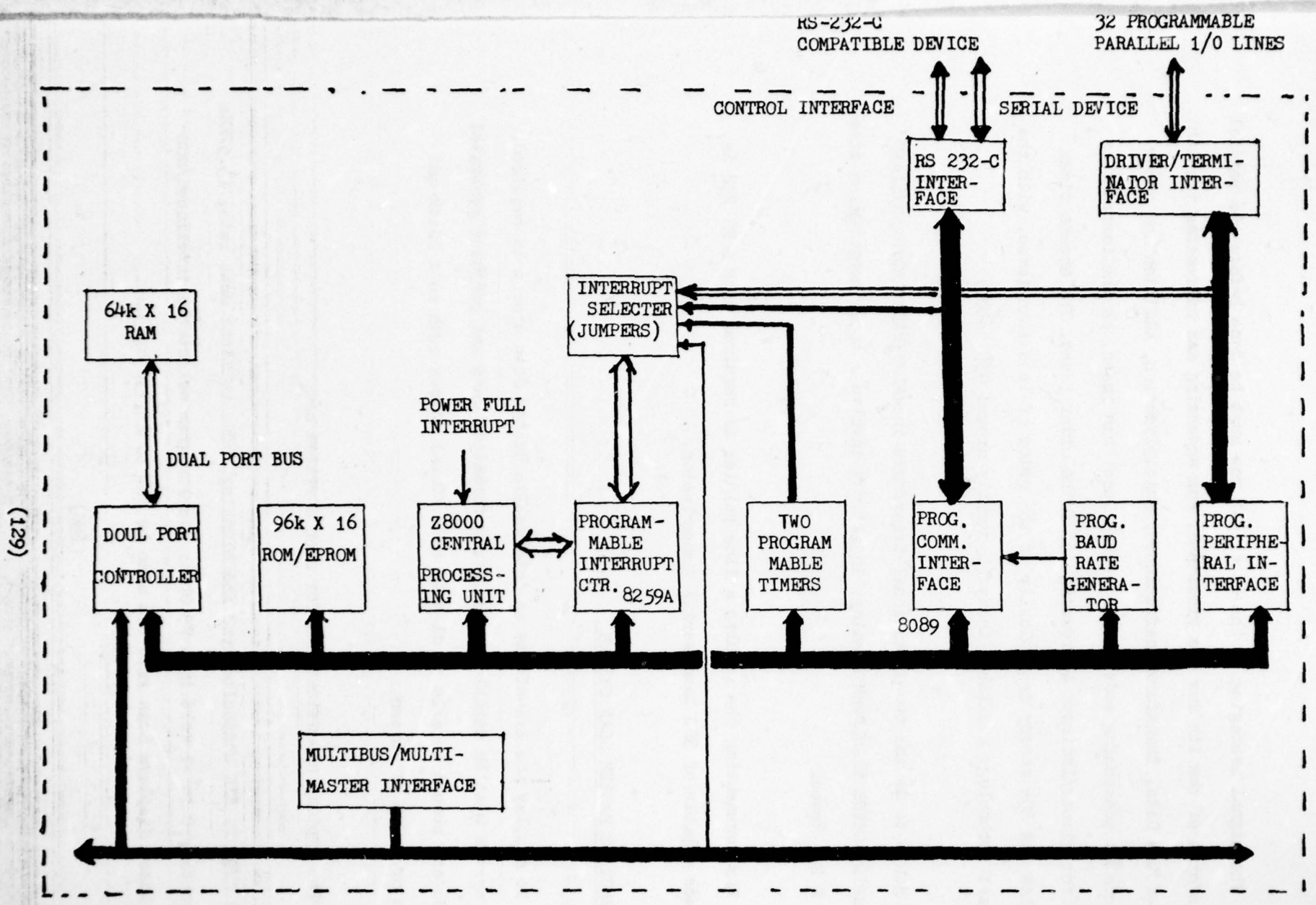


Fig. (5-8) I/0 Arrangement of Z8000.

The signal averaging of the (i + 1)th row must be done before the partial transforms of the ith row are processed. For accessing and addressing the ith row of the Data, Transform Coefficients, Multiplication, Addition and other associated addressing modes will involve less than 1msec. as the instruction time for Direct/Indirect Addressing is of the order 1msec. The Access times for RAM and ROM memory is typically of the order of less than 1msec. with the lateset technology available today (typically around 450 nsec.).

Hence to do the two-dimensional transforms in WALSH/HADAMARD/HAAR/SLANT/ DISCRETE COSINE TRANSFORM domains, in all may involve a total computation time of 4 X 2 = 8msec.

For documenting the results a line printer is required with a RS 232 in terface capable of 300 baud serial transmission.

INTERACTIVE RASTER SCAN SYSTEM

To display the transforms an interactive Raster Scan system is required. This system must be capable of accepting formatted data and computer generated images and should be able to display on a Visual screen such as a Black and White or color TV Screen.

The typical specifications for such a system are:

VIDEO OUTPUT: EIA composite sync and blanking, 60Hz vertical scan rate; 15.57KHz horizontal; 0 to -1 Volt into 75 Ohms: microprogram selectable interlaced/non-interlaced displays Scan rate can also be set to 25, 30 or 50Hz.

ALPHANUMERIC CHARACTER GENERATION: 8 X 8 Dot

MATRIX: 512 X 512: 64 lines at 85 characters

Standard 64 character ASCII upper/lower case font.

CURSOR: Scroll/Zoom/Winking

GRAYSCALE: 16 Gamma-Corrected levels from 256 level look-up table

COLOR: 16 preassigned colors to a standard RGB Color Monitor

COLOR LOOK-UP TABLE: Optional look-up table; maps 4 intensity bits to three 4-bit video levels for standard RGB Color Monitor

Up to 16 out of 4096 colors may appear at one time .

IMAGE MEMORY: Up to 32K bytes of MOS dynamic memory. Various memory segmentations are possible under software control

STANDARD CONFIGURATIONS

	Pixels	Bit(s)
B & W monochrome	1024 X 1024	X1
16-level gray or color	521 X 521	X4

DATA UPDATE: Either over DMA or Command I/O

INPUT DEVICE OPTIONS: Joystick, Trackball, Keyboard and Light-Pen

PIXEL UPDATE TIMES: Based on average times to update Image memory from new data in input buffer

16 pixel update:

Simultaneously:

Sequential Access:

45 nsec/monochrome pixel

140 nsec/grayscale pixel

Random Access:

90 nsec/monochrome pixel

190 nsec/grayscale pixel

Single pixel:	Random Access:
Update Mode:	800 nsec/monochrome pixel
	2 μ sec/grayscale pixel
Changeover:	1.2 μ sec

COMPATIBILITY: Z 8000 Bus

DATA TRANSFER RATE: Better than 2 Megabites per second from host computer

The Fig. (5-9) shows the system configuration for a typical Raster Display Unit meeting with the above specifications. The following are some of the manufactures supplying Raster scan Display systems:

(1) Lexidata corp. 215 Middle Sex Turnpike, Burlington, Massachussetts - 01803, Tel: (617) - 273-2700

(2) Matrox Electronic Systems Ltd., 2795 Bates Rd, Montreal, Quebec, H3s 1B5, Canada

(3)R Ramtek Corporation, 585 N. Mary Ave, Sunnevale, California 94085 Tel: (408) - 735 - 8400

(4) Electronic Image Systems, 2800 Indian Ripple Rd, Dayton, Ohio 45440, Tel:: (513) - 426 - 6000

(5) Grinnel Systems, 2986 Scoll Boulevard, Santa Clara, California 95050, Tel: (408) - 988 - 2100.

(6) Comtal Image Processing Systems, P.O Box 5087, Pasadana, California 91107 Tel: (213) - 793 - 2134.

(7) Megatek Corp. 3931 Sorrento Valley Blvd. San Diego, California 92121 Tel: (714) - 455 - 5590.

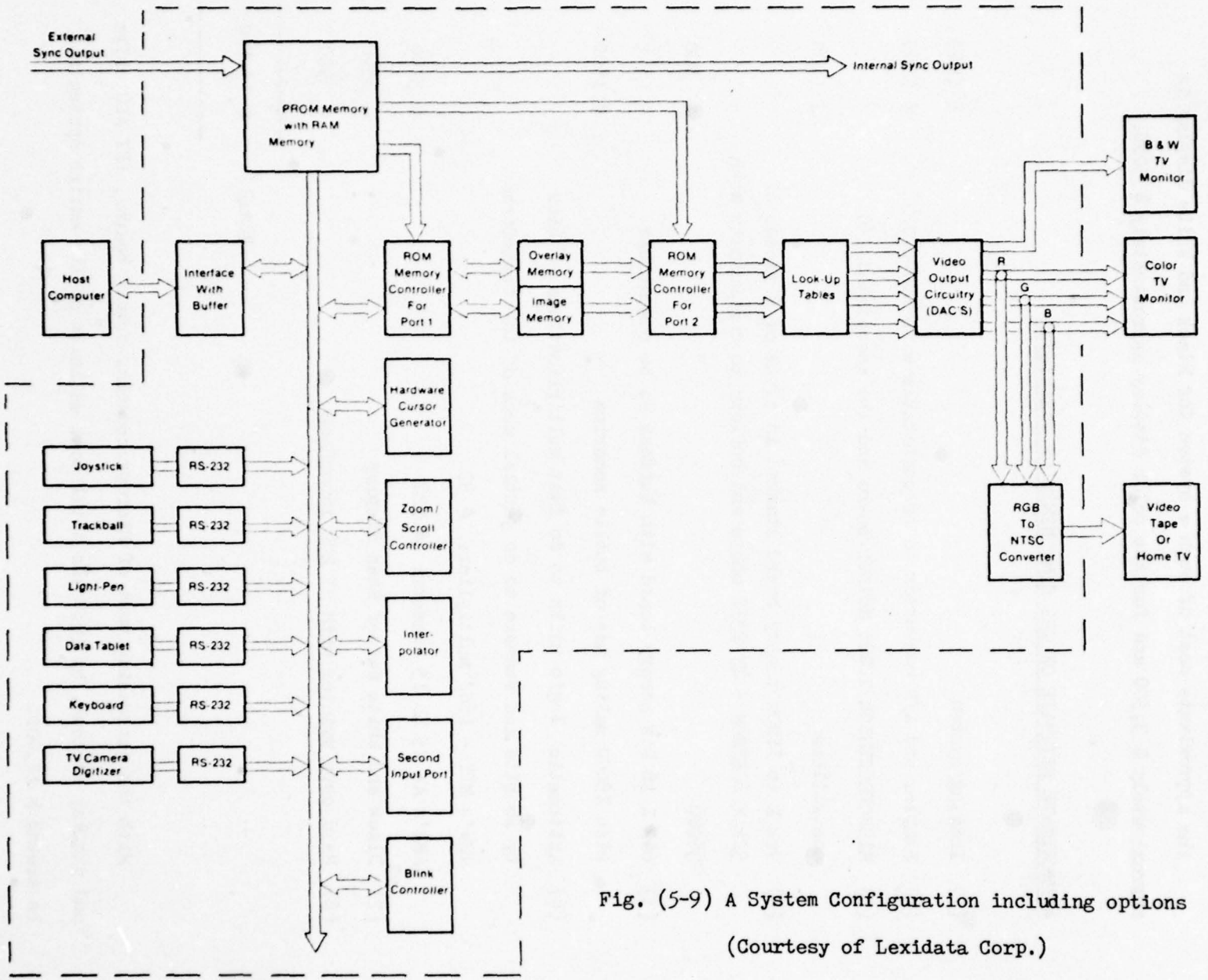


Fig. (5-9) A System Configuration including options

(Courtesy of Lexidata Corp.)

The approximate cost of such a system for Black and White display is approximately $ 3,500 and for the color display approximately $ 5000.

APPROXIMATE ESTIMATE OF THE COMPONENTS OF THE SYSTEM

(1)	Imaging system	$ 1000
(2)	Sampler and A/D converter to be compatable with Z8000 CPU	$ 1200
(3)	MICROPROCESSOR Z8000 Mother board and the associated I/O controllers	$ 1000
(4)	96k X 16 EPROM memory board stacked in terms of 6 pieces of 32k X 8 EPROM - TM 2532 chips and buffers to communicate with Z8000	$ 800
(5)	64k X 16 RAM memory board with buffers to be compatible with Z8000 making use of bubble memories	$ 1000
(6)	Arithmetic logic units to do fast multiplications 16 bits by 16 bits and summers to do partial sums of the transforms TRW's MPY - 16HJ multiplier: # 32 AMD's Am 25 LS 15 Summers: # 128	$ 5700
(7)	Black and White Raster Scan display	$ 3500
(8)	Hard copy printer with RS 232 interface	$ 900
	Total	$ 15,100

With the decreasing cost of Microprocessors, memory boards, LSI ALU chips and display systems, by 1980 the total cost estimate of the entire system may be around $ 12,000.

REFERENCES

(1) Providing software flexibility for optical processor Noise analysis: RICHARD G. LYONS, Computer Design ——— July 1978 pp 89 - 96.

(2) Real-time Walsh-Hadamard transformation: ALEXANDRIDIS, N.A. and KLINGER. A., I.E.E.E.Trans. Comp. C21, 1972 pp 288 - 92.

(3) Z-8000 Microprocessor: Data sheets and application notes: 1979, ZILOG Corp., 10460 Bubb Rd, California 95014.

CHAPTER VI

CONCLUSIONS AND RECOMMENDATIONS

CONCLUSIONS

In this report, we tried to make an investigation of linear transformations which may prove useful for automatic cartographic analysis and feature extraction. For feature extraction in the above sense, we have to do two things, (1) select and classify the standard features usually present in photographs, and (2) select analytically the transform or transforms which would appear to have the most success in enhancing the identification of the objects to be detected.

The selection and classification of cartographic features was done in chapter one. This classification was based on catogorizing the different features into three main groups, point features, line features and area features and is repeated in table (6-I). The classification of cartographic features can also be done according to some other classifiers. Table (6-II) groups the features into two categories. In the first column, the natural features and in the second column those features which are man-made are shown. Man-made features usually have some neat geometrical shape such as straightness (e.g., power lines) parallelism (e.g., dual highways), periodicity (e.g., grounded aircrafts), etc. and even sometimes there can be some a priori information about their structure (e.g.,highway clover leaf). Natural features are usually of irregular shape and their geometry cannot be predicted. Table (6-III) categorizes the cartographic features according to their regularity or irregularity in shape.

The second step of transforms technique in feature extraction is to choose the most suitable transform, if any, for bringing out or enhancing a specific feature. To do this, we must have a good knowledge about the analytical properties of different transforms. In chapter two and three, different continuous and

CARTOGRAPHIC FEATURES		
Point features	Line features	Area features
Isolated buildings	Rivers and streams with water	Large rivers
Storage tanks	Drainage channels without water	Lakes
Quarry or borrow pit	Canals	Wash
Tunnel entrance	Dual highways	Forest
	Primary Roads	Scrub
	Unpaved roads and trails	Marsh and Swamp
	Transmission lines	Mangrove
	Pipe lines	Orchard and vineyard
	Levees	Urban area
	Dam	Suburban area
	Rapids and falls	Industrial area
	Bridges	Railroad yard
	Shoreline	Cemetery
	Airport	

Table (6 - I) : Categorization of Cartographic features according to their geometry.

CARTOGRAPHIC FEATURES	
Natural	Man-Made
Rivers and streams	Isolated buildings
Unpaved roads and trails	Storage tanks
Rapids and falls	Quarry or borrow pit
Shoreline	Tunnel entrance
Large rivers	Canals
Lakes	Dual Highways
Forests	Primary roads
Scrub	Secondary roads
Marsh and swamp	Transmission lines
	Pipe lines
	Levees
	Dam
	Bridges
	Airport
	Orchard and vineyard
	Urban area
	Suburban area
	Industrial area
	Railroad area
	Cemetery

Table (6-II): Categorization of cartographic features according to their nature.

CARTOGRAPHIC FEATURES	
Neat Geometrical shape	Irregular Shape
Isolated buildings	Rivers and Streams
Storage tanks	Unpaved road and trails
Tunnel entrance	Rapids and falls
Canals	Shoreline
Dual highways	Large rivers
Primary roads	Lakes
Transmission lines	Scrub
Pipe lines	Marsh and swamp
Dam	Quarry or borrow pit
Bridges	Secondary roads
Airport	Levees
Orchard and vineyard	Drainage channel
Urban area	Suburban area
Railroad yard	Industrial area
Cemetery	

Table (6-III) : Categorization of cartographic features according to their regularity in shape.

discrete transforms were studied. Among continuous transforms, Fourier and Hankel transform, and among discrete transforms, Hadamard, Haar, Slant and Discrete Cosine transform were selected. Two-dimensional matched filtering and also some new extrapolation techniques were discussed in the context of continuous transforms .

Another important method in feature extraction, namely line and curve detection was not discussed in this report because it was not in the scope of our transforms study. It can be found in the literature under picture processing techniques. Table (6-IV) gives a list of different techniques for feature extraction.

In studying different transforms, emphasis was put on two things :

(a) How a particular transform behaves as the input object image is translated, rotated or scaled. This is a very useful consideration because usually there is no a priori information about the location, orientation or size of an object image in a picture. Morever, since the size of an object image depends on the distance from which a picture has been taken, it will be advantageous to know the response of a transform to the scaled versions of an object. In Table (6-V) we have summarized the translational,rotational and size invariance properties of the transforms studied.

(b) How powerful is a particular transform in bringing out a specific feature ? To answer this question some candidate features were selected and different transforms were applied to those features. Based on these examples and also based on analytical properties of the transformations, some general remarks can be made. It should be noted that all our conclusions are based on the observa-

Two-dimensional linear transforms	Other techniques
Fourier	Matched filtering
Hankel	
Hadamard	Extrapolation techniques
Haar	
Slant	Line and curve detection *
Discrete cosine	

Table (6-IV) : A list of different techniques for feature extraction

* Not discussed in this report

Techniques	Translation Invariant	Rotation Invariant	Size Invariant
Fourier transform	O.K. (spectrum)	The spectrum rotates	The spectrum is scaled
Hankel transform	O.K	O.K	The spectrum is scaled
Hadamard transform	NO	NO	NO
Haar transform	NO	NO	NO
Slant transform	NO	NO	NO
Discrete Cosine transform	NO	NO	NO
Matched filtering	O.K	Can be modified to do so	Can be modified to do so
Extrapolation techniques	O.K	O.K	O.K

Table (6-V) : The behaviour of different feature extraction techniques.

tions we have made and so they will be judgements to best of our knowledge. We do not guarantee that these conclusions are the most accurate ones because we have not made an exhaustive search for all cartographic features. This ia a caveat.

Continuous transformations like Fourier and Hankel transform give a very neat analytical result when applied to the selected features.

The Fourier transform, because of the nature of its kernel, appears to be the most powerful transform in bringing out periodic structures such as parallel streets.

Hankel transform seems very useful in bringing out features with circular symmetry such as storage containers.

The two-dinensional matched filtering, seems the most powerful means of feature extraction provided some a priori information is known about the shape of the feature to be detected (e.g. airport) . If we know exactly what we are looking for in a picture, we can build a suitable matched filter for detecting that particular feature. We saw that the matched filter can be modified to detect scaled versions or rotated versions of an object. Conceptually matched filtering can be performed easily using a digital computer.

In the case of discrete transforms, it was observed that in almost all of the cases of the cartographic features, the Walsh Hadamard transforms have a decided advantage over the other discrete transforms. Only in a couple of cases, are Slant transforms promising when compared to the others. Even the Slant edges may be detected by WHTs if proper threshold filtering or zonal filtering techni-

ques are adopted.

In addition, the WHT offers super fast computational flexibility when compared to the whole set of the other transforms either discrete or continuous. Such fast WHT algorithms are available extensively in the literature. Besides in real time applications, the hardware realization can be done with minimum storage requirements and investment. Such fast WHT processors are already available in the market .

Another subject which was discussed in this report, relates to the extrapolation of bandlimited signals. We proposed that extrapolation techniques can be used to enhance a part of a picture degraded by noise. This idea may seem a little ambitious, but we believe that this technique may be used to bring out features in one part of a picture, from the information conveyed by the rest of that picture. More work has to be done on this subject.

RECOMMENDATIONS

The main purpose of the work done on this report has been to study picture transformations, to illustrate their analytical properties and to compare different characteristics of various transforms. However, thoughout this paper we tried to illustrate which picture transformations are the most suitable ones for bringing out a specific cartographic feature. The selection of condidate features, using examples and applying different transforms to those examples all have been in the scope of accomplishing the above task. In the first section of this chapter, we made some conclusions based on our studies. It was pointed out that those conclusions are best to our knowledge and cannot be taken as the ultimate words.

