USCEE REPORT #386

Orthogonal Decomposition and Seismic Arrays

by

Julius Kane, Harry Andrews
and
William Pratt

November 1969

Signal and Image Processing Institute
UNIVERSITY OF SOUTHERN CALIFORNIA
Department of Electrical Engineering-Systems
Powell Hall of Engineering
University Park/MC-0272
Los Angeles, CA 90089 U.S.A.

AFCRL-70-0012

ORTHOGONAL DECOMPOSITION AND SEISMIC ARRAYS

by

Julius Kane

Harry Andrews

William Pratt

Department of Electrical Engineering
UNIVERSITY OF SOUTHERN CALIFORNIA
University Park, California 90007

Contract No. F 19628-68-c-0342

Project No. 8652

Task No. 865201

FINAL REPORT

Period Covered: 24 April 1968 to 30 July 1969
20 November 1969

This document has been approved for public release and sale; its distribution is unlimited.

Contract Monitor: Ker C. Thomson
Terrestrial Sciences Laboratory

prepared for

AIR FORCE CAMBRIDGE RESEARCH LABORATORIES
OFFICE OF AEROSPACE RESEARCH
UNITED STATES AIR FORCE
BEDFORD, MASSACHUSETTS 01730

Included is one reprint from the Proceedings of the IEEE, Vol. 57, No. 1, January 1969

ORTHOGONAL DECOMPOSITION AND SEISMIC ARRAYS

It was the objective of this study to see if novel computational techniques could significantly improve the analysis and understanding of seismic data, particularly that generated by large aperture seismic arrays. Conventional methods for processing such data are almost exclusively based upon the Fourier transform. To a great extent this approach is based upon historical inertia. But more than that, trigonometric analysis appears to be well-adapted to the representation of seismic signals. On the other hand, trigonometric functions are but one of an infinite variety of orthonormal functions, and of these others, some are exceptionally well suited for digital computation. In particular, there are the Walsh functions. These have the virtue that they assume only two values, +1 and -1. In addition they mimic many of the properties of the trigonometric functions. Indeed, many of them (the subset known as the Rademacher functions) are merely "hard clipped" sine waves.

Further, the Walsh functions are astonishingly easy to generate and have many convenient mathematical properties. For example, consider the basic 2 x 2 orthogonal, symmetric matrix $\mathbb{M}_2$

$$\mathbb{M}_2 = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

where a factor $2^{-1/2}$ is suppressed.