In this part, we like to give some recommendations about using different transforms in extracting different cartographic features. This task is not simple and our judgements may look somehow ambiguous at different points. However, we do not claim that these recommendations are the most accurate ones. They are based on our observations and our observations could not be exhaustive. Much more study is needed on this subject to bring more plausible results.

In Table (6-VI), a general recommendation is given on the use of different feature extraction techniques. Our judgement for this recommendation is based on:

(a) since all picture transforms have some well defined mathematical base functions, or kernels, they are most suitable for extracting features with neat geometrical shape. For example, the Hankel transform with the Bessel function in its kernel is very powerful in bringing out circular object images.

(b) when a man-made cartographic feature has an irregular shape, using image transforms will not be much effective. However there is usually an a priori information about the shape or geometry of that object. A highway clover leaf is an example of this kind. If such information is available, then a replica of that object image can be used to build a matched filter for detecting the object.

(c) Cartographic features which are usually observed in nature, such as rivers, caves, etc., neither have a neat geometrical shape nor is there any a priori information about their structure. The use of picture transforms or matched filter do not seem very useful in this case. It appears that line and curve detection techniques are the best methods to enhance them.

Cartographic features	Neat geometrical shape	Irregular shape
Man-Made	Use Linear Transformation	Use Matched Filtering
Natural		Use Line and Curve detection *

Table (6-VI) : A general recommendation for the use of different techniques in extracting different patterns.

* Not discussed in this report.

In Table (6-VII), we have given detailed recommendations for the use of different techniques in extracting various cartographic features. For each cartographic feature, we have recommended three different transforms with three different ratings. Again we mention that these recommendations are based only on our observation and in no way should be considered as the ultimate truth.

Our judgements in making up Table (6-VII) are based on :

(1) The examples used in chapter two and three and the computer results in chapter four.

(2) The general recommendation given by Table (6-VI).

(3) The power of Fourier transform in bringing out periodic structures.

(4) The power of Hankel transform in bringing out images with circular symmetry.

(5) The power of Hadamard transform in bringing out lines.

(6) The power of Slant transform in bringing out shaded edges.

(7) The power of Matched filters in bringing out images with known shapes.

(8) The power of Line and curve detection techniques in bringing out curvatures.

(9) In the case of any tie between using a transform or the matched filter, preference is given to the transform because of its simpler applicability.

(10) In the case of any tie between using a transform or curve detection techniques preference is given to the transform because this paper is devoted to transforms study.

(11) In the case of any tie between two transforms, preference has been given to the transform with more computational simplicity.

Techniques to be used / Cartographic features	Fourier trans	Hankel trans.	Hadamard trans	Haar trans.	Slant trans.	Discrete Cosine trans	Matched filtering	Line and curve detection
Isolated buildings	B		?	?				
Storage tanks		A					B	C
Tunnel entrance	?						A	B
Canals	B		A					C
Dual highways	A		C					B
Primary roads	B		C					A
Secondary roads			B		?			A
Transmission lines	A		C					B
Pipe lines	C		B					A
Levees			B		C			A
Dam			B				A	C
Rivers and streams			B	C				A
Rapids and falls			B	C				A
Shore line	B				?			A
Lakes	C		B					A
Bridges	B		A					C
Airport	B						A	C
Urban area	A		?					B
Railroad yard	A		B				B	C
Forests	A		B				C	
Industrial area	C						A	B
Cemetery	A						B	C
Orchard	A		B		C			

Table (6-VII): A detailed recommendation for the use of different techniques in extracting different patterns.

A = Good B = Satisfactory C = Fair

RECOMMENDATIONS FOR FUTURE STUDY

The ultimate objective of realizing a completely automatic and computerized picture reading machine for cartographic analysis is a long way off. More likely to be realized within the decade is a hybrid man-machine device which would use feature extraction and object identification to aid cartographic readers to reduce the tedium and increase productivity and accuracy. It is therefore necessary to focus further research effort at those problems which are likely to achieve success in this objective. It appears that at this time, the approach would be to create a system which does cartographic analysis as follows:

FUNCTION	SYSTEM
(a) Selection of general category, man-made or natural	Human
(b) Based on selection (a) apply appropriate signal processing algorithms (Transforms, edge detection, etc)	Computer
(c) Based on results of (b) interactively adjust parameters	Human-Computer interaction
(d) When human is satisfied, select features of highest significance	Computer
(e) Present cartographic objects of interest in sharp contrast and give confidence levels	Computer
(f) Examine more carefully those objects with low confidence level	Human

(g) Repeat (a) for another category Human

To accomplish this objective, it will be necessary not only to further explore signal processing techniques and the technalogy to support it, but also to develop adequate decision rules specifically for discriminating cartographic objects of interest. This will require study of :

(1) Algorithms of clustering and recognition of cartographically interesting objects. This will include more refined feature definition and detection.
(2) Statistical analysis to determine confidence levels.
(3) The generation of software and possibly a specialized high-level language.
(4) Investigation of specialized hardware signal processes for increased speed.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] Mc Glamery, B.L. : Digital Image Processing. Summary Report, Visibility Laboratory, Univ. of Californis, San Diego 1972.

[2] Goodman, J.W. : Introduction to Fourier-Optics. New York : Mc Graw-Hill 1968.

[3] Falconer, D.G. : Image Enhancement and Film-Grain Noise. Optica Acta 17 (1970), 693-705.

[4] Stockham, T.G. : Image Processing in the Context of a Visual Model. Proc. IEEE 60 (1972), 828-842.

(5] Sawchuk, A. A. : Space-Variant Motion Degradation and Restoration. Proc. IEEE 60 (1972), 854-861.

[6] Robbins, G.M., Huang, T.S. : Inverse Filtering for Linear Shift-Variant Imaging Systems. Proc. IEEE 60 (1972), 862-872.

[7] Rosenfeld, A. : Picture Processing by Computer. Academic Press, New York - London 1969.

[8] Cooley, J.W., Tuckey, J.W. : An Algorithm for the Machine Calculation of Complex Fourier Series. Math. of Comp. 19 (1965), 297-301.

[9] Cooley, J.W., Lewis, P.W., Welch, P.D. : The Fast Fourier Transform and its Applications. IEEE Trans. Educ. 12 (1969), 27-34.

[10] Bracewell, R. : The Fourier Transform and its Applications. Mc Graw-Hill, New York 1965.

[11] Barnes, G.H., R.M. Brown, M. Kato, D.J.Kuck, D.L. Slotnik : The ILLIAC IV Computer. IEEE Trans. Computers C-17 (1968), 746-757.

[12] Kuck, D.J. : ILLIAC IV Software and Application Programming. IEEE Trans. Computers C-17 (1968), 758-770.

[13] Pease, M.C. : An Adaptation of the Fast Fourier Transform for Parallel Processing. J. ACM 15 (1968), 252-264.

[14] Bergland, G.D. : Fast Fourier Transform Hardware Implementations- a Survey. IEEE Trans. Audio and Electroacoustics 17 (1969), 109-119.

[15] Johanston, Emily, G. : The PAX II Picture Processing System. In Lipkin, B. S., Rosenfeld, A., eds., Picture Processing and Psychopictorics. Academic Press, New York - London 1970.

[16] Courant, R., Hilbert, D. : Methods of Mathematical Physics. Interscience New York 1953.

[17] Andrews, H.C. : Computer Techniques in Image Processing. Academic Press, New York - London 1970.

[18] Davenport, W. B., Root, W.J. : Random Signals and Noise. Mc Graw-Hill, New York 1958.

[19] Harris, J.L. : Image Evaluation and Restoration. J.Opt. Soc. Amer. 56 (1966), 569-574.

[20] Roetling, P.G., Trabks, E. A., Kinzly, R.E. : Theoretical Prediction of Image Quality. J. Opt. Soc. Amer. 58 (1968), 342-346.

[21] Horner, J.L. : Optical Spatial Filtering with the Least-Mean-Square-Error Filter. J. Opt. Soc. Amer. 59 (1969), 553-558.

[22] Slepian, D. : Linear Least-Squares Filtering of Distorted Images. J.Opt. Soc. Amer. 57 (1967), 918-922.

[23] Mc Adam, D. P. : Digital Image Restoration by Constrained Deconvolution. J. Opt. Soc. Amer. 60 (1970), 1617-1627.

[24] Moik, J.G. : SMIPS-A Small Interactive Image Processing System-Users Manual. GSFC X-Document X-650-73-283.

[25] Nathan, R. : Picture Enahancement for the Moon, Mars and Man. In Cheng, G.C., et al, eds., Pictorial Pattern Recognition, Washington, Thompson, (1968), 239-266.

[26] Twomey, S. : On the Nimerical Solution of Fredholm Integral Equations of the first kind by the Inversion of the Linear System Produced by Quadrature. J. Assoc. Comp. Mach. 10 (1963), 97-101.

[27] Smith, H.A. : Improvement of the Resolution of a Linear Scanning Device. SIAM J. Appl. Math. 14 (1966), 23-40.

[28] Prewitt, Judith, M.S. : Object Enhancement and Extraction. In Lipkin, E.S. and Rosenfeld, A. , eds., Picture Processing and Psychopictorics. Academic Press, New York - London (1970), 75-149.

[29] Freeman, H. : On the Encoding of Arbitrary Geomatric Configurations. IEEE Trans. EC-10 (1961), 260-268.

[30] Montanari, U. : A note one Minimal Length Polygonal Approximation to a Digitized Contour. Comm. ACM 13 (1970), 41-47.

[31] Narasimhan, R. : Labeling Schemata and Syntactic Descriptions for Pictures. Information and Control 7 (1964), 151-179.

[32] Rosenfeld, A., Pfaltz, J.L. : Sequential Operations in Digital Picture Processing. J. ACM 13 (1966), 471-494.

[33] Andrews, H.C., Tescher, A.G., Kruger, R.P. : Image Processing by Digital Computer. IEEE Spectrum 9 (1972), 20-32.

[34] Cooley, J.W., Lewis, A. W., Welch, P.D.: Application of the Fast Fourier Transform to Computation of Fourier Integrals, Fourier Series and Convolution Integrals. IEEE Trams. Audio and Electroacoust. 15 (1967).

[35] Hunt, B.R. : A Matrix Theory Proof of the Discrete Convolution Theorem. IEEE Trans. Audio and Electroacoust. 19 (1971), 285-288.

[36] Uhr, L., Vossler, C. : A Pattern Recognition Program that Generates, Evaluates and Adjusts its own Operators. Proc. WJCC (1961), 555-569.

[37] Block, H.D., Nilson, N.J., Duda, R.O. : Determination and Detection of Features in Patterns. In Tou, J.T., Wilcox, R.H., eds., Computer and Information Sciences, New York, Spartan, (1964), 75-110.

[38] Kazmierczaj, H., Steinbuch, K. : Adaptive Systems in Pattern Recognition. IEEE Trans. EC 12 (1963), 822-835.

[39] Montgomery, W.D., Broome, P.W. : Spatial Filtering, J. Opt. Soc. Amer. 52 (1962), 1259-1275.

[40] Deutsch, E.S., Wann, Z.F. : On the Use of the Hadamard Transform in Digital Image Matching. Univ. of Maryland Techn. Rep. TR-216, (1973).

[41] Arcese, A., Menger, P.H., Trombini. ; Image Detection through Bipolar Correlation. IEEE Trans. IT-16 (19).

[42] Andrews, H.C. : Automatic Interpretation and Classification of Images by Use of the Fourier Domain. In Grasselli, A., ed., Automatic Interpretation and Classification of Images. New York, Academic Press (1968), 187-198.

[43] Duda, R.O., Hart, P.E. : Pattern Classification and Scene Analysis. Wiley-Interscience, New York, London, (1973).

[44] Andrews, H.C. : Introduction to Mathematical Techniques in Pattern Recognition. Wiley-Interscience, New York, London, (1972).

[45] Allison, L.J., Cherrix, G.T. : The Application of Color Display Techniques for the Analysis of Nimbus Infrared Radiation Data. NASA Technical Note D-6786, (1972).

[46] Vander Lugt, A. : A review of Optical Data-processing by Electron-Optical Acta 15 (1968), 1-33.

[47] Hawkins, J.K., Munsey, C.J. : Image Processing by Electron-Optical Techniques. J. Opt. Soc. Amer. 57 (1967), 914-918.

[48] Sheppard, J.J., Stratton, R. H., Gazely, C.G. : Pseudo-color as a Means of Image Enchancement. J. Optom., 46 (1969), 735-754.

[49] Su, M.Y. : The Composite Sequential Clustering Technique for Analysis of Multispectral Scanner Data. NASA Contractor Report CR-128999, (1972).

[50] Rindfleisch, T.C., et al. : Digital Processing of the Mariner 6 and 7 Pictures. J. of Geophysical Research 76 (1971), 394-417.

[51] Rosenfeld, Azriel Picture Processing by Computer Academic Press, New York, (1969).

[52] Rosenfeld, A., and A. Kak Digital Picture Processing Academic Press, New York, (1976).

[53] Wintz, P.A. " Transform Picture Coding " Proc. IEEE. Vol 60, No 7, July (1972) pp. 809-820.

[54] Rao, K.R. et. al. " Image Data Processing by Hadamard-Haar Transform " , IEEE Trans. On Computers Vol c-24-No. 9 Sept. (1975).

[55] Pratt, W.K. et. al. " Slant Transform Image Coding " IEEE Trans. On Communications, Vol. Com-22, No. 8, Aug. (1974).

[56] Billingsley, F.C. " Applied Image Processing " Applied Optics, Vol 9, No. 2 February (1970).

[57] Reed, I.S. et. al. " Image Processing by Transforms over a Finite Field " IEEE Trans. On Computers Vol c-26, No. 9. September (1977).

[58] Bajcsy, R. and M. Tavakoli " Computer Recognition of Roads from Satellite Pictures " IEEE Trans. On Systems, Man and Cybarnetics, Vol 5mc-6, No. 9, September (1976).

[59] Garvey, T.D., and J.M. Tenenbaum " Application of Interactive Scence Analysis Techniques to Cartography " Proc. US-Japan Conference on Picture Processing, August, (1977).

[60] Digital Image Restoration : H.C. Andrews and B.R. Hunt, Prentice Hall Inc, Newjersey , (1977).

[61] Orthogonal Transforms for Digital Signal Processing : N. Ahmed and K.R.Rao, Springer-Verlag New York, (1975).

[62] " Applications of Walsh Functions " Proceedings of 1970, 1971, 1972, 1973, and 1974.

[63] Cannon, T.M. (1974), " Digital Image Deblurring by Nonlinear Homomorphic Filtering, " Ph.D. Thesis, Computer Science Department, University of Utah, Salt Lake City, Utah.

[64] Colas-Baudelaire, P., (1973), " Digital-Picture-Processing and Psychophysics, A Study of Brightness Perception, " report UTEC-Cse-74-025, Computer Science, University of Utah, Salt Lake City, Utah (March).

[65] Hershel, R.S. (1971), " Unified Approach to Restoring Degraded Images in the Presence of Noise, " Technical report 71, Optical Science Center, University of Arizona, Tucson, Arizona (December).

[66] Hou, H.S. (1976), Ph.D. Dissertation, Department of Electrical Engineering, University of Southern California.

[67] Wallis, R. (1975): Ph.D. Dissertation, Department of Electrical Engineering, University of Southern California, Los Angeles, California.

[68] Frieden, B.R. (1971), " Restoring with Maximum Likelihood, " Technical Report 67, University of Arizona, Tucson, Arizona (February)

[69] Beauchamp .K.G. Walsh Functions and their Applications, (1975), Academic Press, New York.

[70] Pearson . D.E. Transmission and Display of Pictorial Information, (1975), John Wiley, New York.

[71] Murthy S.S.R and Ramakrishna P.S: Function generation by Walsh Functions, Journal of the Institution of Electronics and Telecom. Engrs., 1973, Vol 19, No 11 pp608-612.

[72] Rafael C. Gonzalez and Paul Wintz: Digital Image Processing, (1977), Addison Wesley, Reading, Massachusetts.

[73] Julius T. Tou and Rafael C. Gonzalez: Pattern Recognition Principles, (1974) Addison Wesley, Reading, Massachusetts.

[74] Pratt W.K.: Digital Image Processing, (1978), John Wiley, New York.

[75] Harmuth H : Sequency Theory, Foundations and Applications, (1977), Advances in Electronics and Electron Physics Supplement 9, Academic Press, New York.

[76] Harmuth H: Transmission of Information by Orthogonal Functions, 2nd Ed. (1972) Springer Verlag, New York.

[77] Pichler F and Quatember B: Implementation of a Dyadic Correlator, Proc. IEEE - EMC Sym. Montreux, (1975) Supplement on Sequency Techniques pp22 - 25.

[78] " Theory and Applications of Walsh Functions " Proceedings of the Walsh Function Symposiums, Hat Field Polytechnic, Hat Field, Hertford Shire, England (1971, 73, 75)

[79] Bramhall J.N: " An Annotated Bibliography on Walsh and Walsh Related Functions " John Hopkins Iniv., Applied Physics Lab., Silver Spring, Maryland (1972, 73, 74).

[80] Robert F.Rice: RM2: Transform Operations Tech Memorandum 33-680, March 1 March 1, (1974) Jet Propulsion Lab., Pasadena, California.

APPENDIX 1

SCALED VERSIONS OF A TWO-DIMENSIONAL FUNCTION

Let $f(x,y)$ be a curve in the x-y plane whose corresponding equation in polar coordinates is given by $P(\rho, \theta)$. If this curve has a closed contour, we assume that the origin of the x - y plane is located inside the contour. A scaled version of $P(\rho, \theta)$ is given by $P(k\rho, \theta)$ in which k is the scaling factor.

Since $\rho = \sqrt{x^2 + y^2}$, the scaled version of $f(x,y)$ can be obtained by putting,

$$x \longleftarrow kx \qquad \text{and} \qquad y \longleftarrow ky$$

and thus getting $f(kx,ky)$ as the scaled version of $f(x,y)$. If $k < 1$, then the curve is magnified . For $k > 1$, the reduced scaled version of the curve is obtained. The figure below shows a function and its scaled versions.

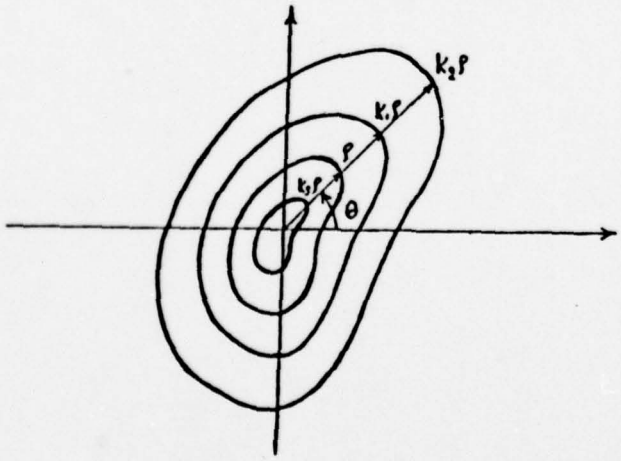


Fig (I-1) Scaled versions of a function.

APPENDIX II

PROLATE SPHEROIDAL WAVE FUNCTIONS

Prolate Spheroidal Wave Functions are a set of functions with the following properties :

(i) The set of functions is countably infinite and all the functions are real.

(ii) The $\varphi_i(t)$ are complete, bandlimited and orthonormal on the real line, i.e.,

$$\int_{-\infty}^{+\infty} \varphi_i(t,c) \quad \varphi_j(t,c)\, dt \;=\; \begin{cases} 0 & i = j \\ 1 & i \neq j \end{cases} \qquad \text{(II-1)}$$

c is a constant related to the cutoff frequency and the known interval T,

$$c = \frac{\Omega T}{2} \qquad \text{(II-2)}$$

(iii) In the interval $|t| \leq \frac{T}{2}$, the $\varphi_i(t)$ are complete and orthogonal i.e.,

$$\int_{-\frac{T}{2}}^{+\frac{T}{2}} \varphi_i(t,c)\ \varphi_j(t,c)\, dt \;=\; \begin{cases} 0 & i = j \\ \lambda_i & i \neq j \end{cases} \qquad \text{(II-3)}$$

(iv) The functions $\varphi_i(t)$ are obtained from the following integral equation,

$$\lambda_i(c)\ \varphi_i(t,c) \;=\; \int_{-\frac{T}{2}}^{+\frac{T}{2}} \frac{\operatorname{Sin} \Omega (t - s)}{\Pi(t - s)}\ \varphi_i(s,c)\, ds \qquad \text{(II-4)}$$

in which $\lambda(c)$'s are real positive numbers (eigenvalues) so that

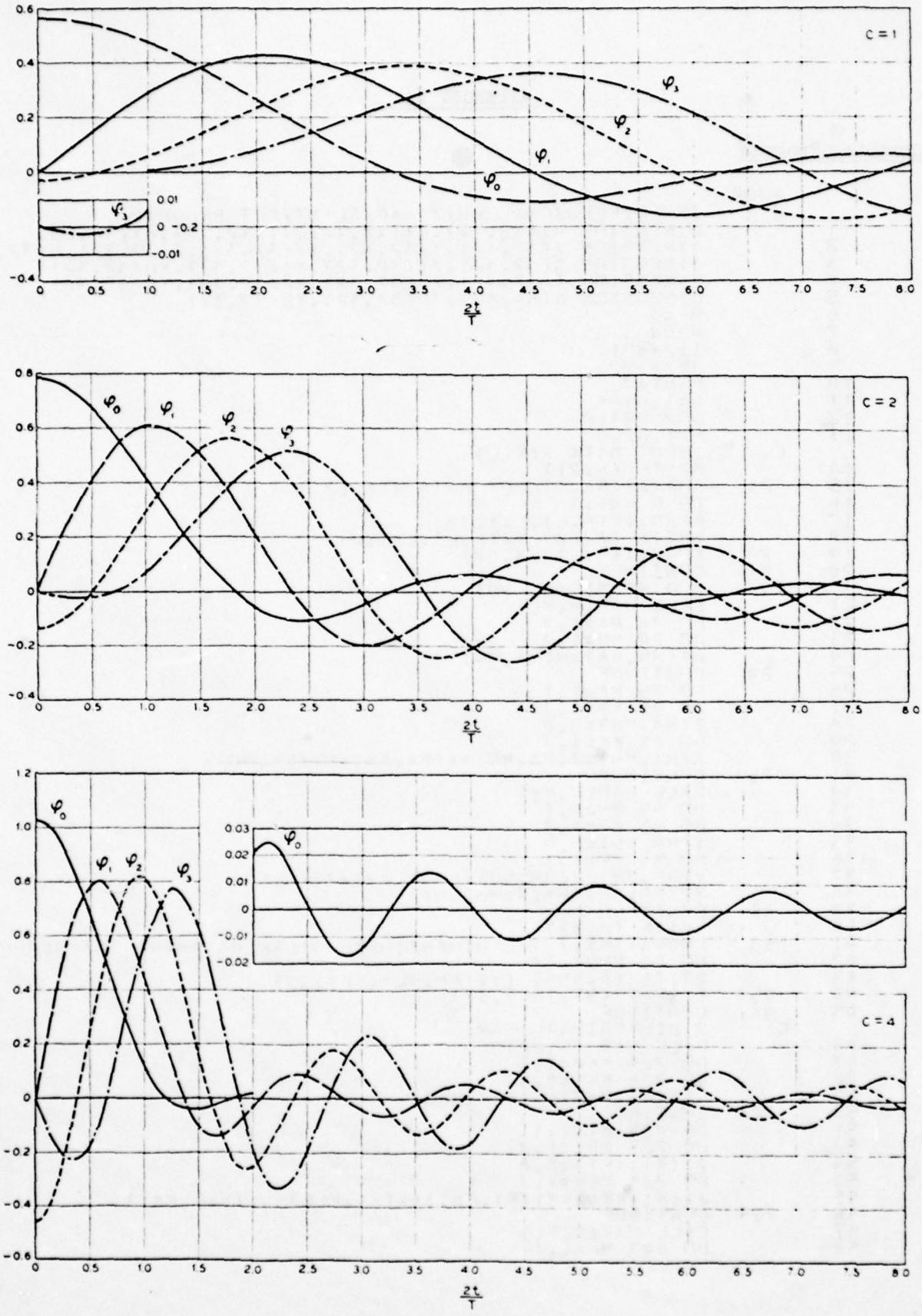
$$\lambda_0 > \lambda_1 > \lambda_2 > \cdots \qquad \text{(II-5)}$$

and each eigenvalue $\lambda_i(c)$ corresponds to an eigenfunction $\lambda_i(t,c)$.

There exists no exact analytic expression for Prolate Spheroidal functions, but these functions have been tabulated. In Fig.(II-1) the Prolate Spheroidal functions for c = 1, 2 and 4 are shown. Table (1) gives the values of $\lambda_i(c)$.

n	c = 0.5		c = 1.0		c = 2.0		c = 4.0		c = 8.0	
	L	p	L	p	L	p	L	p	L	p
0	3.0969	1	5.7258	1	8.8056	1	9.9589	1	1.0000	0
1	8.5811	3	6.2791	2	3.5564	1	9.1211	1	9.9988	1
2	3.9175	5	1.2375	3	3.5868	2	5.1905	1	9.9700	1
3	7.2114	8	9.2010	6	1.1522	3	1.1021	1	9.6055	1
4	7.2714	11	3.7179	8	1.8882	5	8.8279	3	7.4790	1
5	4.6378	14	9.4914	11	1.9359	7	3.8129	4	3.2028	1
6	2.0413	17	1.6716	13	1.3661	9	1.0951	5	6.0784	2
7	6.5766	21	2.1544	16	7.0489	12	2.2786	7	6.1263	3
8	1.6183	24	2.1207	19	2.7768	14	3.6066	9	4.1825	4

Table I - Values of $\lambda_n(C) = L_n(C) \times 10^{-p_n(c)}$



FIG(II-1)Prolate Spheroidal Wave Functions for c = 1,2 and 4.

APPENDIX III

Computer Program

```
      $JOB
      C       TWO DIMENSIONAL WHT/ HAR/SLANT/DCT PROGRAM
    1         DIMENSION H1(32,32),G1(32,32),T1(32,32),D1(64,64)
    2         DIMENSION X(32,32),Y(32,32),YY(32,32),Z(32,32),W(32,32)
    3         DIMENSION G(32,32),Z1(32,32),Y1(32,32),YH(32,32)
    4         DIMENSION T(32,32),TS(32,32),Y2(32,32),YS(32,32)
    5         DIMENSION D(64,64),ZD(32,32),YD(32,32)
    6         M1=4
    7         M2=4
    8         I=2**M1
    9         J=2**M2
   10         M5=I*J
   11         Q=1.0*M5
   12         QQ=SQRT(Q)
   13         F=1.0/QQ
      C       INPUT DATA X(K,L)
   14         WRITE (6,21)
   15      21 FORMAT(5X,'INPUT DATA X(K,L) ')
   16         DO 5 K=1,I
   17         READ,(X(K,L),L=1,J)
   18         WRITE (6,22) (X(K,L),L=1,J)
   19      22 FORMAT(5X,32F3.0/)
   20      5  CONTINUE
      C       2 DIMENSIONAL WHT
   21         CALL HAD(H,M2)
   22         DO 24 MA=1,I
   23         DO 24 MN=1,J
   24         H1(MN,MA)=H(MA,MN)
   25      24 CONTINUE
   26         DO 25 MA=1,I
   27         DO 25 MN=1,J
   28         Z(MA,MN)=0.0
   29         DO 25 KK=1,I
   30         Z(MA,MN)=Z(MA,MN)+X(MA,KK)*H1(KK,MN)
   31      25 CONTINUE
   32         CALL HAD(H,M1)
   33         DO 45 MM=1,I
   34         DO 45 NN=1,J
   35         Y(MM,NN)=0.0
   36         DO 45 KM=1,I
   37         Y(MM,NN)=Y(MM,NN)+H(MM,KM)*Z(KM,NN)
   38         YY(MM,NN)=F*(Y(MM,NN))
   39      45 CONTINUE
   40         WRITE (6,33)
   41      33 FORMAT(5X,' TWO DIMENSIONAL WALSH HADAMARD TRANSFORM ')
   42         DO 44 MM=1,I
   43         WRITE (6,450) (YY(MM,KL),KL=1,J)
   44     450 FORMAT (5X,16F7.3/)
   45      44 CONTINUE
      C       2 DIMENSIONAL HAR
   46         CALL HAR (G,M2)
   47         DO 226 KK2=1,I
   48         DO 226 KK3=1,J
   49         G1(KK3,KK2)=G(KK2,KK3)
   50      226 CONTINUE
   51         DO 225 K1=1,I
   52         DO 225 KK1=1,J
   53         Z1(K1,KK1)=0.0
   54         DO 225 KKL=1,J
   55         Z1(K1,KK1)=Z1(K1,KK1)+X(K1,KKL)*G1(KKL,KK1)
   56      225 CONTINUE
   57         CALL HAR(G,M1)
   58         DO 445 MK=1,I
```

```
59            DO 445 NK=1,J
60            Y1(MK,NK)=0.0
61            DO 445 KKM=1,I
62            Y1(MK,NK)=Y1(MK,NK)+G(MK,KKM)*Z1(KKM,NK)
63            YH(MK,NK)=F*(Y1(MK,NK))
64       445  CONTINUE
65            WRITE (6,330)
66       330  FORMAT(5X,' TWO DIMENSIONAL HAR TRANSFORM ')
67            DO 444 MK=1,I
68            WRITE (6,410) (YH(MK,L1),L1=1,J)
69       410  FORMAT (5X,16F7.3)
70       444  CONTINUE
         C    2 DIMENSIONAL SLANT
71            CALL SLANT(T,M2)
72            DO 204 JK=1,I
73            DO 204 JL=1,J
74            T1(JL,JK)=T(JK,JL)
75       204  CONTINUE
76            DO 205 JK=1,I
77            DO 205 JL=1,J
78            ZS(JK,JL)=0.0
79            DO 205 LK=1,J
80            ZS(JK,JL)=ZS(JK,JL)+X(JK,LK)*T1(LK,JL)
81       205  CONTINUE
82            CALL SLANT(T,M1)
83            DO 405 JJK=1,I
84            DO 405 JNK=1,J
85            Y2(JJK,JNK)=0.0
86            DO 405 INK=1,I
87            Y2(JJK,JNK)=Y2(JJK,JNK)+T(JJK,INK)*ZS(INK,JNK)
88            YS(JJK,JNK)=F*(Y2(JJK,JNK))
89       405  CONTINUE
90       429  WRITE (6,307)
91       307  FORMAT(5X,' TWO DIMENSIONAL SLANT TRANSFORM ')
92            DO  404 JJK=1,I
93            WRITE (6,415) (YS(JJK,JNK),JNK=1,J)
94       415  FORMAT(5X,16F7.3)
95       404  CONTINUE
         C    2 DIMENSIONAL DCT
96            CALL DCT(D,M2)
97            DO 228 JMK=1,J
98            DO 228 JLK=1,I
99            D1(JMK,JLK)=D(JLK,JMK)
00       228  CONTINUE
01            DO 229 JLK=1,I
02            DO 229 JMK=1,J
03            ZD(JLK,JMK)=0.0
04            DO 229 MJK=1,J
05            ZD(JLK,JMK)=ZD(JLK,JMK)+X(JLK,MJK)*D1(MJK,JMK)
06       229  CONTINUE
07            CALL DCT(D,M1)
08            DO 311 MG=1,I
09            DO 311 NG=1,J
10            YD(MG,NG)=0.0
11            DO 311 LG=1,I
12            YD(MG,NG)=YD(MG,NG)+D(MG,LG)*ZD(LG,NG)
13       311  CONTINUE
14            WRITE (6,660)
15       660  FORMAT(5X,' TWO DIMENSIONAL DISCRETE COSINE TRANSFORM ')
16            DO 312 MG=1,I
17            WRITE (6,701)(YD(MG,NG),NG=1,J)
18       701  FORMAT(5X,16F7.3)
19       312  CONTINUE
```

```
      STOP
      END

      SUBROUTINE HAD(A,N)
     DIMENSION A(32,32)
      N2=2**N
      A(1,1)=1
      A(1,2)=1
      A(2,1)=1
      A(2,2)=-1
      DO 100 I=2, N
      I2= 2**(I-1)
      DO 10 J1=1,I2
      DO 15 J2=1,I2
      A(J1,J2+I2)=A(J1,J2)
      A(J1+I2,J2)=A(J1,J2)
      A(J1+I2,J2+I2)=-A(J1,J2)
15    CONTINUE
10    CONTINUE
100   CONTINUE
     RETURN
      END

     SUBROUTINE HAR(E,M)
     DIMENSION E(32,32)
     DO 100 M=2,5
     N=2**M
     N1=N/2
     N4=N/4
     N5=N/8
     N6=N/16
     S=SORT(2.0)
     DO 5 I1=1,N
     DO 5 J1=1,N
     E(I1,J1)=0.0
5    CONTINUE
     DO 10 I=1,N
     E(1,I)=1.0
10   CONTINUE
     DO 20 J=1,N1
     E(2,J)=1.0
     E(2,J+N1)=-1.0
20   CONTINUE
     DO 30 K=1,N4
     E(3,K)=S
     E(3,K+N4)=-S
     E(4,K+N1)=S
     E(4,K+N1+N4)=-S
30   CONTINUE
     IF (M-2) 120,120,150
150  DO 40 KK=1,N1
     KL=(2*KK)-1
     KM=2*KK
     E(N1+KK,KL)=2.0
     E(N1+KK,KM)=-2.0
40   CONTINUE
     IF (M-4)120,140,140
140  DO 50  K1=1,N5
     DO  70 L1=1,4
     E(4+L1,K1+(L1-1)*N4)=2.0
     E(4+L1,K1+(L1-1)*N4+N5)=-2.0
70   CONTINUE
50   CONTINUE
```

```
181             IF(M-5)120,125,125
182       125   DO 80 L2=1,8
183             DO 60 K2=1,N6
184             F(8+L2,K2+(I2-1)*N5)=2.0
185             F(8+L2,K2+(I2-1)*N5+N6)=-2.0
186       60    CONTINUE
187       80    CONTINUE
188       120   CONTINUE
189       100   CONTINUE
190             RETURN
191             END

192             SUBROUTINE SLANT(S,M)
193             DIMENSION S1(32,32),S(32,32)
194             A1=1.0
195             DO 100 M=1,5
196             N=2**M
197             I1=N/2
198             I2=N/2+1
199             I3=I1+2
200             IF (M-2) 600,950,950
201       600   DO 6 K=1,2
202             S(1,K)=1
203             S(2,1)=1
204             S(2,2)=-1
205       6     CONTINUE
206             GO TO 100
207       950   A=A1
208             B=1.0/(SQRT(1.0+4.0*A*A))
209             A=2.0*B*A1
210             A1=A
211             DO 10 I=3,I1
212             DO 20 J=1,I1
213             S1(1,J)=1
214             S1(1,I1+J)=1
215             S1(2,J)=A*S(1,J)+B*S(2,J)
216             S1(2,J+I1)=-A*S(1,J)+B*S(2,I)
217             IF (M-3) 20,930,930
218       930   S1(I,J)=S(I,J)
219             S1(I,J+I1)=S(I,J)
220             S1(I+I1,J)=S(I,J)
221             S1(I+I1,J+I1)=-S(I,J)
222       20    CONTINUE
223       10    CONTINUE
224             DO 40 J=1,I1
225             S1(I2,J)=S(2,J)
226             S1(I2,J+I1)=-S(2,J)
227             S1(I3,J)=-B*S(1,J)+A*S(2,J)
228             S1(I3,J+I1)=B*S(1,J)+A*S(2,I)
229       40    CONTINUE
230             DO 70 KK=1,N
231             DO 80 JJ=1,N
232             S(KK,JJ)=S1(KK,JJ)
233       80    CONTINUE
234       70    CONTINUE
235       100   CONTINUE
236             RETURN
237             END

238             SUBROUTINE DCT(C,M)
239             DIMENSION C(64,64)
240             DO 20 M=1,5
241             N=2**M
```

```
242             P=1.0*N
243             G=1.0/SQRT(P)
244             R=SQRT(2.0/P)
245             DO 50 K=1,N
246             DO 60 L=1,N
247             A=((2.0*(L-1)+1.0)*(22.0/7.0))/(2.0*N)
248             B=COS(K*A)
249             G=R*B
250             C(1,L)=0
251             C(K+1,L)=G
252     60      CONTINUE
253     50      CONTINUE
254     20      CONTINUE
255             RETURN
256             END
```

Horizontal edge

```
$JOB                ,NOLIST
      $ENTRY
INPUT DATA X(K,L)
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
TWO DIMENSIONAL WALSH HADAMARD TRANSFORM
  9.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.0
  7.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.0
 -1.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.0
  1.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.0
 -1.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.0
  1.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.0
  9.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.0
  7.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.0
  1.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.0
 -1.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.0
 -9.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.0
 -7.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.0
 -9.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.0
 -7.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.0
  1.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.0
 -1.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.0
TWO DIMENSIONAL HAR TRANSFORM
  9.000   9.000  -0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.0
  9.000   9.000  -0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.0
  1.414   1.414   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.0
  0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.0
-10.000 -10.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.0
  8.000   8.000  -0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.0

  0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000
  0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000
  0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000
-10.000 -10.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000
  8.000   8.000  -0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000
  0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000
  0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000
  0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000
  0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000
TWO DIMENSIONAL SLANT TRANSFORM
  9.000   7.794   0.000   0.000   0.000  -0.000   0.000   0.000   0.000  -0.000   0.000   0.000   0.000  -0.000   0.000   0.000
  8.285   7.179   0.000   0.000   0.000  -0.000   0.000   0.000   0.000  -0.000   0.000   0.000   0.000  -0.000   0.000   0.000
  1.000   0.866   0.000   0.000   0.000  -0.000   0.000   0.000   0.000  -0.000   0.000   0.000   0.000  -0.000   0.000   0.000
  6.708   5.812   0.000   0.000   0.000  -0.000   0.000   0.000   0.000  -0.000   0.000   0.000   0.000  -0.000   0.000   0.000
  8.497   7.362   0.000   0.000   0.000  -0.000   0.000   0.000   0.000  -0.000   0.000   0.000   0.000  -0.000   0.000   0.000
  2.480   2.114   0.000   0.000   0.000  -0.000   0.000   0.000   0.000  -0.000   0.000   0.000   0.000  -0.000   0.000   0.000
  7.000   6.065   0.000   0.000   0.000  -0.000   0.000   0.000   0.000  -0.000   0.000   0.000   0.000  -0.000   0.000   0.000
 -3.130  -2.712   0.000   0.000   0.000  -0.000   0.000   0.000   0.000  -0.000   0.000   0.000   0.000  -0.000   0.000   0.000
-12.002 -10.399   0.000   0.000   0.000  -0.000   0.000   0.000   0.000  -0.000   0.000   0.000   0.000  -0.000   0.000   0.000
 -0.308  -0.267   0.000   0.000   0.000  -0.000   0.000   0.000   0.000  -0.000   0.000   0.000   0.000  -0.000   0.000   0.000
 -7.000  -6.065   0.000   0.000   0.000  -0.000   0.000   0.000   0.000  -0.000   0.000   0.000   0.000  -0.000   0.000   0.000
  3.130   2.712   0.000   0.000   0.000  -0.000   0.000   0.000   0.000  -0.000   0.000   0.000   0.000  -0.000   0.000   0.000
 -2.234  -1.937   0.000   0.000   0.000  -0.000   0.000   0.000   0.000  -0.000   0.000   0.000   0.000  -0.000   0.000   0.000
 -3.025  -2.621   0.000   0.000   0.000  -0.000   0.000   0.000   0.000  -0.000   0.000   0.000   0.000  -0.000   0.000   0.000
 -1.000  -0.866   0.000   0.000   0.000  -0.000   0.000   0.000   0.000  -0.000   0.000   0.000   0.000  -0.000   0.000   0.000
 -6.708  -5.812   0.000   0.000   0.000  -0.000   0.000   0.000   0.000  -0.000   0.000   0.000   0.000  -0.000   0.000   0.000
TWO DIMENSIONAL DISCRETE COSINE TRANSFORM
  8.500   8.051  -0.003  -1.355   0.003   0.818  -0.003  -0.590   0.003   0.465  -0.003  -0.387   0.003   0.338  -0.003  -0.296
  4.498   4.226  -0.003  -1.413   0.003   0.853  -0.003  -0.614   0.003   0.485  -0.003  -0.403   0.003   0.348  -0.003  -0.309
  0.409   0.549  -0.000  -0.183   0.000   0.111  -0.000  -0.080   0.000   0.063  -0.000  -0.052   0.000   0.045  -0.000  -0.040
 -3.450  -3.286   0.002   1.099  -0.002  -0.664   0.002   0.479  -0.002  -0.377   0.002   0.314  -0.002  -0.271   0.002   0.240
 -5.820  -5.240   0.003   1.752  -0.003  -1.058   0.003   0.763  -0.003  -0.601   0.003   0.500  -0.004   0.432   0.004  -0.383
 -4.841  -4.394   0.003   1.469  -0.003  -0.887   0.003   0.640  -0.003  -0.504   0.003   0.419  -0.003   0.362   0.003  -0.321
 -1.541  -1.404   0.001   0.470  -0.001  -0.284   0.001   0.205  -0.001  -0.161   0.001   0.134  -0.001   0.116   0.001  -0.103
  2.181   1.964  -0.001  -0.657   0.001   0.397  -0.001  -0.286   0.001   0.225  -0.001  -0.187   0.001   0.162  -0.001  -0.143
  4.352   3.918  -0.002  -1.310   0.002   0.791  -0.003  -0.571   0.003   0.450  -0.003  -0.374   0.003   0.323  -0.003  -0.286
  3.630   3.437  -0.002  -1.214   0.002   0.734  -0.002  -0.530   0.002   0.417  -0.002  -0.347   0.002   0.300  -0.002  -0.266
  1.808   1.428  -0.001  -0.543   0.001   0.328  -0.001  -0.237   0.001   0.186  -0.001  -0.155   0.001   0.134  -0.001  -0.119
 -0.830  -0.787   0.000   0.250  -0.000  -0.151   0.000   0.109  -0.000  -0.086   0.000   0.071  -0.000  -0.062   0.001   0.055
 -2.391  -2.153   0.001   0.720  -0.001  -0.435   0.001   0.314  -0.001  -0.247   0.001   0.206  -0.001  -0.177   0.001   0.157
 -2.320  -2.088   0.001   0.698  -0.001  -0.422   0.001   0.304  -0.001  -0.240   0.001   0.199  -0.001  -0.172   0.001   0.153
 -1.141  -1.027   0.000   0.344  -0.001  -0.207   0.001   0.150  -0.001  -0.118   0.001   0.098  -0.001  -0.085   0.001   0.075
  0.067   0.061  -0.000  -0.020   0.000   0.012  -0.000  -0.009   0.000   0.007  -0.000  -0.006   0.000   0.005  -0.000  -0.004
$E USAGE       OBJECT CODE= 12112 BYTES,ARRAY AREA= 110592 BYTES,TOTAL AREA AVAILABLE= 143704 BYTES
```

Vertical edge

$JOB .NOLIST

$ENTRY

INPUT DATA X(K,L)

0.	0.	0.	0.	0.	0.	0.	4.	1.	4.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	4.	1.	4.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	4.	1.	4.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	4.	1.	4.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	4.	1.	4.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	4.	1.	4.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	4.	1.	4.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	4.	1.	4.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	4.	1.	4.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	4.	1.	4.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	4.	1.	4.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	4.	1.	4.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	4.	1.	4.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	4.	1.	4.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	4.	1.	4.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	4.	1.	4.	0.	0.	0.	0.	0.	0.