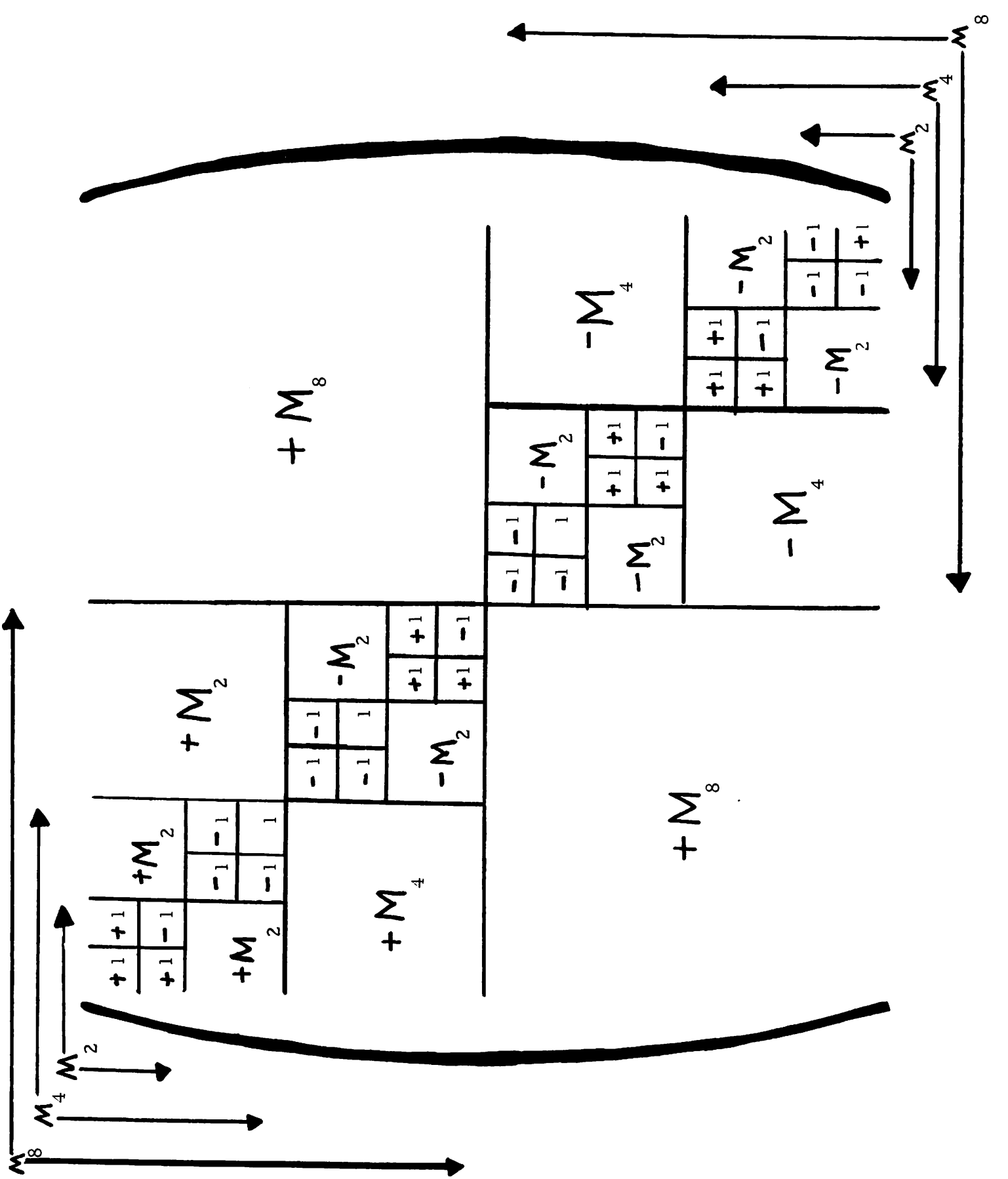


Figure 1

Its row (or column) vectors are the first two Walsh functions and they span R_2. All the Walsh functions for any $n=2^m$ can be obtained after $m=\log n$ steps by m-fold formations of the Kronecker product, schematically indicated in Figure 1. For example, for $8=2^3$ we have

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{pmatrix}$$

It is a very easy matter to show that such generation yields orthogonal matrices for any $n=2^m$. Further, it is easy to reorder the row vectors so that they are in serial order (numbered according to <u>sequency</u>, a square wave analogue of frequency). Further discussion can be found in Harmuth (1969) as well as an extensive bibliography. Techniques for computer implementation of these functions have been worked out by Kane and Andrews (1970) under the subject contract and is here reprinted as Appendix A.

With the power of these computational techniques at our disposal we then considered seismic application. The tracings of an array can be considered to be a two-dimensional picture after the fashion of a

video presentation. The signal from each element is a line of the screen with an intensity proportional to the square of its magnitude at time t. Coherent signals arriving at the array will give a reasonably continuous presentation as we proceed from line to line. (To first order, apart from noise, each line will be a displaced version of the proceeding line). If we take a finite window of this display and quantize each line into i components then we have an i x j array for say, j elements in the array. Call this the signal matrix $\mathbb{S}_{ij}$. While not necessary, it is convenient to suppose that $\mathbb{S}_{ij}$ is a square matrix. Because of the computer facilities available, we generally considered 256 x 256 arrays.