TWO DIMENSIONAL WALSH HADAMARD TRANSFORM

9.000	-7.000	1.000	1.000	1.000	1.000	9.000	-7.000	-1.000	-1.000	-9.000	7.000	-9.000	7.000	-1.000	-1.00
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.00
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.00
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.00
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.00
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.00
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.00
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.00
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.00
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.00
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.00
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.00
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.00
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.00
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.00
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.00

TWO DIMENSIONAL HAR TRANSFORM

9.000	9.000	-1.414	0.000	-8.000	10.000	0.000	0.000	0.000	-8.000	10.000	0.000	0.000	0.000	0.000	0.00
9.000	9.000	-1.414	0.000	-8.000	10.000	0.000	0.000	0.000	-8.000	10.000	0.000	0.000	0.000	0.000	0.00
-0.000	-0.000	1.000	0.000	0.000	-0.000	0.000	0.000	0.000	0.000	-0.000	0.000	0.000	0.000	0.000	0.00
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.00
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.00
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.00
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

TWO DIMENSIONAL SLANT TRANSFORM

9.000	7.311	1.000	-6.708	8.497	-2.440	-7.000	-3.130	-12.002	0.308	7.000	3.130	2.236	-3.025	-1.000	6.708
7.798	6.334	0.866	-5.812	7.362	-2.114	-6.065	-2.712	-10.399	0.267	6.065	2.712	1.937	-2.621	-0.866	5.812
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.000	-0.000	0.000	0.000	0.000	0.000	0.000	0.000	-0.000	-0.000	-0.000	-0.000	-0.000	0.000	-0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	-0.000	0.000	0.000	0.000
-0.000	-0.000	0.000	0.000	0.000	0.000	0.000	0.000	-0.000	-0.000	0.000	-0.000	0.000	0.000	0.000	-0.000
0.000	0.000	-0.000	-0.000	0.000	0.000	-0.000	0.000	0.000	-0.000	-0.000	-0.000	-0.000	0.000	0.000	-0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	-0.000	-0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	-0.000	0.000	0.000	0.000
-0.000	0.000	-0.000	0.000	-0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

TWO DIMENSIONAL DISCRETE COSINE TRANSFORM

4.500	4.254	-0.617	-4.919	-5.817	-2.920	1.581	4.594	4.343	1.434	-1.825	-3.274	-2.379	-0.334	1.153	1.257
[illegible]	[illegible]	-0.556	-4.429	-5.238	-2.629	1.423	4.138	3.910	1.291	-1.643	-2.947	-2.142	-0.301	1.038	1.132
[illegible]	-0.002	0.000	0.003	0.003	0.002	-0.001	-0.003	-0.002	-0.001	0.001	0.002	0.001	0.000	-0.001	-0.001
-1.355	-1.281	0.186	1.481	1.751	0.879	-0.476	-1.384	-1.307	-0.432	0.550	0.986	0.716	0.101	-0.347	-0.378
[illegible]	[illegible]	-0.000	-0.003	-0.003	-0.002	0.001	0.003	0.002	0.001	-0.001	-0.002	-0.001	-0.000	0.001	0.001
[illegible]	[illegible]	-0.112	-0.494	-1.058	-0.531	0.297	0.836	0.790	0.261	-0.332	-0.595	-0.433	-0.061	0.210	0.229
[illegible]	[illegible]	[illegible]	0.003	0.003	0.002	-0.001	-0.003	-0.003	-0.001	0.001	0.002	0.001	0.000	-0.001	-0.001
[illegible]	[illegible]	[illegible]	[illegible]	0.763	0.383	-0.207	-0.603	-0.569	-0.188	0.239	0.429	0.312	0.044	-0.151	-0.165
[illegible]	[illegible]	[illegible]	[illegible]	-0.003	-0.002	0.001	0.003	0.003	0.001	-0.001	-0.002	-0.001	-0.000	0.001	0.001
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	-0.302	0.143	0.475	0.449	0.148	-0.189	-0.338	-0.246	-0.035	0.119	0.130
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0.002	-0.001	-0.003	-0.003	-0.001	0.001	0.002	0.001	0.000	-0.001	-0.001
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0.251	-0.134	-0.395	-0.373	-0.123	0.157	0.281	0.204	0.029	-0.099	-0.108
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0.001	0.003	0.003	0.001	-0.001	-0.002	-0.001	-0.000	0.001	0.001
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	-0.217	0.117	0.341	0.322	0.106	-0.135	-0.243	-0.176	-0.025	0.086	0.093
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0.002	-0.001	-0.003	-0.003	-0.001	0.001	0.002	0.001	0.000	-0.001	-0.001
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0.192	-0.104	-0.302	-0.286	-0.094	0.120	0.215	0.157	0.022	-0.076	-0.083

[illegible] OBJECT CODE= 12112 BYTES,ARRAY AREA= 110592 BYTES,TOTAL AREA AVAILABLE= 143704 BYTES

[illegible] NUMBER OF ERRORS= 0, NUMBER OF WARNINGS= 0, NUMBER OF EXTENSIONS= 0

Slant line @ 45°

```
$JOB                ,NOLIST
    SENTRY
INPUT DATA X(K,L)
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
TWO DIMENSIONAL WALSH HADAMARD TRANSFORM
 1.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.00
 0.000 -1.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.00
 0.000  0.000 -1.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.00
 0.000  0.000  0.000  1.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.00
 0.000  0.000  0.000  0.000 -1.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.00
 0.000  0.000  0.000  0.000  0.000  1.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.00
 0.000  0.000  0.000  0.000  0.000  0.000  1.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.00
 0.000  0.000  0.000  0.000  0.000  0.000  0.000 -1.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.00
 0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000 -1.000  0.000  0.000  0.000  0.000  0.000  0.000  0.00
 0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  1.000  0.000  0.000  0.000  0.000  0.000  0.00
 0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  1.000  0.000  0.000  0.000  0.000  0.00
 0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000 -1.000  0.000  0.000  0.000  0.00
 0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  1.000  0.000  0.000  0.00
 0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000 -1.000  0.000  0.00
 0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000 -1.000  0.00
 0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  1.00
TWO DIMENSIONAL HAR TRANSFORM
 1.000  1.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
 1.000  1.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
 0.000  0.000 -2.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
 0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
 0.000  0.000  0.000  0.000  0.000 -2.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
 0.000  0.000  0.000  0.000 -2.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000

 0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
 0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
 0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
 0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000 -1.000  0.000  0.000  0.000  0.000  0.000
 0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000 -1.000  0.000  0.000  0.000  0.000  0.000  0.000
 0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000 -1.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
 0.000  0.000  0.000  0.000  0.000  0.000  0.000 -1.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
 0.000  0.000  0.000  0.000  0.000  0.000 -1.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
 0.000  0.000  0.000  0.000  0.000 -1.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
 0.000  0.000  0.000  0.000 -1.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
TWO DIMENSIONAL SLANT TRANSFORM
 1.000  0.844  0.000 -0.000 -0.000 -0.000 -0.000  0.000 -0.000  0.000 -0.000  0.000 -0.000  0.000 -0.000  0.000
 0.844  0.501  0.000 -0.000 -0.000 -0.000 -0.000  0.000 -0.000  0.000 -0.000  0.000 -0.000  0.000 -0.000 -0.000
 0.000  0.000  1.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
 0.000  0.000 -0.000 -1.000  0.000  0.000 -0.000 -0.000  0.000  0.000 -0.000 -0.000  0.000  0.000 -0.000 -0.000
 0.000  0.000 -0.000  0.000  1.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000 -0.000 -0.000  0.000  0.000
 0.000  0.000 -0.000  0.000 -0.000 -1.000 -0.000  0.000  0.000  0.000  0.000  0.000 -0.000 -0.000  0.000  0.000
 0.000 -0.000  0.000 -0.000  0.000 -0.000 -1.000 -0.000  0.000  0.000 -0.000  0.000 -0.000  0.000 -0.000  0.000
 0.000  0.000  0.000  0.000  0.000  0.000  0.000  1.000  0.000 -0.000  0.000  0.000  0.000  0.000  0.000  0.000
 0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  1.000  0.000  0.000  0.000 -0.000 -0.000  0.000  0.000
-0.000 -0.000 -0.000  0.000 -0.000  0.000  0.000  0.000 -0.000 -1.000 -0.000  0.000 -0.000  0.000 -0.000  0.000
 0.000 -0.000  0.000 -0.000  0.000 -0.000  0.000 -0.000  0.000 -0.000 -1.000 -0.000  0.000 -0.000  0.000 -0.000
 0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  1.000  0.000 -0.000  0.000  0.000
 0.000  0.000  0.000  0.000 -0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  1.000  0.000  0.000  0.000
-0.000 -0.000 -0.000  0.000 -0.000  0.000  0.000  0.000 -0.000 -0.000 -0.000 -0.000  0.000  1.000  0.000  0.000
 0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  1.000 -1.000
TWO DIMENSIONAL DISCRETE COSINE TRANSFORM
 0.500  0.450 -0.000 -0.151  0.000  0.091 -0.000 -0.066  0.000  0.052 -0.000 -0.043  0.000  0.037 -0.000 -0.033
 0.450  0.318 -0.212 -0.319 -0.082  0.108 -0.018 -0.108 -0.000  0.066 -0.007 -0.066 -0.004  0.049 -0.001 -0.049
-0.000 -0.212 -0.500 -0.382  0.001  0.153 -0.000 -0.101  0.000  0.077 -0.000 -0.063  0.000  0.054 -0.000 -0.047
-0.151 -0.319 -0.382 -0.107  0.273  0.319  0.049 -0.066  0.034  0.108  0.020 -0.089  0.018  0.066  0.009 -0.080
 0.000 -0.082  0.001  0.273  0.500  0.354 -0.001 -0.137  0.000  0.000 -0.001 -0.070  0.000  0.058 -0.001 -0.050
 0.091  0.108  0.153  0.319  0.354  0.065 -0.289 -0.319 -0.040  0.049 -0.082 -0.108 -0.025  0.040 -0.018 -0.066
-0.000 -0.018 -0.000  0.049 -0.001 -0.289 -0.500 -0.343  0.001  0.130 -0.000 -0.085  0.201  0.066 -0.000 -0.055
-0.066 -0.108 -0.101 -0.066 -0.137 -0.319 -0.343 -0.047  0.297  0.319  0.045 -0.040  0.045  0.108  0.028 -0.035
 0.000 -0.009  0.000  0.034  0.000 -0.040  0.001  0.297  0.500  0.338 -0.002 -0.126  0.000  0.082 -0.001 -0.064
 0.052  0.066  0.077  0.108  0.090  0.049  0.130  0.319  0.338  0.038 -0.301 -0.319 -0.087  0.035 -0.048 -0.108
-0.000 -0.007 -0.000  0.020 -0.001 -0.082 -0.000  0.085 -0.002 -0.301 -0.500 -0.335  0.002  0.124 -0.000 -0.041
-0.043 -0.066 -0.063 -0.089 -0.070 -0.108 -0.085 -0.040 -0.126 -0.319 -0.335 -0.032  0.304  0.500  0.333 -0.000
 0.000 -0.004  0.000  0.014  0.000 -0.025  0.001  0.045  0.000 -0.087  0.002  0.304  0.500  0.333  0.000 -0.033
 0.037  0.049  0.054  0.066  0.058  0.080  0.066  0.108  0.082  0.035  0.124  0.319  0.333  0.320 -0.305 -0.319
-0.000 -0.003 -0.000  0.009 -0.001 -0.018 -0.000  0.028 -0.001 -0.048 -0.000  0.089 -0.003 -0.305 -0.500 -0.331
 0.033 -0.049 -0.047 -0.040 -0.050 -0.066 -0.055 -0.035 -0.064 -0.108 -0.041 -0.033 -0.125 -0.319 -0.331 -0.327

PAGE        OBJECT CODE=  12112 BYTES,ARRAY AREA=  110592 BYTES,TOTAL AREA AVAILABLE=  143704  BYTES
STICS       NUMBER OF ERRORS=     0, NUMBER OF WARNINGS=     0, NUMBER OF EXTENSIONS=     0
  TIME=  1.22 SEC,EXECUTION TIME=  20.37 SEC.      12.54.39    SATURDAY     3 MAR 79
```

Slant line with shades @ 45°

```
      $JOB                ,NOLIST
            $ENTRY
INPUT DATA X(K,L)
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 2. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 3. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 4. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 5. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 6. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 7. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 8. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 7. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 6. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 5. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 4. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 3. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 2. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
TWO DIMENSIONAL WALSH HADAMARD TRANSFORM
 4.438  0.063  0.063 -0.063  0.063 -0.063 -0.063  0.063 -0.063 -0.438 -0.938 -0.063 -1.938 -0.063 -0.063  0.063
-0.063 -4.438  0.063 -0.063  0.063 -0.063 -0.063  0.063  0.438  0.063  0.063  0.938  0.063  1.938 -0.063  0.063
-0.063  0.063 -4.438 -0.063  0.063 -0.063 -0.063  0.063  0.938  0.063  0.063  0.438  0.063 -0.063  1.938  0.063
-0.063  0.063  0.063  4.438  0.063 -0.063 -0.063  0.063 -0.063 -0.938 -0.438 -0.063  0.063 -0.063 -0.063 -1.938
-0.063  0.063  0.063 -0.063 -4.438 -0.063 -0.063  0.063  1.938  0.063  0.063 -0.063  0.063  0.438  0.938  0.063
-0.063  0.063  0.063 -0.063  0.063  4.438 -0.063  0.063 -0.063 -1.938  0.063 -0.063 -0.438 -0.063 -0.063 -0.938
-0.063  0.063  0.063 -0.063  0.063 -0.063  4.438  0.063 -0.063  0.063 -1.938 -0.063 -0.938 -0.063 -0.063 -0.438
-0.063  0.063  0.063 -0.063  0.063 -0.063 -0.063 -4.438 -0.063  0.063  0.063  1.938  0.063  0.938  0.438  0.063
-0.063  0.438  0.938  0.063  1.938  0.063  0.063 -0.063 -4.438 -0.063 -0.063  0.063 -0.063  0.063  0.063 -0.063
 0.438 -0.063 -0.063 -0.938 -0.063 -1.938  0.063 -0.063  0.063  4.438 -0.063  0.063 -0.063  0.063  0.063 -0.063
 0.938 -0.063 -0.063 -0.438 -0.063  0.063 -1.938 -0.063  0.063 -0.063  4.438  0.063 -0.063  0.063  0.063 -0.063
 0.063  0.938  0.438  0.063 -0.063  0.063  0.063  1.938  0.063 -0.063 -0.063 -4.438 -0.063  0.063  0.063 -0.063
 1.938 -0.063 -0.063  0.063 -0.063 -0.438 -0.938 -0.063  0.063 -0.063 -0.063  0.063  4.438  0.063  0.063 -0.063
 0.063  1.938 -0.063  0.063  0.438  0.063  0.063  0.938  0.063 -0.063 -0.063  0.063 -0.063 -4.438  0.063 -0.063
 0.063 -0.063  1.938  0.063  0.938  0.063  0.063  0.438  0.063 -0.063 -0.063  0.063 -0.063  0.063 -4.438 -0.063
 0.063 -0.063 -0.063 -1.938 -0.063 -0.938 -0.438 -0.063  0.063 -0.063 -0.063  0.063 -0.063  0.063  0.063  4.438
NO DIMENSIONAL HAR TRANSFORM
 4.438  4.438 -0.088  0.000 -1.875  2.000  0.000  0.000 -0.500 -0.375  0.500  0.500  0.000  0.000  0.000  0.000
 4.438  4.438 -0.088  0.000 -1.875  2.000  0.000  0.000 -0.500 -0.375  0.500  0.500  0.000  0.000  0.000  0.000
-0.088 -0.088 -4.875  0.000  2.652  2.828  0.000  0.000  0.707  0.530  0.707  0.707  0.000  0.000  0.000  0.000
 2.000  2.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
 2.000 -2.000  2.828  0.000  0.000 -0.000  0.000  0.000  0.000  0.000  0.000 -1.000  1.000  0.000  0.000  0.000  0.000
 1.875  1.875  2.652  0.000 -4.750  0.000  0.000  0.000  1.000 -0.750  0.000  1.000  0.000  0.000  0.000  0.000  0.000

 1.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
 1.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
 1.500 -0.500  0.707  0.000  0.000  1.000  0.000  0.000  0.000  0.000  0.000  0.000 -2.500  0.000  0.000  0.000  0.000
 1.500 -0.500  0.707  0.000  0.000 -1.000  0.000  0.000  0.000  0.000 -4.500  0.000  0.000  0.000  0.000  0.000  0.000
 1.375  0.375  0.530  0.000 -0.750  0.000  0.000  0.000  0.000 -6.250  0.000  0.000  0.000  0.000  0.000  0.000  0.000
 1.500  0.500  0.707  0.000  1.000  0.000  0.000  0.000 -2.500  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
 1.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
 1.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
 1.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
 1.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
DIMENSIONAL SLANT TRANSFORM
 4.438  3.881 -0.063  0.028 -0.088  0.063  0.063 -0.028 -2.196  0.114 -0.063  0.028 -0.088  0.063  0.063 -0.028
 4.438  2.703 -0.051  0.023 -0.088  0.035  0.051 -0.023 -1.908  0.671  0.058  0.023  0.416  0.035  0.051 -0.023
 0.063 -0.058  4.438 -0.028 -0.088  0.063  0.063 -0.028 -0.341  0.114 -0.063 -0.643 -0.088 -0.738 -1.938 -0.028
 0.028 -0.026 -0.028 -4.487 -0.037  0.010  0.028 -0.013  0.043  0.051 -0.643  0.012 -0.038  0.010  0.028  1.987
 0.088 -0.077 -0.088  0.037  4.387  0.057  0.088 -0.038 -1.828  0.153 -0.088  0.038 -0.112 -1.143 -0.811 -0.037
 0.063 -0.039 -0.063  0.010 -0.057 -4.471  0.063 -0.010  0.065  0.078  0.738  0.010  1.143  0.029  0.063 -0.010
 0.063 -0.058 -0.063  0.028 -0.088  0.063 -4.438 -0.028  0.095  0.114  1.938  0.028  0.811  0.063  0.063 -0.643
 0.028 -0.026 -0.028  0.013 -0.037  0.010  0.028  4.487  0.043  0.051 -0.028 -1.987 -0.037  0.010 -0.643 -0.012
 0.196 -1.897 -0.301 -0.043 -1.828 -0.065 -0.095  0.043  4.358 -0.178  0.095 -0.043  0.128 -0.065 -0.095  0.043
 0.114  0.474 -0.114  0.051 -0.153  0.078  0.114 -0.051  0.178 -6.269  0.265  0.051  1.541  0.078  0.114 -0.051
 0.063  0.166  0.063  0.643  0.088  0.738  1.938  0.028 -0.095  0.265 -4.438 -0.028  0.088 -0.643 -0.063  0.028
 0.028  0.024 -0.643 -0.012  0.038 -0.010 -0.028 -1.987 -0.043 -0.051  0.028  4.487  0.037 -0.010 -0.028  0.013
 0.088  0.562  0.088  0.112  1.143  0.811  0.037 -0.128  1.541  0.088 -0.037 -4.347 -0.057 -0.088  0.038  0.038
 0.063  0.039 -0.738 -0.010 -1.143 -0.029 -0.043  0.010 -0.065 -0.078  0.043 -0.010  0.057  4.471 -0.043  0.010
 0.063 -0.058 -1.938 -0.028 -0.811 -0.063 -0.643 -0.095 -0.114  0.063 -0.028  0.088 -0.063  1.438  4.438  0.028
 0.028  0.026  0.028  1.987  0.037 -0.010  0.643  0.012 -0.043 -0.051  0.028 -0.013  0.037 -0.010 -0.028 -4.487
DIMENSIONAL DISCRETE COSINE TRANSFORM
 2.219  2.000 -0.004 -1.275 -1.098 -0.358  0.013 -0.014 -0.041  0.007 -0.021 -0.098 -0.082  0.002  0.028 -0.011
 1.903  1.745 -0.588 -1.753 -1.154 -0.086  0.286  0.105 -0.001  0.056  0.022 -0.085 -0.071  0.037  0.062  0.003
 1.105 -0.579 -1.485 -1.226 -0.003  0.913  0.803  0.245  0.005  0.079  0.087 -0.006 -0.005  0.077  0.040  0.015
 0.466 -1.741 -1.217  0.007  1.459  1.735  0.868 -0.076 -0.308 -0.076  0.049 -0.007 -0.014  0.086  0.030 -0.035
 0.140 -1.150  0.000  1.472  2.140  1.470 -0.023 -0.973 -0.852 -0.248  0.029 -0.018 -0.051 -0.012 -0.039 -0.101
 0.140 -0.098  0.921  1.758  1.500  0.061 -1.440 -1.764 -0.915  0.028  0.300  0.115 -0.009  0.005 -0.028 -0.094
 0.113  0.241  0.803  0.803  0.012 -1.415 -2.161 -1.474 -0.000  0.913  0.796  0.266  0.015  0.042  0.036 -0.031
 0.135  0.081  0.237 -0.064 -0.944 -1.720 -1.459 -0.009  1.459  1.724  0.873 -0.028 -0.259 -0.067  0.036 -0.011
 0.141 -0.005 -0.006 -0.320 -0.832 -0.906  0.029  1.474  2.107  1.458 -0.037 -0.938 -0.741 -0.252  0.037 -0.006
 0.138  0.080  0.071 -0.111 -0.285  0.016  0.936  1.749  1.489  0.004 -1.493 -1.772 -0.900  0.035  0.200  0.122
 0.121  0.044  0.087  0.004 -0.029  0.250  0.706  0.913  0.017 -1.456 -2.211 -1.495 -0.012  0.904  0.786  0.057
 0.145 -0.051  0.001 -0.031 -0.062  0.071  0.248 -0.015 -0.896 -1.720 -1.459 -0.008  1.455  1.718  0.864 -0.025
 0.182 -0.067  0.005 -0.003 -0.051 -0.026 -0.016 -0.248 -0.782 -0.868  0.037  1.493  2.207  1.488 -0.087 -0.037
 0.135  0.002  0.084  0.090  0.040  0.029  0.021 -0.119 -0.271  0.022  0.937  1.775  1.492 -0.006 -1.507 -1.777
 0.128  0.006  0.080  0.085  0.038  0.027  0.036 -0.036  0.249  0.786  0.912  0.019 -1.462 -2.219 -1.805
 0.153 -0.041  0.009 -0.000 -0.043 -0.042 -0.012 -0.035 -0.061  0.064  0.236 -0.013 -0.888 -1.716 -1.454  0.004
F ------OBJECT CODE= 12112 BYTES.ARRAY AREA= 110592 BYTES.TOTAL AREA AVAILABLE= 163708 BYTES
```

Slant line with shades @ 135°

```
      $JOB              ,NOLIST

      SENTRY
INPUT DATA X(K,L)

 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 2. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 3. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 4. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 5. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 6. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 7. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 8. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 7. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 6. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 5. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 4. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 3. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 2. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0.
TWO DIMENSIONAL WALSH HADAMARD TRANSFORM

 8.438  0.063  0.063 -0.063  0.063 -0.063 -0.063  0.063 -0.063 -0.438 -0.938 -0.063 -1.938 -0.063 -0.063  0.06
 0.063  8.438 -0.063  0.063 -0.063  0.063  0.063 -0.063 -0.438 -0.063 -0.063 -0.938 -0.063 -1.938  0.063 -0.06
 0.063 -0.063  8.438  0.063 -0.063  0.063  0.063 -0.063 -0.938 -0.063 -0.063 -0.438 -0.063  0.063 -1.938 -0.06
-0.063  0.063  0.063  8.438  0.063 -0.063 -0.063  0.063 -0.063 -0.938 -0.438 -0.063  0.063 -0.063 -0.063 -1.93
 0.063 -0.063 -0.063  0.063  8.438  0.063  0.063 -0.063 -1.938 -0.063 -0.063  0.063 -0.063 -0.438 -0.938 -0.06
-0.063  0.063  0.063 -0.063  0.063  8.438 -0.063  0.063 -0.063 -1.938  0.063 -0.063 -0.438 -0.063 -0.063 -0.93
-0.063  0.063  0.063 -0.063  0.063 -0.063  8.438  0.063 -0.063  0.063 -1.938 -0.063 -0.938 -0.063 -0.063 -0.43
 0.063 -0.063 -0.063  0.063 -0.063  0.063  0.063  8.438  0.063 -0.063 -0.063 -1.938 -0.063 -0.938 -0.438 -0.06
-0.063 -0.438 -0.938 -0.063 -1.938 -0.063 -0.063  0.063  8.438  0.063  0.063 -0.063  0.063 -0.063 -0.063  0.06
-0.438 -0.063 -0.063 -0.938 -0.063 -1.938  0.063 -0.063  0.063  8.438 -0.063  0.063 -0.063  0.063  0.063 -0.06
-0.938 -0.063 -0.063 -0.438 -0.063  0.063 -1.938 -0.063  0.063 -0.063  8.438  0.063 -0.063  0.063  0.063 -0.06
-0.063 -0.938 -0.438 -0.063  0.063 -0.063 -0.063 -1.938 -0.063  0.063  0.063  8.438  0.063 -0.063 -0.063  0.06
-1.938 -0.063 -0.063  0.063 -0.063 -0.438 -0.938 -0.063  0.063 -0.063 -0.063  0.063  8.438  0.063  0.063 -0.06
-0.063 -1.938  0.063 -0.063 -0.438 -0.063 -0.063 -0.938 -0.063  0.063  0.063 -0.063  0.063  8.438 -0.063  0.06
-0.063  0.063 -1.938 -0.063 -0.938 -0.063 -0.063 -0.438 -0.063  0.063  0.063 -0.063  0.063 -0.063  8.438  0.06
 0.063 -0.063 -0.063 -1.938 -0.063 -0.938 -0.438 -0.063  0.063 -0.063 -0.063  0.063 -0.063  0.063  0.063  8.43
TWO DIMENSIONAL HAR TRANSFORM

 8.438  8.438 -0.088  0.000 -1.875  2.000  0.000  0.000 -0.500 -0.375  0.500  0.500  0.000  0.000  0.000  0.00
 8.438  8.438 -0.088  0.000 -1.875  2.000  0.000  0.000 -0.500 -0.375  0.500  0.500  0.000  0.000  0.000  0.00
-0.088 -0.088  8.875  0.000 -2.452 -2.828  0.000  0.000 -0.707 -0.530 -0.707 -0.707  0.000  0.000  0.000  0.00
 0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.00
-1.875 -1.875 -2.452  0.000  8.750  0.000  0.000  0.000 -1.000  0.750  0.000  0.000  0.000  0.000  0.000  0.00
 2.000  2.000 -2.828  0.000  0.000  0.000  0.000  0.000  0.000  0.000  1.000 -1.000  0.000  0.000  0.000  0.00

 0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.00
 0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.00
-0.500 -0.500 -0.707  0.000 -1.000  0.000  0.000  0.000  2.500  0.000  0.000  0.000  0.000  0.000  0.000  0.00
-0.375 -0.375 -0.530  0.000  0.750  0.000  0.000  0.000  0.000  6.250  0.000  0.000  0.000  0.000  0.000  0.00
 0.500  0.500  0.707  0.000  0.000  1.000  0.000  0.000  0.000  0.000  6.500  0.000  0.000  0.000  0.000  0.00
 0.500  0.500  0.707  0.000  0.000 -1.000  0.000  0.000  0.000  0.000  0.000  2.500  0.000  0.000  0.000  0.00
 0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.00
 0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.00
 0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.00
 0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.00
TWO DIMENSIONAL SLANT TRANSFORM

 8.438  3.841 -0.063  0.029 -0.088  0.063 -0.029 -2.194  0.118 -0.063  0.029 -0.088  0.063 -0.063 -0.02
 3.841  3.954 -0.059  0.034 -0.077  0.039  0.059 -0.034 -1.497 -0.474 -0.144  0.029 -0.592  0.039 -0.059 -0.02
-0.063 -0.059  8.438  0.029 -0.088  0.063 -0.029 -0.841  0.118 -0.063 -0.683 -0.088 -0.778 -1.938 -0.059 -0.02
 0.029  0.034  0.029  4.487  0.037 -0.019 -0.029  0.012 -0.083 -0.651 -0.443 -0.013  0.059 -0.019 -0.029 -1.99
-0.088 -0.077 -0.088  0.037  8.187 -0.057  0.088 -0.038 -1.826  0.153 -0.088 -0.038 -0.115 -1.183 -0.911 -0.07
 0.063  0.039  0.063 -0.019  0.057  8.471 -0.063  0.019 -0.065 -0.078 -0.733 -0.019 -1.143 -0.029 -0.043 -0.01
-0.029 -0.059 -0.029  0.088 -0.063  0.029  8.438 -0.029 -0.095 -0.118 -1.938 -0.029 -0.811 -0.063 -0.063 -0.64
-2.194 -1.497 -0.841 -0.083 -1.826 -0.065 -0.095  0.043  8.156 -0.174  0.095 -0.063  0.128 -0.065 -0.095  0.00
 0.118 -0.474  0.118 -0.051  0.153 -0.078 -0.118  0.051 -0.174  6.249 -0.265 -0.051 -1.581 -0.078 -0.118  0.05
-0.063 -0.144 -0.063 -0.683 -0.088 -0.733 -1.938 -0.029  0.095 -0.265  8.438  0.029 -0.088  0.063 -0.029 -0.07
 0.029  0.029 -0.443 -0.013  0.038 -0.019 -0.029 -1.947 -0.063 -0.051  0.029  8.487  0.037 -0.019 -0.029  0.01
-0.088 -0.562 -0.088  0.038 -0.113 -1.183 -0.811 -0.037  0.128 -1.581 -0.088  0.037  8.187  0.057 -0.088 -0.07
 0.063  0.039 -0.733 -0.019 -1.143 -0.029 -0.065 -0.078  0.063 -0.019 -0.057  8.471 -0.063 -0.043  0.01
 0.029 -0.059 -1.938 -0.029 -0.811 -0.063 -0.063 -0.095 -0.118  0.063 -0.029 -0.063  8.438 -0.063 -0.02
-0.029 -0.029 -0.029 -1.947 -0.037  0.019 -0.663 -0.013  0.063  0.051 -0.029  0.012 -0.037  0.019 -0.029  8.48
TWO DIMENSIONAL DISCRETE COSINE TRANSFORM

 2.219  2.000 -0.006 -1.275 -1.006 -0.354  0.013 -0.014 -0.041  0.007 -0.021 -0.008 -0.002  0.002 -0.029 -0.01
 2.000  0.215  0.576 -0.774 -0.152 -0.760 -0.020 -0.005 -0.004 -0.064 -0.073 -0.007 -0.038 -0.004 -0.004 -0.00
-0.006  0.576  1.445  1.228  0.005 -0.911 -0.807 -0.245 -0.004 -0.079 -0.087  0.007  0.005 -0.077 -0.088 -0.01
-1.275 -0.774  1.228  2.224  1.468 -0.033 -0.749 -0.510 -0.040 -0.008 -0.008 -0.009 -0.007 -0.007 -0.008 -0.00
-1.006 -0.152  0.005  1.468  2.190  1.483 -0.019 -0.971 -0.832 -0.248  0.029 -0.018 -0.051 -0.012 -0.030 -0.10
-0.354 -0.760 -0.911 -0.033  1.483  2.208  1.489 -0.062 -0.900 -0.758 -0.254 -0.027 -0.013 -0.027 -0.053 -0.02
 0.013 -0.020 -0.807 -0.749 -0.019  1.489  2.161  1.488 -0.014 -0.910 -0.794 -0.247 -0.015 -0.062 -0.034  0.02
-0.014 -0.005 -0.245 -0.510 -0.971 -0.062  1.488  2.238  1.470 -0.026 -0.919 -0.788  0.002  0.017 -0.007 -0.00
-0.041 -0.004 -0.004 -0.040 -0.832 -0.900 -0.014  1.470  2.197  1.441 -0.028 -0.933 -0.781 -0.224  0.037 -0.00
 0.007 -0.064 -0.079 -0.008 -0.248 -0.758 -0.910 -0.026  1.441  2.198  1.484 -0.012 -0.975 -0.786 -0.247 -0.00
-0.021 -0.073 -0.087  0.007  0.029 -0.254 -0.794 -0.919 -0.028  1.484  2.211  1.504  0.023 -0.108 -0.784 -0.24
-0.008 -0.007 -0.008 -0.009 -0.018 -0.027 -0.247 -0.788 -0.933 -0.012  1.504  2.236  1.481 -0.016 -0.020 -0.80
-0.029 -0.038 -0.004 -0.007 -0.051 -0.013 -0.015 -0.062 -0.781 -0.975 -0.023  1.481  2.207  1.469 -0.038 -0.93
-0.004 -0.004 -0.077 -0.027 -0.012 -0.053 -0.042  0.002 -0.224 -0.786 -0.808 -0.016  1.469  2.189  1.484 -0.00
-0.006 -0.080 -0.088 -0.086 -0.030 -0.027 -0.034  0.017 -0.037 -0.247 -0.784 -0.920 -0.038  1.484  2.210  1.50
-0.002 -0.038 -0.015 -0.002 -0.101 -0.022  0.031 -0.061 -0.004 -0.003 -0.269 -0.808 -0.930 -0.004  1.508  2.20

USAGE       OBJECT CODE= 12112 BYTES, ARRAY AREA= 110592 BYTES, TOTAL AREA AVAILABLE= 143708 BYTES

OSTICS      NUMBER OF ERRORS=    0, NUMBER OF WARNINGS=    0, NUMBER OF EXTENSIONS=    0
```

Horizontal line with shades

$JOB .NOLIST

$ENTRY

INPUT DATA X(K,L)

0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
1.	2.	3.	4.	5.	6.	7.	8.	7.	6.	5.	4.	3.	2.	1.	1.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.

TWO DIMENSIONAL WALSH HADAMARD TRANSFORM

4.063	-0.063	-0.063	0.063	-0.063	0.063	0.063	-0.063	0.438	-0.438	-0.938	-0.063	-1.938	-0.063	-0.063	0.063
4.063	-0.063	-0.063	0.063	-0.063	0.063	0.063	-0.063	0.438	-0.438	-0.938	-0.063	-1.938	-0.063	-0.063	0.063
-4.063	0.063	0.063	-0.063	0.063	-0.063	-0.063	0.063	-0.438	0.438	0.938	0.063	1.938	0.063	0.063	-0.063
-4.063	0.063	0.063	-0.063	0.063	-0.063	-0.063	0.063	-0.438	0.438	0.938	0.063	1.938	0.063	0.063	-0.063
-4.063	0.063	0.063	-0.063	0.063	-0.063	-0.063	0.063	-0.438	0.438	0.938	0.063	1.938	0.063	0.063	-0.063
-4.063	0.063	0.063	-0.063	0.063	-0.063	-0.063	0.063	-0.438	0.438	0.938	0.063	1.938	0.063	0.063	-0.063
4.063	-0.063	-0.063	0.063	-0.063	0.063	0.063	-0.063	0.438	-0.438	-0.938	-0.063	-1.938	-0.063	-0.063	0.063
4.063	-0.063	-0.063	0.063	-0.063	0.063	0.063	-0.063	0.438	-0.438	-0.938	-0.063	-1.938	-0.063	-0.063	0.063
4.063	-0.063	-0.063	0.063	-0.063	0.063	0.063	-0.063	0.438	-0.438	-0.938	-0.063	-1.938	-0.063	-0.063	0.063
4.063	-0.063	-0.063	0.063	-0.063	0.063	0.063	-0.063	0.438	-0.438	-0.938	-0.063	-1.938	-0.063	-0.063	0.063
-4.063	0.063	0.063	-0.063	0.063	-0.063	-0.063	0.063	-0.438	0.438	0.938	0.063	1.938	0.063	0.063	-0.063
-4.063	0.063	0.063	-0.063	0.063	-0.063	-0.063	0.063	-0.438	0.438	0.938	0.063	1.938	0.063	0.063	-0.063
-4.063	0.063	0.063	-0.063	0.063	-0.063	-0.063	0.063	-0.438	0.438	0.938	0.063	1.938	0.063	0.063	-0.063
-4.063	0.063	0.063	-0.063	0.063	-0.063	-0.063	0.063	-0.438	0.438	0.938	0.063	1.938	0.063	0.063	-0.063
4.063	-0.063	-0.063	0.063	-0.063	0.063	0.063	-0.063	0.438	-0.438	-0.938	-0.063	-1.938	-0.063	-0.063	0.063
4.063	-0.063	-0.063	0.063	-0.063	0.063	0.063	-0.063	0.438	-0.438	-0.938	-0.063	-1.938	-0.063	-0.063	0.063

TWO DIMENSIONAL HAR TRANSFORM

4.063	4.063	0.619	0.000	-2.000	1.875	0.000	0.000	-0.500	-0.500	0.500	0.375	0.000	0.000	0.000	0.000
4.063	4.063	0.619	0.000	-2.000	1.875	0.000	0.000	-0.500	-0.500	0.500	0.375	0.000	0.000	0.000	0.000
5.745	5.745	0.875	0.000	-2.828	2.652	0.000	0.000	-0.707	-0.707	0.707	0.530	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
-4.125	-4.125	-1.237	0.000	4.000	-3.750	0.000	0.000	1.000	1.000	-1.000	-0.750	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
-4.125	-4.125	-1.237	0.000	4.000	-3.750	0.000	0.000	1.000	1.000	-1.000	-0.750	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