Denote by $\mathbb{H}_{nn}$, the square Hadamard matrix obtained by arranging the first $n=2^m$ Walsh functions. $\mathbb{H}_{nn}$ is symmetric, and when normalized by $2^{-m/z}$, is also orthogonal, and indeed its own inverse. That is

$$\mathbb{H}_{nn}^{-1} = 2^{-m}\mathbb{H}_{nn}$$

Then, the similarity transformation

$$\mathbb{T} = \mathbb{H} \ \mathbb{S} \ \mathbb{H}^{-1}$$

creates a transformed matrix $\mathbb{T}$ such that all the coherent information is crowded into one corner where as, loosely speaking, "noise" is largely uniformly distributed. Thus extraordinary filtering operations and data reduction schemes become possible. Owing to our equipment

limitations we were unable to explore these with actual seismic data, but did at least model these with some digitized video signals of the Surveyor moon shots which were available. These results were published in the Proceedings of the IEEE and are presented as Appendix B.

Data reduction of 6:1 is available by these means with but the simplest of methods, e.g. crude rectangular filtering of $\mathbb{T}$. With more sophisticated methods, such as the introduction of error correcting codes, band-with reductions of 15:1 and even 20:1 become quite feasible.

In addition, iterated matched filtering becomes possible. Traditionally mathced filtering is done once, in the Fourier Domain. But for quantization $n=2^m$ there are $2m$ orthogonal domains which can be used to advantage before the signal is reconstructed. With a "partial matched filter" in each domain extraordinary filtering techniques become feasible. For such applications <u>any</u> orthogonal representation will do, but at least for digital implementation, processing via Walsh functions presents major computational advantages. Unfortunately, much of this work is being done after the termination of the present contract, and the details are still under investigation. Acknowledgement, and reprints of this work will be forwarded when published.

References

(1) Harmuth, F. Henning: Transmission of Information by Orthogonal Functions, Springer-Verlag, New York, 1969.

APPENDIX A

KRONECKER MATRICES, COMPUTER IMPLEMENTATION, AND GENERALIZED SPECTRA

H. C. Andrews

and

J. Kane

Electrical Engineering Department
University of Southern California
Los Angeles, California 90007

Submitted to:

Association for Computing Machinery

December 1968

Revised: June, 1969

The work reported in this paper was supported by NASA under Grant NGR 05-018-044, JPL Grant No. 952312 and AF CRL-68-C-0342.

KRONECKER MATRICES, COMPUTER IMPLEMENTATION, AND GENERALIZED SPECTRA

ABSTRACT

A closed product form and a simplified algorithm for efficient matrix operation is presented for a class of generalized kronecker matrices. Powers of two kronecker matrices are further described where the closed form representation is easily implemented with parallel binary register operations. Orthogonal and symmetric orthogonal kronecker matrices are described in the context of generalized spectral analyses. Specific applications are presented which include the Fourier, Hadamard, and other transformations.

KRONECKER MATRICES, COMPUTER IMPLEMENTATION, AND GENERALIZED SPECTRA

Introduction

Often orthogonal matrices will play a major role in the study of spectral analysis and efficient decomposition schemes for unknown functions on a digital computer. A certain class of orthogonal matrices are described which are very efficiently implemented for computer application. The class of orthogonal matrices described are shown to be a subset of a generalized class of Kronecker matrices which also have a very efficient computer implementation algorithm. It is hoped that certain of these matrices could be useful for efficient signal processing in the form of possibly multi-dimensional transformations.

A Generalized Class of Kronecker Matrices

Consider the class of matrices formed by the kronecker product operation. Let the sub-matrices be square and of dimension p by p with entries $m_{r,i,j}$ where i and j range from zero through p - 1.

$$M_r = \begin{bmatrix} m_{r,0,0} & m_{r,0,1} & \cdots & m_{r,0,p-1} \\ m_{r,1,0} & & & \\ \vdots & & & \\ m_{r,p-1,0} & \cdots & \cdots & m_{r,p-1,p-1} \end{bmatrix} \tag{1}$$