TWO DIMENSIONAL SLANT TRANSFORM

4.063	3.686	0.063	-0.028	0.084	-0.043	-0.063	0.028	-2.196	-0.300	-0.063	0.028	-0.084	0.043	0.063	-0.028
4.180	3.792	0.064	-0.029	0.086	-0.044	-0.064	0.029	-2.259	-0.309	-0.064	0.029	-0.086	0.044	0.064	-0.029
-4.063	-3.686	-0.063	0.028	-0.084	0.043	0.063	-0.028	2.196	0.300	0.063	-0.028	0.084	-0.043	-0.063	0.028
5.450	4.945	0.084	-0.037	0.112	-0.057	-0.084	0.038	-2.946	-0.403	-0.084	0.037	-0.112	0.057	0.084	-0.037
1.817	1.648	0.028	-0.012	0.037	-0.019	-0.028	0.013	-0.982	-0.134	-0.028	0.012	-0.037	0.019	0.028	-0.012
0.396	0.360	0.006	-0.003	0.008	-0.004	-0.006	0.003	-0.214	-0.029	-0.006	0.003	-0.008	0.004	0.006	-0.003
4.063	3.686	0.063	-0.028	0.084	-0.043	-0.063	0.028	-2.196	-0.300	-0.063	0.028	-0.084	0.043	0.063	-0.028
-5.450	-4.945	-0.084	0.037	-0.112	0.057	0.084	-0.038	2.946	0.403	0.084	-0.037	0.112	-0.057	-0.084	0.037
-4.433	-4.022	-0.068	0.030	-0.091	0.047	0.068	-0.030	2.396	0.328	0.068	-0.030	0.091	-0.047	-0.068	0.030
-5.865	-5.322	-0.090	0.040	-0.121	0.062	0.090	-0.040	3.170	0.434	0.090	-0.040	0.121	-0.062	-0.090	0.040
-4.063	-3.686	-0.063	0.028	-0.084	0.043	0.063	-0.028	2.196	0.300	0.063	-0.028	0.084	-0.043	-0.063	0.028
5.450	4.945	0.084	-0.037	0.112	-0.057	-0.084	0.038	-2.946	-0.403	-0.084	0.037	-0.112	0.057	0.084	-0.037
1.817	1.648	0.028	-0.012	0.037	-0.019	-0.029	0.013	-0.982	-0.134	-0.028	0.012	-0.037	0.019	0.028	-0.012
0.396	0.360	0.006	-0.003	0.008	-0.004	-0.006	0.003	-0.214	-0.029	-0.006	0.003	-0.008	0.004	0.006	-0.003
4.063	3.686	0.063	-0.028	0.084	-0.043	-0.063	0.028	-2.196	-0.300	-0.063	0.028	-0.084	0.043	0.063	-0.028
-5.450	-4.945	-0.084	0.037	-0.112	0.057	0.084	-0.038	2.946	0.403	0.084	-0.037	0.112	-0.057	-0.084	0.037

TWO DIMENSIONAL DISCRETE COSINE TRANSFORM

2.031	1.993	0.180	-1.051	-1.095	-0.536	-0.118	0.010	0.041	0.052	0.008	-0.058	-0.082	-0.079	-0.069	-0.029
2.307	2.264	0.204	-1.194	-1.243	-0.609	-0.134	0.011	0.046	0.059	0.009	-0.065	-0.093	-0.089	-0.078	-0.033
0.832	0.817	0.074	-0.431	-0.449	-0.220	-0.048	0.004	0.017	0.021	0.003	-0.024	-0.034	-0.032	-0.028	-0.012
-0.970	-0.952	-0.086	0.502	0.523	0.256	0.056	-0.005	-0.020	-0.025	-0.004	0.028	0.039	0.038	0.033	0.014
-2.390	-2.345	-0.212	1.237	1.288	0.631	0.139	-0.011	-0.048	-0.061	-0.009	0.068	0.097	0.093	0.081	0.034
-2.869	-2.815	-0.254	1.484	1.546	0.757	0.166	-0.014	-0.058	-0.074	-0.011	0.081	0.116	0.111	0.097	0.040
-2.218	-2.176	-0.197	1.147	1.195	0.585	0.129	-0.011	-0.045	-0.057	-0.008	0.063	0.090	0.086	0.075	0.031
-0.693	-0.680	-0.061	0.359	0.374	0.183	0.040	-0.003	-0.014	-0.018	-0.003	0.020	0.028	0.027	0.023	0.010
1.105	1.084	0.098	-0.572	-0.595	-0.292	-0.064	0.005	0.022	0.028	0.004	-0.031	-0.045	-0.043	-0.037	-0.016
2.467	2.421	0.219	-1.277	-1.330	-0.651	-0.143	0.012	0.050	0.063	0.009	-0.070	-0.100	-0.096	-0.084	-0.035
2.858	2.805	0.253	-1.479	-1.540	-0.754	-0.166	0.014	0.058	0.073	0.011	-0.081	-0.116	-0.111	-0.097	-0.040
2.123	2.083	0.188	-1.098	-1.144	-0.560	-0.123	0.010	0.043	0.055	0.008	-0.060	-0.086	-0.082	-0.072	-0.030
0.552	0.541	0.049	-0.285	-0.297	-0.146	-0.032	0.003	0.011	0.014	0.002	-0.016	-0.022	-0.021	-0.019	-0.008
-1.237	-1.214	-0.110	0.640	0.667	0.326	0.072	-0.006	-0.025	-0.032	-0.005	0.035	0.050	0.048	0.042	0.017
-2.538	-2.491	-0.225	1.313	1.368	0.670	0.147	-0.012	-0.051	-0.065	-0.010	0.072	0.103	0.098	0.086	0.036
-2.840	-2.787	-0.252	1.469	1.531	0.750	0.165	-0.013	-0.057	-0.073	-0.011	0.081	0.115	0.110	0.096	0.040

$AGE OBJECT CODE= 12112 BYTES,ARRAY AREA= 110592 BYTES,TOTAL AREA AVAILABLE= 143704 BYTES

Vertical line with shades

```
$JOB          ,NOLIST
$ENTRY
```

INPUT DATA X(K,L)

0.	0.	0.	0.	0.	0.	0.	0.	1.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	2.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	2.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	3.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	4.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	5.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	6.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	7.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	8.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	7.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	6.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	5.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	4.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	3.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	2.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	1.	0.	0.	0.	0.	0.	0.	0.

TWO DIMENSIONAL WALSH HADAMARD TRANSFORM

4.125	4.125	4.125	4.125	4.125	4.125	4.125	4.125	-4.125	-4.125	-4.125	-4.125	-4.125	-4.125	-4.125	-4.1
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.0
0.125	0.125	0.125	0.125	0.125	0.125	0.125	0.125	-0.125	-0.125	-0.125	-0.125	-0.125	-0.125	-0.125	-0.1
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.0
0.125	0.125	0.125	0.125	0.125	0.125	0.125	0.125	-0.125	-0.125	-0.125	-0.125	-0.125	-0.125	-0.125	-0.1
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.0
0.125	0.125	0.125	0.125	0.125	0.125	0.125	0.125	-0.125	-0.125	-0.125	-0.125	-0.125	-0.125	-0.125	-0.12
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.00
-0.375	-0.375	-0.375	-0.375	-0.375	-0.375	-0.375	-0.375	0.375	0.375	0.375	0.375	0.375	0.375	0.375	0.37
-0.500	-0.500	-0.500	-0.500	-0.500	-0.500	-0.500	-0.500	0.500	0.500	0.500	0.500	0.500	0.500	0.500	0.50
-0.875	-0.875	-0.875	-0.875	-0.875	-0.875	-0.875	-0.875	0.875	0.875	0.875	0.875	0.875	0.875	0.875	0.87
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.00
-1.875	-1.875	-1.875	-1.875	-1.875	-1.875	-1.875	-1.875	1.875	1.875	1.875	1.875	1.875	1.875	1.875	1.87
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.00
0.125	0.125	0.125	0.125	0.125	0.125	0.125	0.125	-0.125	-0.125	-0.125	-0.125	-0.125	-0.125	-0.125	-0.12
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.00

TWO DIMENSIONAL HAR TRANSFORM

4.125	4.125	-5.834	0.000	0.000	8.250	0.000	0.000	0.000	0.000	8.250	0.000	0.000	0.000	0.000	0.000
4.125	4.125	-5.834	0.000	0.000	8.250	0.000	0.000	0.000	0.000	8.250	0.000	0.000	0.000	0.000	0.000
-0.530	-0.530	0.750	0.000	0.000	-1.061	0.000	0.000	0.000	0.000	-1.061	0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
-1.750	-1.750	2.475	0.000	0.000	-3.500	0.000	0.000	0.000	0.000	-3.500	0.000	0.000	0.000	0.000	0.000
2.000	2.000	-2.828	0.000	0.000	4.000	0.000	0.000	0.000	0.000	4.000	0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
-0.250	-0.250	0.354	0.000	0.000	-0.500	0.000	0.000	0.000	0.000	-0.500	0.000	0.000	0.000	0.000	0.000
-0.500	-0.500	0.707	0.000	0.000	-1.000	0.000	0.000	0.000	0.000	-1.000	0.000	0.000	0.000	0.000	0.000
0.500	0.500	-0.707	0.000	0.000	1.000	0.000	0.000	0.000	0.000	1.000	0.000	0.000	0.000	0.000	0.000
0.500	0.500	-0.707	0.000	0.000	1.000	0.000	0.000	0.000	0.000	1.000	0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

TWO DIMENSIONAL SLANT TRANSFORM

4.125	3.351	4.125	1.845	5.534	2.818	4.125	1.845	-6.301	7.518	-4.125	-1.845	-5.534	-2.818	-4.125	-1.845
3.452	2.804	3.452	1.544	4.632	2.358	3.452	1.544	-5.273	6.292	-3.452	-1.544	-4.632	-2.358	-3.452	-1.544
0.000	0.000	0.000	-0.000	0.000	0.000	0.000	-0.000	0.000	0.000	0.000	0.000	0.000	-0.000	0.000	0.000
-0.056	-0.045	-0.056	-0.025	-0.075	-0.038	-0.056	-0.025	0.085	-0.102	0.056	0.025	0.075	0.038	0.056	0.025
0.112	0.091	0.112	0.050	0.150	0.076	0.112	0.050	-0.171	0.204	-0.112	-0.050	-0.150	-0.076	-0.112	-0.050
0.037	0.030	0.037	0.016	0.049	0.025	0.037	0.016	-0.056	0.067	-0.037	-0.016	-0.049	-0.025	-0.037	-0.016
0.000	-0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	-0.000	0.000	-0.000	0.000	-0.000
-0.056	-0.045	-0.056	-0.025	-0.075	-0.038	-0.056	-0.025	0.085	-0.102	0.056	0.025	0.075	0.038	0.056	0.025
-2.128	-1.728	-2.128	-0.952	-2.855	-1.453	-2.128	-0.952	3.250	-3.878	2.128	0.952	2.855	1.453	2.128	0.952
0.328	0.267	0.328	0.147	0.441	0.224	0.328	0.147	-0.502	0.599	-0.328	-0.147	-0.441	-0.224	-0.328	-0.147
0.000	0.000	0.000	0.000	0.000	-0.000	0.000	0.000	0.000	0.000	0.000	-0.000	0.000	0.000	0.000	-0.000
-0.056	-0.045	-0.056	-0.025	-0.075	-0.038	-0.056	-0.025	0.085	-0.102	0.056	0.025	0.075	0.038	0.056	0.025
0.112	0.091	0.112	0.050	0.150	0.076	0.112	0.050	-0.171	0.204	-0.112	-0.050	-0.150	-0.076	-0.112	-0.050
0.037	0.030	0.037	0.016	0.049	0.025	0.037	0.016	-0.056	0.067	-0.037	-0.016	-0.049	-0.025	-0.037	-0.016
0.000	0.000	0.000	-0.000	0.000	0.000	0.000	-0.000	0.000	0.000	0.000	0.000	0.000	-0.000	0.000	0.000
-0.056	-0.045	-0.056	-0.025	-0.075	-0.038	-0.056	-0.025	0.085	-0.102	0.056	0.025	0.075	0.038	0.056	0.025

TWO DIMENSIONAL DISCRETE COSINE TRANSFORM

2.062	1.958	-0.288	-2.345	-2.860	-1.495	0.952	2.640	2.692	0.974	-1.384	-2.832	-2.419	-0.415	1.861	2.914
1.892	1.797	-0.264	-2.151	-2.624	-1.372	0.782	2.422	2.470	0.894	-1.269	-2.598	-2.219	-0.381	1.707	2.674
-0.141	-0.134	0.020	0.160	0.195	0.102	-0.058	-0.180	-0.184	-0.066	0.094	0.193	0.165	0.029	-0.127	-0.199
-1.324	-1.257	0.185	1.505	1.836	0.960	-0.547	-1.695	-1.728	-0.626	0.888	1.818	1.553	0.267	-1.195	-1.871
-1.058	-1.004	0.148	1.202	1.466	0.767	-0.437	-1.354	-1.380	-0.500	0.709	1.452	1.240	0.213	-0.954	-1.494
-0.251	-0.239	0.035	0.286	0.349	0.182	-0.104	-0.322	-0.328	-0.119	0.169	0.345	0.295	0.051	-0.227	-0.355
0.147	0.139	-0.020	-0.167	-0.203	-0.106	0.061	0.188	0.192	0.069	-0.098	-0.201	-0.172	-0.030	0.132	0.207
0.116	0.110	-0.016	-0.131	-0.160	-0.084	0.048	0.148	0.151	0.055	-0.078	-0.159	-0.136	-0.023	0.104	0.163
0.058	0.055	-0.008	-0.066	-0.080	-0.042	0.024	0.074	0.075	0.027	-0.039	-0.079	-0.068	-0.012	0.052	0.082
0.063	0.059	-0.009	-0.071	-0.087	-0.045	0.026	0.080	0.082	0.030	-0.042	-0.086	-0.073	-0.013	0.056	0.088
-0.004	-0.004	0.001	0.005	0.006	0.003	-0.002	-0.005	-0.005	-0.002	0.003	0.006	0.005	0.001	-0.004	-0.006
-0.103	-0.098	0.014	0.117	0.143	0.075	-0.043	-0.132	-0.134	-0.049	0.069	0.141	0.121	0.021	-0.093	-0.145
-0.091	-0.086	0.013	0.103	0.126	0.066	-0.037	-0.116	-0.118	-0.043	0.061	0.124	0.106	0.018	-0.082	-0.128
0.005	0.005	-0.001	-0.006	-0.008	-0.004	0.002	0.007	0.007	0.003	-0.004	-0.008	-0.006	-0.001	0.005	0.008
0.048	0.046	-0.007	-0.055	-0.067	-0.035	0.020	0.062	0.063	0.023	-0.032	-0.066	-0.056	-0.010	0.043	0.068
0.021	0.019	-0.003	-0.023	-0.028	-0.015	0.008	0.026	0.027	0.010	-0.014	-0.028	-0.024	-0.004	0.019	0.029

Rectangular enclosure with shades

$JOB .NOLIST

$ENTRY

INPUT DATA X(K,L)

0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	1.	2.	3.	4.	5.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	1.	2.	3.	4.	5.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	1.	2.	3.	4.	5.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	1.	2.	3.	4.	5.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.

TWO DIMENSIONAL WALSH HADAMARD TRANSFORM

4.688	-0.938	-1.563	0.313	4.063	-0.313	-0.938	-0.313	-4.063	0.313	0.938	0.313	-4.688	0.938	1.563	-0.313
-0.938	0.188	0.313	-0.063	-0.813	0.063	0.188	0.063	0.813	-0.063	-0.188	-0.063	0.938	-0.188	-0.313	0.063
0.938	-0.188	-0.313	0.063	0.813	-0.063	-0.188	-0.063	-0.813	0.063	0.188	0.063	-0.938	0.188	0.313	-0.063
-0.938	0.188	0.313	-0.063	-0.813	0.063	0.188	0.063	0.813	-0.063	-0.188	-0.063	0.938	-0.188	-0.313	0.063
-0.938	0.188	0.313	-0.063	-0.813	0.063	0.188	0.063	0.813	-0.063	-0.188	-0.063	0.938	-0.188	-0.313	0.063
0.938	-0.188	-0.313	0.063	0.813	-0.063	-0.188	-0.063	-0.813	0.063	0.188	0.063	-0.938	0.188	0.313	-0.063
2.813	-0.563	-0.938	0.188	2.438	-0.188	-0.563	-0.188	-2.438	0.188	0.563	0.188	-2.813	0.563	0.938	-0.188
0.938	-0.188	-0.313	0.063	0.813	-0.063	-0.188	-0.063	-0.813	0.063	0.188	0.063	-0.938	0.188	0.313	-0.063
0.938	-0.188	-0.313	0.063	0.813	-0.063	-0.188	-0.063	-0.813	0.063	0.188	0.063	-0.938	0.188	0.313	-0.063
-0.938	0.188	0.313	-0.063	-0.813	0.063	0.188	0.063	0.813	-0.063	-0.188	-0.063	0.938	-0.188	-0.313	0.063
-2.813	0.563	0.938	-0.188	-2.438	0.188	0.563	0.188	2.438	-0.188	-0.563	-0.188	2.813	-0.563	-0.938	0.188
-0.938	0.188	0.313	-0.063	-0.813	0.063	0.188	0.063	0.813	-0.063	-0.188	-0.063	0.938	-0.188	-0.313	0.063
-4.688	0.938	1.563	-0.313	-4.063	0.313	0.938	0.313	4.063	-0.313	-0.938	-0.313	4.688	-0.938	-1.563	0.313
0.938	-0.188	-0.313	0.063	0.813	-0.063	-0.188	-0.063	-0.813	0.063	0.188	0.063	-0.938	0.188	0.313	-0.063
-0.938	0.188	0.313	-0.063	-0.813	0.063	0.188	0.063	0.813	-0.063	-0.188	-0.063	0.938	-0.188	-0.313	0.063
0.938	-0.188	-0.313	0.063	0.813	-0.063	-0.188	-0.063	-0.813	0.063	0.188	0.063	-0.938	0.188	0.313	-0.063

TWO DIMENSIONAL HAR TRANSFORM

4.688	4.688	-5.745	0.000	-0.625	8.750	0.000	0.000	0.000	-0.625	-2.500	0.000	0.000	0.000	0.000	0.000
4.688	4.688	-5.745	0.000	-0.625	8.750	0.000	0.000	0.000	-0.625	-2.500	0.000	0.000	0.000	0.000	0.000
1.326	1.326	-1.625	0.000	-0.177	2.475	0.000	0.000	0.000	-0.177	-0.707	0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
-5.625	-5.625	6.894	0.000	0.750	-10.500	0.000	0.000	0.000	0.750	3.000	0.000	0.000	0.000	0.000	0.000
3.750	3.750	-4.596	0.000	-0.500	7.000	0.000	0.000	0.000	-0.500	-2.000	0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
-1.875	-1.875	2.298	0.000	0.250	-3.500	0.000	0.000	0.000	0.250	1.000	0.000	0.000	0.000	0.000	0.000
3.750	3.750	-4.596	0.000	-0.500	7.000	0.000	0.000	0.000	-0.500	-2.000	0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

TWO DIMENSIONAL SLANT TRANSFORM

4.688	2.961	0.313	-0.140	-0.978	-3.568	-0.313	0.140	-3.618	4.327	0.313	-0.140	1.817	3.181	-0.313	0.140
4.315	2.726	0.288	-0.129	-0.901	-3.285	-0.288	0.129	-3.327	3.983	0.288	-0.129	1.673	2.892	-0.288	0.129
-0.938	-0.592	-0.063	0.028	0.196	0.714	0.063	-0.028	0.723	-0.865	-0.063	0.028	-0.363	-0.628	0.063	-0.028
-1.258	-0.795	-0.084	0.037	0.262	0.957	0.084	-0.037	0.970	-1.161	-0.084	0.037	-0.487	-0.843	0.084	-0.037
2.935	1.854	0.196	-0.087	-0.612	-2.234	-0.196	0.087	-2.263	2.709	0.196	-0.087	1.137	1.967	-0.196	0.087
0.823	0.520	0.055	-0.025	-0.172	-0.627	-0.055	0.025	-0.635	0.760	0.055	-0.025	0.319	0.552	-0.055	0.025
0.938	0.592	0.063	-0.028	-0.196	-0.714	-0.063	0.028	-0.723	0.865	0.063	-0.028	0.363	0.628	-0.063	0.028
-0.419	-0.265	-0.028	0.012	0.087	0.319	0.028	-0.012	0.323	-0.387	-0.028	0.012	-0.162	-0.281	0.028	-0.012
-5.524	-3.490	-0.368	0.165	1.153	4.205	0.368	-0.165	4.259	-5.099	-0.368	0.165	-2.141	-3.701	0.368	-0.165
-0.999	-0.631	-0.067	0.030	0.208	0.760	0.067	-0.030	0.770	-0.922	-0.067	0.030	-0.387	-0.669	0.067	-0.030
-0.938	-0.592	-0.063	0.028	0.196	0.714	0.063	-0.028	0.723	-0.865	-0.063	0.028	-0.363	-0.628	0.063	-0.028
0.419	0.265	0.028	-0.012	-0.087	-0.319	-0.028	0.012	-0.323	0.387	0.028	-0.012	0.162	0.281	-0.028	0.012
-0.419	-0.265	-0.028	0.012	0.087	0.319	0.028	-0.012	0.323	-0.387	-0.028	0.012	-0.162	-0.281	0.028	-0.012
-0.274	-0.173	-0.018	0.008	0.057	0.209	0.018	-0.008	0.212	-0.253	-0.018	0.008	-0.106	-0.184	0.018	-0.008
0.938	0.592	0.063	-0.028	-0.196	-0.714	-0.063	0.028	-0.723	0.865	0.063	-0.028	0.363	0.628	-0.063	0.028
1.258	0.795	0.084	-0.037	-0.262	-0.957	-0.084	0.037	-0.970	1.161	0.084	-0.037	0.487	0.843	-0.084	0.037