Here the first index represents the class of entries corresponding to a particular dimension in the kronecker product operation. In general

$$H_1 = M_0 \tag{2a}$$

$$H_2 = M_1 \otimes H_1 \tag{2b}$$

$$\vdots$$

$$H_n = M_{n-1} \otimes H_{n-1} \tag{2c}$$

where $\otimes$ is the kronecker product operator. Thus

$$H_n = \begin{bmatrix} m_{n-1,0,0} H_{n-1} & \cdots & m_{n-1,0,p-1} H_{n-1} \\ m_{n-1,1,0} H_{n-1} & \cdots & m_{n-1,1,p-1} H_{n-1} \\ \vdots & & \vdots \\ m_{n-1,p-1,0} H_{n-1} & \cdots & m_{n-1,p-1,p-1} H_{n-1} \end{bmatrix} \tag{3}$$

where H_n is a p^n by p^n matrix.

When operating with kronecker matrices within a computer, it becomes desirable to store a representation (algorithm) of the entries of the product matrix rather than the matrix itself. Towards this end, consider the locations in

the matrix to be described by their lexicographic or dictionary sequence representation. In other words, a given index of matrix H_n can be represented by n digits each of which can take on the value zero through p-1. Representing the horizontal index by u and the vertical index by x, the names of the rows and columns in dictionary sequence for the H_2 matrix with p = 3 are

$$H_2 = \begin{array}{c|c} & u \longrightarrow \\ & \begin{array}{ccccccccc} 00 & 01 & 02 & 10 & 11 & 12 & 20 & 21 & 22 \end{array} \\ \hline x \downarrow \begin{array}{c} 00 \\ 01 \\ 02 \\ 10 \\ 11 \\ 12 \\ 20 \\ 21 \\ 22 \end{array} & \left[\begin{array}{ccccccccc} & & & & & & & & \\ & & & & & & & & \\ & & & & & & & & \\ & & & & & & & & \\ & & & & H_2(x,u) & & & & \\ & & & & & & & & \\ & & & & & & & & \\ & & & & & & & & \\ & & & & & & & & \end{array} \right] \end{array} \tag{4}$$

Representing the u and x variables in the dictionary number system mod p requires n digits to allow u and x to range over zero to p^n. Therefore u and x can be described by

$$u = u_{n-1}\, u_{n-2} \ldots u_1 u_0 \qquad u_i \;\varepsilon\{0, 1, \ldots, p-1\} \tag{5a}$$

$$x = x_{n-1}\, x_{n-2} \ldots x_1 x_0 \qquad x_i \;\varepsilon\{0, 1, \ldots, p-1\} \tag{5b}$$

Using such a notation allows the entries of the p by p core matrix H_1 [equation (2a)] to be described by the equation

$$H_1(x, u) = \prod_{i=0}^{p-1} \prod_{j=0}^{p-1} m_{0,i,j}^{\delta(x_0-i)\,\delta(u_0-j)}\ , \text{ and} \tag{6a}$$

$$H_1(x, u) = m_{0,i,j} \tag{6b}$$

where $\delta(a-b)$ is the delta function which takes on the value one whenever $a=b$ and zero otherwise. The representation of equation (6a) can be interpreted as multiplying all entries of the core matrix, equation (2a), together and noting that all but one entry will be raised to the zero power. The entries of the p^2 by p^2 matrix, H_2, equation (2b), can now be represented as

$$H_2(x,u) = \prod_{i=0}^{p-1} \prod_{j=0}^{p-1} m_{1,i,j}^{\delta(x_1-i)\delta(u_1-j)} \prod_{i=0}^{p-1} \prod_{j=0}^{p-1} m_{0,i,j}^{\delta(x_0-i)\delta(u_0-j)} \tag{7}$$

where, again, the exponents determine the correct product of entries for a given u and x. In general, the entries for H_n can be represented as

$$H_n(x,u) = \prod_{r=0}^{n-1} \prod_{i=0}^{p-1} \prod_{j=0}^{p-1} m_{r,i,j}^{\delta(x_r-i)\delta(u_r-j)} \tag{8}$$

following the recursive notation of equation (6a) and (7). Representation of the rows or columns of a kronecker matrix in the form of equation (8) now allows the generation of any single element, column, or row of the matrix without storage of the entire matrix array. This becomes particularly important for large matricies especially in the area of generalized spectral analysis to be described later.