TWO DIMENSIONAL DISCRETE COSINE TRANSFORM

2.344	1.781	-1.338	-3.066	-1.908	0.821	2.432	1.608	-0.490	-1.678	-0.998	0.469	1.058	0.336	-0.635	-0.677
2.432	1.848	-1.388	-3.181	-1.976	0.852	2.523	1.668	-0.518	-1.781	-1.036	0.486	1.098	0.388	-0.659	-0.702
0.311	0.236	-0.177	-0.406	-0.252	0.109	0.322	0.213	-0.066	-0.222	-0.132	0.042	0.140	0.044	-0.088	-0.090
-1.809	-1.375	1.030	2.367	1.470	-0.634	-1.877	-1.238	0.385	1.295	0.771	-0.362	-0.817	-0.259	0.490	0.523
-2.771	-2.106	1.577	3.625	2.252	-0.971	-2.875	-1.897	0.590	1.984	1.181	-0.554	-1.251	-0.397	0.751	0.800
-2.201	-1.673	1.253	2.879	1.788	-0.771	-2.283	-1.506	0.469	1.575	0.938	-0.440	-0.994	-0.315	0.596	0.636
-0.656	-0.498	0.373	0.858	0.533	-0.230	-0.680	-0.449	0.140	0.469	0.279	-0.131	-0.296	-0.094	0.178	0.190
0.835	0.635	-0.476	-1.093	-0.679	0.293	0.867	0.572	-0.178	-0.598	-0.356	0.167	0.377	0.120	-0.226	-0.241
1.479	1.124	-0.842	-1.935	-1.202	0.518	1.534	1.012	-0.315	-1.059	-0.630	0.296	0.668	0.212	-0.401	-0.427
1.170	0.889	-0.666	-1.531	-0.951	0.410	1.218	0.801	-0.249	-0.838	-0.498	0.234	0.528	0.168	-0.317	-0.338
0.418	0.317	-0.238	-0.546	-0.339	0.146	0.433	0.286	-0.089	-0.299	-0.178	0.083	0.189	0.060	-0.113	-0.121
-0.135	-0.103	0.077	0.177	0.110	-0.047	-0.140	-0.093	0.029	0.097	0.058	-0.027	-0.061	-0.019	0.037	0.039
-0.193	-0.147	0.110	0.252	0.157	-0.068	-0.200	-0.132	0.041	0.138	0.082	-0.039	-0.087	-0.028	0.052	0.056
0.055	0.042	-0.032	-0.072	-0.045	0.019	0.057	0.038	-0.012	-0.040	-0.024	0.011	0.025	0.008	-0.015	-0.016
0.192	0.146	-0.109	-0.251	-0.156	0.067	0.199	0.132	-0.041	-0.138	-0.082	0.039	0.087	0.028	-0.052	-0.056
-0.027	-0.021	0.015	0.036	0.022	-0.010	-0.028	-0.019	0.006	0.019	0.012	-0.005	-0.012	-0.004	0.007	0.008

USAGE OBJECT CODE= 12112 BYTES,ARRAY AREA= 110592 BYTES,TOTAL AREA AVAILABLE= 123708 BYTES

NOSTICS NUMBER OF ERRORS= 0, NUMBER OF WARNINGS= 0, NUMBER OF EXTENSIONS=

Slant edge @ 135°

```
      $JOB            ,NOLIST

      SENTRY
INPUT DATA X(K,L)

 1. 3. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 3. 1. 3. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 3. 1. 3. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 3. 1. 3. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 3. 1. 3. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 3. 1. 3. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 3. 1. 3. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 3. 1. 3. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 3. 1. 3. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 3. 1. 3. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 3. 1. 3. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 3. 1. 3. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 3. 1. 3. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 3. 1. 3. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 3. 1. 3.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 3. 1.
TWO DIMENSIONAL WALSH HADAMARD TRANSFORM

 6.625  0.000  0.000 -0.375  0.000 -0.375 -0.375  0.000  0.000 -0.375 -0.375  0.000 -0.375  0.000  0.000 -0.375
 0.000 -4.625  0.375 -0.000  0.375  0.000  0.000  0.375  0.375  0.000  0.000  0.375  0.000  0.375  0.375  0.000
 0.000  0.375  1.375  0.000  0.375  0.000  0.000  0.375  0.375  0.000  0.000  0.375  0.000  0.375  0.375  0.000
-0.375  0.000  0.000  0.625  0.000 -0.375 -0.375  0.000  0.000 -0.375 -0.375  0.000 -0.375  0.000  0.000 -0.375
 0.000  0.375  0.375  0.000  4.375  0.000  0.000 -2.625  0.375  0.000  0.000  0.375  0.000  0.375  0.375  0.000
-0.375  0.000  0.000 -0.375  0.000 -2.375  2.625  0.000  0.000 -0.375 -0.375  0.000 -0.375  0.000  0.000 -0.375
-0.375  0.000  0.000 -0.375  0.000  2.625  3.625  0.000  0.000 -0.375 -0.375  0.000 -0.375  0.000  0.000 -0.375
 0.000  0.375  0.375  0.000 -2.625  0.000  0.000 -1.625  0.375  0.000  0.000  0.375  0.000  0.375  0.375  0.000
 0.000  0.375  0.375  0.000  0.375  0.000  0.000  0.375  5.875  0.000  0.000 -1.125  0.000 -1.125 -1.125  0.000
-0.375  0.000  0.000 -0.375  0.000 -0.375 -0.375  0.000  0.000 -3.875  1.125  0.000  1.125  0.000  0.000  1.125
-0.375  0.000  0.000 -0.375  0.000 -0.375 -0.375  0.000  0.000  1.125  2.125  0.000  1.125  0.000  0.000  1.125
 0.000  0.375  0.375  0.000  0.375  0.000  0.000  0.375 -1.125  0.000  0.000 -0.125  0.000 -1.125 -1.125  0.000
-0.375  0.000  0.000 -0.375  0.000 -0.375 -0.375  0.000  0.000  1.125  1.125  0.000  5.125  0.000  0.000 -1.875
 0.000  0.375  0.375  0.000  0.375  0.000  0.000  0.375 -1.125  0.000  0.000 -1.125  0.000 -3.125  1.875  0.000
 0.000  0.375  0.375  0.000  0.375  0.000  0.000  0.375 -1.125  0.000  0.000 -1.125  0.000  1.875  2.875  0.000
-0.375  0.000  0.000 -0.375  0.000 -0.375 -0.375  0.000  0.000  1.125  1.125  0.000 -1.875  0.000  0.000 -0.875
TWO DIMENSIONAL HAR TRANSFORM

 6.625  6.625 -0.000  0.000 -0.375  0.375  0.000  0.000 -0.375  0.000  0.000  0.375  0.000  0.000  0.000  0.000
 6.625  6.625 -0.000  0.000 -0.375  0.375  0.000  0.000 -0.375  0.000  0.000  0.375  0.000  0.000  0.000  0.000
-0.000 -0.000 11.750  0.000 -0.530  0.530  0.000  0.000 -0.530  1.061  1.061 -0.530  0.000  0.000  0.000  0.000
 0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
-0.375 -0.375 -0.530  0.000  9.500 -0.750  0.000  0.000  0.750  0.750 -0.750  0.000  0.000  0.000  0.000  0.000
 0.375  0.375  0.530  0.000 -0.750  9.500  0.000  0.000  0.000 -0.750  0.750  0.750  0.000  0.000  0.000  0.000

 0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
 0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
-0.375 -0.375 -0.530  0.000  0.750  0.000  0.000  0.000  2.500 -0.750  0.000  0.000  0.000  0.000  0.000  0.000
 0.000  0.000  1.061  0.000 -0.750  0.750  0.000  0.000 -0.750  2.500 -0.750  0.000  0.000  0.000  0.000  0.000
 0.000  0.000  1.061  0.000 -0.750  0.750  0.000  0.000  0.000 -0.750  2.500 -0.750  0.000  0.000  0.000  0.000
 0.375  0.375 -0.530  0.000  0.000  0.750  0.000  0.000  0.000  0.000 -0.750  2.500  0.000  0.000  0.000  0.000
 0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
 0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
 0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
 0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
TWO DIMENSIONAL SLANT TRANSFORM

 4.625  5.780 -0.375  0.000 -0.503 -0.000  0.000 -0.148 -0.573  0.000  0.000 -0.148  0.000 -0.256 -0.375  0.000
 5.780  6.438 -0.325 -0.154 -0.436 -0.236 -0.345 -0.195 -0.894 -0.286 -0.345 -0.195 -0.867 -0.222 -0.325 -0.154
-0.375 -0.325  0.625  0.000 -0.503  0.000  0.000 -0.148 -0.573 -0.000  0.000 -0.148  0.000 -0.256 -0.375  0.000
 0.000 -0.154  0.000 -3.725  0.000 -1.980  0.000  0.168  0.000  0.000 -0.877  0.168  0.000  0.225  0.000 -0.000  0.075
-0.503 -0.436 -0.503  0.000  8.525 -0.000  0.000 -0.825 -0.769 -0.000  0.000 -0.225 -0.000 -0.344 -0.503  0.000
-0.000 -0.236  0.000 -1.980 -0.000  0.975  1.720 -0.000  0.000 -1.380  0.254  0.000  0.748 -0.000 -0.000  0.115
 0.000 -0.345  0.000  0.168  0.000  1.720 -1.625 -0.000  0.000 -1.942  0.375  0.000  0.507 -0.000  0.000  0.168
-0.148 -0.195 -0.148  0.000 -0.825 -0.000 -0.254  3.275 -0.254  0.000  0.000 -0.075  0.000 -0.115 -0.168  0.000
-0.573 -0.894 -0.573  0.000 -0.769  0.000  0.000 -0.254  5.425 -0.000  0.000 -0.403  0.000 -0.615 -0.900  0.000
 0.000 -0.286  0.000 -0.877 -0.000 -1.380 -1.942  0.000 -0.000  3.875  1.054  0.000  1.817  0.000  0.000  0.472
 0.000 -0.345  0.000  0.168  0.000  0.254  0.375  0.000  0.000  1.054 -0.125 -0.000 -1.509  0.000  0.000 -0.503
-0.168 -0.195 -0.168  0.000 -0.225  0.000 -0.000 -0.075 -0.403  0.000 -0.000 -3.575 -0.000 -1.751  0.503  0.000
 0.000 -0.443  0.000  0.225 -0.000  0.384  0.503  0.000  0.000  1.817 -1.509 -0.000  3.179 -0.000  0.000 -1.275
-0.256 -0.222 -0.256  0.000 -0.344 -0.000 -0.000 -0.115 -0.615 -0.000  0.000 -1.751 -0.000  1.325  2.232  0.000
-0.375 -0.325 -0.375  0.000 -0.503 -0.000  0.000 -0.168 -0.900  0.000  0.000  0.503  0.000  2.232 -0.875 -0.000
 0.000 -0.154  0.000  0.075  0.000  0.115  0.168  0.000  0.000  0.472 -0.503  0.000 -1.275  0.000 -0.000 -3.825
TWO DIMENSIONAL DISCRETE COSINE TRANSFORM

 3.312  3.012 -0.002 -1.146 -0.258  0.476 -0.002 -0.540 -0.243  0.186 -0.002 -0.347 -0.219  0.075 -0.001 -0.200
 3.012  3.298  1.281 -0.147 -0.468 -0.179 -0.003 -0.179 -0.233 -0.145 -0.131 -0.165 -0.177 -0.185 -0.131 -0.195
-0.002  1.281  3.072  2.422 -0.003 -1.198 -0.358  0.456 -0.003 -0.618 -0.127  0.177 -0.003 -0.409 -0.286  0.063
-1.146 -0.147  2.422  3.190  1.428 -0.187 -0.622 -0.145  0.031 -0.179 -0.275 -0.185 -0.088 -0.145 -0.184 -0.110
-0.258 -0.468 -0.003  1.428  2.012  2.009 -0.003 -1.007 -0.337  0.248 -0.002 -0.520  0.308  0.121 -0.002 -0.340
 0.476 -0.179 -1.198 -0.187  2.009  2.979  1.593 -0.187 -0.430 -0.185  0.057 -0.179 -0.272 -0.119 -0.060 -0.145
-0.002 -0.003 -0.358 -0.622 -0.003  1.593  2.652  1.934 -0.002 -0.843 -0.314  0.246 -0.002 -0.433 -0.274  0.074
-0.540 -0.179  0.456 -0.145 -1.007 -0.187  1.934  2.672  1.841 -0.184 -0.580 -0.119  0.057 -0.179 -0.245 -0.088
-0.243 -0.233 -0.003  0.031 -0.337 -0.430 -0.002  1.841  2.302  1.589 -0.002 -0.721 -0.284  0.191 -0.002 -0.387
 0.186 -0.145 -0.618 -0.179  0.248 -0.185 -0.843 -0.184  1.589  2.215  1.275 -0.232 -0.002 -0.572 -0.258  0.114
-0.002 -0.131 -0.127 -0.275 -0.002  0.057 -0.314 -0.580 -0.002  1.275  1.232 -0.002 -0.572 -0.258  0.114
-0.347 -0.165  0.177 -0.185 -0.520 -0.179  0.246 -0.119 -0.721 -0.284  0.191 -0.232  0.937 -0.186 -0.349 -0.055
-0.219 -0.177 -0.002 -0.088  0.308 -0.272 -0.002  0.057 -0.284 -0.497 -0.002  0.937  1.188  0.496 -0.001 -0.416
 0.075 -0.185 -0.409 -0.145  0.121 -0.119 -0.433 -0.179  0.191 -0.088 -0.572 -0.186  0.496  1.318  0.423 -0.184
-0.001 -0.131 -0.286 -0.184 -0.002 -0.060 -0.274 -0.285 -0.002  0.032 -0.258 -0.392 -0.001  0.423  0.860 -0.521
-0.200 -0.195  0.063 -0.119 -0.340 -0.145  0.076 -0.088 -0.387 -0.179  0.114 -0.055 -0.416 -0.184 -0.521  0.774

USAGE       OBJECT CODE=   12112 BYTES, ARRAY AREA=  110592 BYTES, TOTAL AREA AVAILABLE=  183708  BYTES

NOSTICS          NUMBER OF ERRORS=       0, NUMBER OF WARNINGS=       0, NUMBER OF EXTENSIONS=       0
```

Rectangular enclosure

```
$JOB              ,NOLIST
$ENTRY
```

INPUT DATA X(K,L)

0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	1.	1.	1.	1.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	1.	1.	1.	1.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	1.	1.	1.	1.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	1.	1.	1.	1.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	1.	1.	1.	1.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	1.	1.	1.	1.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.

TWO DIMENSIONAL WALSH HADAMARD TRANSFORM

1.500	0.000	0.000	0.000	0.750	0.750	0.750	-0.750	-0.750	-0.750	-0.750	0.750	-1.500	0.000	0.000	0.00
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.00
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.00
-0.500	0.000	0.000	0.000	-0.250	-0.250	-0.250	0.250	0.250	0.250	0.250	-0.250	0.500	0.000	0.000	0.00
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.00
0.500	0.000	0.000	0.000	0.250	0.250	0.250	-0.250	-0.250	-0.250	-0.250	0.250	-0.500	0.000	0.000	0.00
0.500	0.000	0.000	0.000	0.250	0.250	0.250	-0.250	-0.250	-0.250	-0.250	0.250	-0.500	0.000	0.000	0.00
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.00
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.00
-0.500	0.000	0.000	0.000	-0.250	-0.250	-0.250	0.250	0.250	0.250	0.250	-0.250	0.500	0.000	0.000	0.00
-0.500	0.000	0.000	0.000	-0.250	-0.250	-0.250	0.250	0.250	0.250	0.250	-0.250	0.500	0.000	0.000	0.00
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.00
-1.500	0.000	0.000	0.000	-0.750	-0.750	-0.750	0.750	0.750	0.750	0.750	-0.750	1.500	0.000	0.000	0.00
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.00
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.00
0.500	0.000	0.000	0.000	0.250	0.250	0.250	-0.250	-0.250	-0.250	-0.250	0.250	-0.500	0.000	0.000	0.00

TWO DIMENSIONAL HAR TRANSFORM

1.500	1.500	-1.061	0.000	-0.750	2.250	0.000	0.000	0.000	-0.750	0.750	0.000	0.000	0.000	0.000	0.00
1.500	1.500	-1.061	0.000	-0.750	2.250	0.000	0.000	0.000	-0.750	0.750	0.000	0.000	0.000	0.000	0.00
-0.000	-0.000	0.000	0.000	0.000	-0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.00
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.00
-1.500	-1.500	1.061	0.000	0.750	-2.250	0.000	0.000	0.000	0.750	-0.750	0.000	0.000	0.000	0.000	0.00
1.500	1.500	-1.061	0.000	-0.750	2.250	0.000	0.000	0.000	-0.750	0.750	0.000	0.000	0.000	0.000	0.00
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.00
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.00
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.00
-0.500	-0.500	0.354	0.000	0.250	-0.750	0.000	0.000	0.000	0.250	-0.250	0.000	0.000	0.000	0.000	0.00
0.500	0.500	-0.354	0.000	-0.250	0.750	0.000	0.000	0.000	-0.250	0.250	0.000	0.000	0.000	0.000	0.00
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.00
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.00
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.00
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.00
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.00

TWO DIMENSIONAL SLANT TRANSFORM

1.500	1.137	0.000	0.000	1.006	-0.366	-0.750	0.335	-1.800	0.941	0.750	-0.335	0.000	-0.146	0.000	0.00
1.300	0.985	0.000	0.000	0.872	-0.317	-0.650	0.291	-1.560	0.815	0.650	-0.291	0.000	-0.127	0.000	0.00
-0.500	-0.379	0.000	0.000	-0.335	0.122	0.250	-0.112	0.600	-0.314	-0.250	0.112	0.000	0.049	0.000	0.00
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.00
0.671	0.509	0.000	0.000	0.450	-0.164	-0.335	0.150	-0.805	0.421	0.335	-0.150	0.000	-0.065	0.000	0.00
-0.000	-0.000	0.000	0.000	-0.000	0.000	0.000	0.000	0.000	-0.000	-0.000	0.000	0.000	0.000	0.000	0.00
0.000	0.000	0.000	0.000	0.000	0.000	0.000	-0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.00
0.224	0.170	0.000	0.000	0.150	-0.055	-0.112	0.050	-0.268	0.140	0.112	-0.050	0.000	-0.022	0.000	0.00
-1.637	-1.241	0.000	0.000	-1.099	0.399	0.818	-0.366	1.964	-1.027	-0.818	0.366	0.000	0.160	0.000	0.00
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	-0.000	0.000	0.000	-0.000	0.000	-0.000	0.000	0.00
0.000	0.000	0.000	0.000	0.000	-0.000	0.000	0.000	0.000	-0.000	0.000	-0.000	0.000	0.000	0.000	0.00
-0.224	-0.170	0.000	0.000	-0.150	0.055	0.112	-0.050	0.268	-0.140	-0.112	0.050	0.000	0.022	0.000	0.00
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.00
0.146	0.111	0.000	0.000	0.098	-0.036	-0.073	0.033	-0.176	0.092	0.073	-0.033	0.000	-0.014	0.000	0.00
0.500	0.379	0.000	0.000	0.335	-0.122	-0.250	0.112	-0.600	0.314	0.250	-0.112	0.000	-0.049	0.000	0.00
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.00