In addition to representing the kronecker matrices in closed product form, it is important to point out that vector multiplication with the above described matrices can be implemented on the order of $pN \log_p N$ operations where $N = p^n$ is the dimension of the H_n kronecker matrix. This should be contrasted with the N^2 operations normally required. This result was pointed out by Good [1], is referenced by the Cooley Tukey algorithm [2], and leads to the Fast Hadamard transform algorithm [3, 4]. A variation of the Fourier algorithm, for parallel processing, is presented by Pease [5]. The class

kronecker matrices described above can be decomposed into a product of matrices each of which has only p entries in a given row or column. Thus for the H_n kronecker matrix of equation (3) there exist n matrices, each of dimension p^n, such that when multiplied together, they will equal H_n. These matrices can be described as

$$G_r = \begin{bmatrix} m_{r,0,0},\ldots,m_{r,0,p-1} & & \\ & m_{r,0,0},\ldots,m_{r,0,p-1} & \vdots \\ & & m_{r,0,0},\ldots,m_{r,0,p-1} \\ m_{r,1,0},\ldots,m_{r,1,p-1} & & \\ & m_{r,1,0},\ldots,m_{r,1,p-1} & \vdots \\ \cdots\cdots & & m_{r,1,0},\ldots,m_{r,1,p-1} \\ & \cdots\cdots & \cdots\cdots \\ m_{r,p-1,0},\ldots,m_{r,p-1,p-1} & & \\ & m_{r,p-1,0},\ldots,m_{r,p-1,p-1} & \vdots \\ & & m_{r,p-1,0},\ldots,m_{r,p-1,p-1} \end{bmatrix} \quad (9)$$

In this matrix there are p^{n+1} non-zero entries and only p^2 non-redundant elements.

Then

$$H_n = G_{n-1} \; G_{n-2} \; \ldots \; G_1 G_0 \tag{10}$$

Now if a row vector is multiplied on the left by H_n, $N^2 = (p^n)^2$ operations will be required whereas if the vector is multiplied by G_{n-1}, pN operations will be required. If the resulting vector is multiplied by G_{n-2}, another pN operations will be required. If this step is carried out $n = \log_p N$ times, then a total of $p\,N \log_p N$ operations are necessary.

By using the Good algorithm described above and by using the closed product representation of equation (8) a kronecker matrix of large dimension can be generated and matrix-manipulated without storing the N^2 term matrix. Conceivably the set of coeffcients $m_{r,i,j}$ of equation (8) could all be distinct, in which case a total of only np^2 coefficients must be stored. However when the class of $m_{r,i,j}$ are not all distinct, considerably more savings can be achieved. It is instructive to investigate the class of matrices generated by the kronecker operation with the $m_{r,i,j} = m_{s,i,j}$ for all r and s. In such a situation equation (8) reduces to

$$H_n(x,u) = \prod_{i=0}^{p-1} \prod_{j=0}^{p-1} m_{i,j}^{\sum_{r=0}^{n-1} \delta(x_r - i)\delta(u_r - j)} \tag{11}$$

Now only p^2 coefficients need be stored compared to np^2 terms.

<u>Powers of Two Kronecker Matrices</u>

If the general matrices described above are generated from a two by two core matrix, the closed product representation analogous to equation (8) becomes particularly convenient to implement. Let the core matrix H_1 be

$$H_1 = \begin{bmatrix} A_0 & B_0 \\ C_0 & D_0 \end{bmatrix} \tag{12}$$

and H_n can be represented as

$$H_n = \begin{bmatrix} A_{n-1} & H_{n-1} & B_{n-1} & H_{n-1} \\ C_{n-1} & H_{n-1} & D_{n-1} & H_{n-1} \end{bmatrix} \tag{13}$$