TWO DIMENSIONAL DISCRETE COSINE TRANSFORM

0.750	0.669	-0.203	-0.886	-0.887	-0.262	0.470	0.768	0.488	-0.042	-0.433	-0.410	-0.128	0.117	0.157	0.05
0.739	0.659	-0.200	-0.873	-0.875	-0.258	0.464	0.757	0.481	-0.041	-0.427	-0.404	-0.126	0.116	0.154	0.05
-0.001	-0.001	0.000	0.001	0.001	0.000	-0.000	-0.001	-0.000	0.000	0.000	0.000	0.000	-0.000	-0.000	-0.00
-0.659	-0.587	0.178	0.779	0.780	0.230	-0.413	-0.675	-0.429	0.054	0.380	0.360	0.112	-0.103	-0.138	-0.05
-0.837	-0.746	0.226	0.989	0.990	0.292	-0.525	-0.857	-0.545	0.049	0.483	0.457	0.142	-0.131	-0.175	-0.06
-0.511	-0.455	0.138	0.604	0.605	0.178	-0.320	-0.523	-0.333	0.042	0.295	0.279	0.087	-0.080	-0.107	-0.03
0.001	0.001	-0.000	-0.001	-0.001	-0.000	0.001	0.001	0.001	-0.000	-0.001	-0.001	-0.000	0.000	0.000	0.00
0.328	0.292	-0.089	-0.387	-0.388	-0.114	0.205	0.336	0.213	-0.027	-0.189	-0.179	-0.056	0.051	0.068	0.02
0.326	0.291	-0.088	-0.385	-0.386	-0.114	0.205	0.334	0.212	-0.027	-0.188	-0.178	-0.056	0.051	0.068	0.02
0.137	0.122	-0.037	-0.162	-0.162	-0.048	0.086	0.140	0.089	-0.011	-0.079	-0.075	-0.023	0.021	0.029	0.01
-0.000	-0.000	0.000	0.000	0.000	-0.000	-0.000	-0.000	-0.000	0.000	0.000	0.000	0.000	-0.000	-0.000	-0.00
0.024	0.022	-0.007	-0.029	-0.029	-0.008	0.015	0.025	0.016	-0.002	-0.014	-0.013	-0.004	0.004	0.005	0.00
0.122	0.109	-0.033	-0.144	-0.145	-0.043	0.077	0.125	0.080	-0.010	-0.070	-0.067	-0.021	0.019	0.025	0.01
0.133	0.118	-0.036	-0.157	-0.157	-0.046	0.083	0.136	0.086	-0.011	-0.077	-0.072	-0.023	0.021	0.028	0.01
-0.001	-0.001	0.000	0.001	0.001	0.000	-0.001	-0.001	-0.001	0.000	0.001	0.001	0.000	-0.000	-0.000	-0.00
-0.179	-0.160	0.048	0.211	0.212	0.062	-0.112	-0.183	-0.117	0.015	0.103	0.098	0.030	-0.028	-0.037	-0.01

```
USAGE      OBJECT CODE=  12112 BYTES,ARRAY AREA=  110592 BYTES,TOTAL AREA AVAILABLE=  183704 BYTES
DSTICS       NUMBER OF ERRORS=      0, NUMBER OF WARNINGS=      0, NUMBER OF EXTENSIONS=
```

A circle @ the centre

```
      $JOB                ,NOLIST

      $ENTRY
INPUT DATA X(K,L)

 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 1. 1. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 1. 1. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
TWO DIMENSIONAL WALSH HADAMARD TRANSFORM

 0.250  0.000  0.125  0.125  0.125  0.125  0.250  0.000 -0.125 -0.125 -0.250  0.000 -0.250  0.000 -0.125 -0.125
 0.000 -0.250 -0.125 -0.125 -0.125 -0.125  0.000 -0.250  0.125  0.125  0.000  0.250  0.000  0.250  0.125  0.125
 0.125 -0.125  0.000  0.000  0.000  0.000  0.125 -0.125  0.000  0.000 -0.125  0.125 -0.125  0.125  0.000  0.000
 0.125 -0.125  0.000  0.000  0.000  0.000  0.125 -0.125  0.000  0.000 -0.125  0.125 -0.125  0.125  0.000  0.000
 0.125 -0.125  0.000  0.000  0.000  0.000  0.125 -0.125  0.000  0.000 -0.125  0.125 -0.125  0.125  0.000  0.000
 0.125 -0.125  0.000  0.000  0.000  0.000  0.125 -0.125  0.000  0.000 -0.125  0.125 -0.125  0.125  0.000  0.000
 0.250  0.000  0.125  0.125  0.125  0.125  0.250  0.000 -0.125 -0.125 -0.250  0.000 -0.250  0.000 -0.125 -0.125
 0.000 -0.250 -0.125 -0.125 -0.125 -0.125  0.000 -0.250  0.125  0.125  0.000  0.250  0.000  0.250  0.125  0.125
-0.125  0.125  0.000  0.000  0.000  0.000 -0.125  0.125  0.000  0.000  0.125 -0.125  0.125 -0.125  0.000  0.000
-0.125  0.125  0.000  0.000  0.000  0.000 -0.125  0.125  0.000  0.000  0.125 -0.125  0.125 -0.125  0.000  0.000
-0.250  0.000 -0.125 -0.125 -0.125 -0.125 -0.250  0.000  0.125  0.125  0.250  0.000  0.250  0.000  0.125  0.125
 0.000  0.250  0.125  0.125  0.125  0.125  0.000  0.250 -0.125 -0.125  0.000 -0.250  0.000 -0.250 -0.125 -0.125
-0.250  0.000 -0.125 -0.125 -0.125 -0.125 -0.250  0.000  0.125  0.125  0.250  0.000  0.250  0.000  0.125  0.125
 0.000  0.250  0.125  0.125  0.125  0.125  0.000  0.250 -0.125 -0.125  0.000 -0.250  0.000 -0.250 -0.125 -0.125
-0.125  0.125  0.000  0.000  0.000  0.000 -0.125  0.125  0.000  0.000  0.125 -0.125  0.125 -0.125  0.000  0.000
-0.125  0.125  0.000  0.000  0.000  0.000 -0.125  0.125  0.000  0.000  0.125 -0.125  0.125 -0.125  0.000  0.000
TWO DIMENSIONAL HAR TRANSFORM

 0.250  0.250 -0.177  0.000 -0.125  0.375  0.000  0.000  0.000 -0.125  0.375  0.000  0.000  0.000  0.000  0.000
 0.250  0.250 -0.177  0.000 -0.125  0.375  0.000  0.000  0.000 -0.125  0.375  0.000  0.000  0.000  0.000  0.000
-0.177 -0.177  0.000  0.000  0.177 -0.177  0.000  0.000  0.000  0.177 -0.177  0.000  0.000  0.000  0.000  0.000
 0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
-0.125 -0.125  0.177  0.000  0.000 -0.250  0.000  0.000  0.000  0.000 -0.250  0.000  0.000  0.000  0.000  0.000
 0.375  0.375 -0.177  0.000 -0.250  0.500  0.000  0.000  0.000 -0.250  0.500  0.000  0.000  0.000  0.000  0.000

 0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
 0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
 0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
-0.125 -0.125  0.177  0.000  0.000 -0.250  0.000  0.000  0.000  0.000 -0.250  0.000  0.000  0.000  0.000  0.000
 0.375  0.375 -0.177  0.000 -0.250  0.500  0.000  0.000  0.000 -0.250  0.500  0.000  0.000  0.000  0.000  0.000
 0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
 0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
 0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
 0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
 0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
TWO DIMENSIONAL SLANT TRANSFORM

 0.250  0.203  0.125 -0.056  0.280  0.037  0.000  0.000 -0.355  0.208  0.000 -0.000 -0.112 -0.122 -0.125  0.056
 0.203  0.145  0.102 -0.045  0.227  0.030  0.000  0.000 -0.288  0.166 -0.000 -0.000 -0.091 -0.099 -0.102  0.045
 0.125  0.102  0.000 -0.112  0.112 -0.089 -0.125 -0.056 -0.168 -0.024  0.125  0.056  0.056 -0.037  0.000 -0.112
-0.056 -0.045 -0.112 -0.100 -0.100 -0.094 -0.168 -0.075  0.094 -0.214 -0.168 -0.075  0.175  0.040  0.112 -0.100
 0.280  0.227  0.112 -0.100  0.300  0.011 -0.056 -0.025 -0.390  0.172  0.056  0.025 -0.075 -0.125 -0.112  0.100
 0.037  0.030 -0.000 -0.094  0.011 -0.047 -0.134 -0.040 -0.037 -0.105  0.134  0.040  0.104 -0.008 -0.000 -0.094
 0.000 -0.000 -0.125 -0.168 -0.056 -0.134 -0.250 -0.112  0.027 -0.251  0.250  0.112  0.223  0.089  0.125 -0.168
 0.000 -0.000 -0.056 -0.075 -0.025 -0.040 -0.112 -0.050  0.012 -0.112  0.112  0.050  0.100  0.022  0.056 -0.075
-0.355 -0.288 -0.168 -0.094 -0.390 -0.037 -0.027  0.012  0.500 -0.242 -0.027 -0.012  0.134  0.148  0.168 -0.094
 0.208  0.166 -0.024 -0.214  0.172 -0.105 -0.251 -0.112 -0.242  0.262 -0.251 -0.112  0.134 -0.051  0.024 -0.214
 0.000 -0.000  0.125  0.168  0.056  0.134  0.250  0.112 -0.027  0.251 -0.250 -0.112 -0.223 -0.089 -0.125  0.168
-0.000 -0.000  0.056  0.075  0.025  0.040  0.112  0.050 -0.012  0.112 -0.112 -0.050 -0.100 -0.022 -0.056  0.075
-0.112 -0.091  0.056  0.175 -0.075  0.104  0.223  0.100  0.134 -0.134 -0.223 -0.100  0.150  0.011 -0.056  0.175
-0.122 -0.099 -0.037 -0.040 -0.125  0.008  0.089  0.040 -0.148  0.051  0.089  0.040 -0.011  0.050  0.037 -0.040
-0.125 -0.102  0.000  0.112 -0.112  0.089  0.125  0.056 -0.168  0.024 -0.125 -0.056  0.056  0.037  0.000 -0.112
 0.056  0.045 -0.112 -0.100  0.100 -0.098  0.168  0.075 -0.098  0.214 -0.168 -0.075  0.175 -0.040 -0.112 -0.100
TWO DIMENSIONAL DISCRETE COSINE TRANSFORM

 0.125  0.118 -0.017 -0.139 -0.167 -0.085  0.047  0.182  0.139  0.088 -0.065 -0.126 -0.101 -0.016  0.067  0.097
 0.118  0.112 -0.016 -0.132 -0.158 -0.081  0.045  0.134  0.132  0.046 -0.062 -0.119 -0.096 -0.015  0.064  0.092
-0.017 -0.016  0.002  0.019  0.023  0.012 -0.007 -0.020 -0.019 -0.007  0.009  0.017  0.014  0.002 -0.009 -0.013
-0.139 -0.132  0.019  0.155  0.185  0.095 -0.053 -0.157 -0.154 -0.053  0.072  0.139  0.112  0.018 -0.074 -0.106
-0.167 -0.158  0.023  0.185  0.222  0.113 -0.063 -0.188 -0.184 -0.064  0.086  0.166  0.133  0.021 -0.087 -0.125
-0.085 -0.081  0.012  0.095  0.113  0.058 -0.032 -0.096 -0.094 -0.032  0.044  0.085  0.068  0.011 -0.044 -0.063
 0.047  0.045 -0.007 -0.053 -0.063 -0.032  0.018  0.053  0.052  0.018 -0.024 -0.048 -0.037 -0.006  0.024  0.034
 0.182  0.134 -0.020 -0.157 -0.188 -0.096  0.053  0.159  0.156  0.053 -0.073 -0.141 -0.108 -0.017  0.070  0.098
 0.139  0.132 -0.019 -0.154 -0.184 -0.094  0.052  0.156  0.151  0.052 -0.069 -0.132 -0.106 -0.016  0.066  0.093
 0.088  0.046 -0.007 -0.053 -0.064 -0.032  0.018  0.053  0.052  0.018 -0.024 -0.048 -0.037 -0.006  0.024  0.034
-0.065 -0.062  0.009  0.072  0.086  0.044 -0.024 -0.071 -0.069 -0.024  0.031  0.059  0.044  0.007 -0.028 -0.039
-0.126 -0.119  0.017  0.139  0.166  0.085 -0.046 -0.137 -0.132 -0.045  0.059  0.111  0.084  0.013 -0.053 -0.074
-0.101 -0.096  0.014  0.112  0.133  0.068 -0.037 -0.108 -0.104 -0.035  0.044  0.086  0.064  0.010 -0.038 -0.050
-0.016 -0.015  0.002  0.018  0.021  0.011 -0.006 -0.017 -0.016 -0.005  0.007  0.013  0.010  0.001 -0.006 -0.007
 0.067  0.064 -0.009 -0.074 -0.087 -0.044  0.024  0.070  0.068  0.022 -0.028 -0.052 -0.038 -0.006  0.024  0.023
 0.097  0.092 -0.013 -0.106 -0.125 -0.063  0.034  0.098  0.097  0.031 -0.039 -0.069 -0.050 -0.007  0.023  0.024

$AGE        OBJECT CODE= 12112 BYTES,ARRAY AREA= 110592 BYTES,TOTAL AREA AVAILABLE= 183708 BYTES

$TICS       NUMBER OF ERRORS= 0, NUMBER OF WARNINGS= 0, NUMBER OF EXTENSIONS= 0
```

4 Circles @ equal distance

```
      $JOB              ,NOLIST

      $ENTRY
INPUT DATA X(K,L)
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0.
 0. 1. 0. 1. 0. 0. 0. 0. 0. 1. 0. 1. 0. 0. 0. 0.
 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0.
 0. 1. 0. 1. 0. 0. 0. 0. 0. 1. 0. 1. 0. 0. 0. 0.
 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
TWO DIMENSIONAL WALSH HADAMARD TRANSFORM

 1.000  0.000  0.000  0.500  0.000 -0.500 -1.000  0.000  0.000  0.000 -0.500  0.000  0.500  0.000  0.000  0.000
 0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
 0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
 0.500  0.000  0.000  0.000  0.000  0.000 -0.500  0.000  0.000  0.500  0.000  0.000  0.000  0.000  0.000 -0.500
 0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
-0.500  0.000  0.000  0.000  0.000  0.000  0.500  0.000  0.000 -0.500  0.000  0.000  0.000  0.000  0.000  0.500
-1.000  0.000  0.000 -0.500  0.000  0.500  1.000  0.000  0.000  0.000  0.500  0.000 -0.500  0.000  0.000  0.000
 0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
 0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
 0.000  0.000  0.000 -0.500  0.000  0.500  0.000  0.000  0.000  1.000  0.500  0.000 -0.500  0.000  0.000 -1.000
 0.500  0.000  0.000  0.000  0.000  0.000 -0.500  0.000  0.000  0.500  0.000  0.000  0.000  0.000  0.000 -0.500
 0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
-0.500  0.000  0.000  0.000  0.000  0.000  0.500  0.000  0.000 -0.500  0.000  0.000  0.000  0.000  0.000  0.500
 0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
 0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
 0.000  0.000  0.000  0.500  0.000 -0.500  0.000  0.000  0.000 -1.000 -0.500  0.000  0.500  0.000  0.000  1.000
TWO DIMENSIONAL HAR TRANSFORM

 1.000  1.000  0.000  0.000  0.500 -0.500  0.000  0.000 -0.750  0.250 -0.250  0.750  0.000  0.000  0.000  0.000
 1.000  1.000  0.000  0.000  0.500 -0.500  0.000  0.000 -0.750  0.250 -0.250  0.750  0.000  0.000  0.000  0.000
 0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
 0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
 0.500 -0.500  0.000  0.000  0.000  0.000  0.000  0.000 -0.250 -0.250  0.250 -0.250  0.000  0.000  0.000  0.000
 0.500  0.500  0.000  0.000  0.000  0.000  0.000  0.000 -0.250  0.250 -0.250  0.250  0.000  0.000  0.000  0.000

 0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
 0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
 0.250 -0.250  0.000  0.000 -0.250  0.250  0.000  0.000  0.250  0.000  0.000 -0.250  0.000  0.000  0.000  0.000
 0.750 -0.750  0.000  0.000 -0.250  0.250  0.000  0.000 -0.500  0.250 -0.250  0.500  0.000  0.000  0.000  0.000
 0.750 -0.750  0.000  0.000 -0.250  0.250  0.000  0.000  0.500 -0.250  0.250 -0.500  0.000  0.000  0.000  0.000
 0.250  0.250  0.000  0.000  0.250 -0.250  0.000  0.000 -0.250  0.000  0.000  0.250  0.000  0.000  0.000  0.000
 0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
 0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
 0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
 0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
TWO DIMENSIONAL SLANT TRANSFORM

 1.000  0.866  0.500 -0.000 -1.118  0.000  0.000 -0.000  0.218  0.000  0.000  0.224  0.000 -0.632  0.000  0.000
 0.866  0.751  0.433 -0.000 -0.969  0.000  0.000 -0.000  0.189  0.000  0.000  0.194  0.000 -0.550  0.000  0.000
 0.500  0.433  0.000 -0.000 -0.447  0.000  0.000 -0.224  0.109  0.000  0.000  0.447  0.000 -0.105 -1.500  0.000
 0.000  0.000  0.000  0.000  0.000  0.000  0.000 -0.000  0.000  0.000  0.000 -0.000  0.000 -0.000  0.000  0.000
-1.118 -0.969 -0.447  0.000  1.200  0.000  0.000 -0.100 -0.244  0.000  0.000 -0.000  0.000 -0.000  0.000  0.000
 0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000 -0.000  0.000  0.000  0.224  0.000
 0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000 -0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
 0.000 -0.000  0.224  0.000 -0.100  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
 0.218 -0.189 -0.109  0.000  0.244  0.000  0.000 -0.200  0.000  0.000  0.000 -0.000  0.000 -0.458  0.447  0.000
 0.000  0.000  0.000  0.000 -0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
 0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000 -0.000  0.000  0.000
 0.224 -0.194 -0.447 -0.000  0.400  0.000  0.000  0.300 -0.049  0.000  0.000  0.000  0.000  0.829 -0.471  0.000
 0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000 -0.000 -0.000  0.000
 0.632  0.550 -0.105 -0.000 -0.000  0.000  0.000 -0.458  0.139  0.000  0.000  0.829  0.000 -0.648 -1.025  0.000
 0.000  0.000  0.500  0.000 -0.224  0.000  0.000 -0.447  0.000  0.000  0.000 -0.671 -0.000 -1.025  1.000  0.000
 0.000  0.000 -0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000 -0.000 -0.000  0.000
TWO DIMENSIONAL DISCRETE COSINE TRANSFORM

 .500  0.451 -0.000 -0.119  0.133  0.280 -0.001 -0.488 -0.557 -0.273  0.001 -0.055 -0.272 -0.276  0.002  0.259
 .466  0.423 -0.000 -0.111  0.125  0.263 -0.001 -0.416 -0.521 -0.256  0.000 -0.051 -0.255 -0.259  0.002  0.243
 .000 -0.000  0.000  0.000 -0.000  0.000  0.000  0.000  0.000  0.000 -0.000  0.000  0.000  0.000 -0.000 -0.000
 .252 -0.237  0.000  0.000 -0.067 -0.141  0.001  0.223  0.279  0.137 -0.000  0.027  0.136  0.137 -0.001 -0.130
 .132 -0.119  0.000  0.031 -0.035 -0.074  0.000  0.117  0.148  0.072 -0.000  0.014  0.071  0.071 -0.000 -0.066
 .000  0.042 -0.000 -0.016  0.018  0.039 -0.000 -0.041 -0.074 -0.037 -0.000 -0.007 -0.037 -0.037  0.000  0.034
 .001 -0.001  0.000  0.000 -0.000 -0.000  0.000  0.001  0.000  0.000 -0.000  0.000  0.000  0.000 -0.000 -0.000
 .320 -0.297  0.000  0.078 -0.087 -0.183  0.001  0.288  0.359  0.175 -0.000  0.038  0.169  0.169 -0.001 -0.153
 .258 -0.243  0.000  0.132 -0.148 -0.308  0.001  0.484  0.603  0.293 -0.001  0.057  0.281  0.279 -0.002 -0.248
 .207 -0.347  0.000  0.094 -0.107 -0.225  0.001  0.351  0.436  0.211 -0.000  0.041  0.199  0.197 -0.001 -0.172
 .002  0.001 -0.000 -0.000  0.000  0.001 -0.000 -0.001 -0.002 -0.001  0.000 -0.000 -0.001 -0.001  0.000  0.001
 .304  0.247 -0.000 -0.070  0.078  0.162 -0.001 -0.251 -0.310 -0.149  0.000 -0.029 -0.145 -0.143  0.001  0.128
 .271  0.244 -0.000 -0.044  0.049  0.102 -0.001 -0.157 -0.194 -0.093  0.000 -0.018 -0.091 -0.089  0.001  0.080
 .077  0.070 -0.000 -0.013  0.020  0.042 -0.000 -0.064 -0.078 -0.037  0.000 -0.007 -0.037 -0.037  0.000  0.032
 .000  0.000 -0.000 -0.000  0.000  0.000 -0.000 -0.000 -0.000 -0.000  0.000 -0.000 -0.000 -0.000  0.000  0.000
 .118  0.106 -0.000 -0.028  0.030  0.063 -0.000 -0.094 -0.113 -0.053  0.000 -0.009 -0.051 -0.050  0.000  0.050

AGE       OBJECT CODE=  12112 BYTES,ARRAY AREA=  110592 BYTES,TOTAL AREA AVAILABLE=  183704 BYTES

TICS      NUMBER OF ERRORS=  0, NUMBER OF WARNINGS=  0, NUMBER OF EXTENSIONS=  0
```