The closed form product representation now becomes

$$H_n(x, u) = \prod_{r=0}^{n-1} A_r^{\bar{x}_r \bar{u}_r} B_r^{\bar{x}_r u_r} C_r^{x_r \bar{u}_r} D_r^{x_r u_r} \tag{14}$$

where the exponent operations become Boolean "and" operations, and the bar over the binary variable represents the complement value. For the case in which $A_r = A_s$, $B_r = B_s$, $C_r = C_s$, $D_r = D_s$ for all r and s, the representation again simplified and becomes

$$H_n(x, u) = A^{\sum_{r=0}^{n-1} \bar{u}_r \bar{x}_r} B^{\sum_{r=0}^{n-1} \bar{x}_r u_r} C^{\sum_{r=0}^{n-1} x_r \bar{u}_r} D^{\sum_{r=0}^{n-1} x_r u_r} \tag{15}$$

Equation (15) is particularly suited for special purpose digital implementation as the exponent operations require simply counting the number of "ones" obtained from a parallel component wise register "and" operation on the values of u and x.

This means that generation of the rows, columns, or specific elements of the matrix H_n requires storage of only four variables (A, B, C, D) and simple register "and" operations for any dimension $N = 2^n$. Then to implement a vector-matrix product will require $2Nn = 2N \log_2 N$ operations with a storage requirement of only 4 variables.

Orthogonal Matrices and Generalized Spectra

In the area of signal processing, it is often desirable to perform some type of transformation on a function in order to learn more about that function. Certain computational processes of particular interest are discrete orthogonal transformations on input functions in the search for characteristic properties of the transform domains otherwise obscurred in the original data. Because of the restriction of digital computation to discrete operations, it is natural to think of digital transformation processes on digitized functions in the context of matrix algebra. In such a situation let the input function be f(x) considered as a vector in the x dimension with $N = 2^n$ samples and let the transformation matrix be H_n. Then the transformation operation can be expressed in matrix notation as

$$\begin{bmatrix} f(x) \end{bmatrix} \begin{array}{c} \xrightarrow{u} \\ x\downarrow \end{array} \begin{bmatrix} \\ H_n(x, u) \\ \\ \end{bmatrix} = \begin{bmatrix} F(u) \end{bmatrix} \tag{16}$$

Where the vector F(u) is the transformation result of the vector-matrix multiplication. The function F(u) can now be expressed in arithmetic form as

$$F(u) = \sum_x f(x) \; H_n(x, u) \tag{17}$$

where $H_n(x, u)$ can be considered the kernel of the transformation process.

If the matrix H_n is constrained to be orthogonal, then the linear transformation can be interpreted as a decomposition of the input data into generalized spectra where each spectral component in the transform domain corresponds to the amount of energy of that spectral orthogonal function within the input data. Using such a concept, the idea of frequency can now be generalized to include transformations of orthogonal functions other than sine and cosine waves. This type of generalized spectral analysis will allow the investigation of specific orthogonal decompositions which could be "better" matched (in possibly an eigenvector sense) to specific purposes and input data classifications [6]. In passing it should be noted that the above discussion can be generalized to multi-dimensional transformations resulting in

$$F(u^{(1)}, u^{(2)}, \ldots, u^{(p)}) = \sum_{x^{(1)}} \cdots \sum_{x^{(p)}} f(x^{(1)}, \ldots x^{(p)}) H(x^{(1)}, u^{(1)}, x^{(2)}, u^{(2)}, \ldots, x^{(p)}, u^{(p)}) \tag{18}$$

where the 2-tuples $\{x^{(i)}, u^{(i)}\}$ relate the $x^{(i)}$ dimension data space with the corresponding $u^{(i)}$ generalized spectral components in the transform spectral space obtained from that particular $x^{(i)}$ data dimension. An example of the application of such a multi-dimensional concept is one in which the orthogonal basis vectors in a particular dimension of the data space are matched to the natural eigenvectors of that dimension. In other words a Fourier transform might be applicable to one dimension of data whereas a Hadamard transform might be more applicable to data transformations in a second dimension [2, 3, 4].

A general class of orthogonal transforms of particular interest can be obtained from the kronecker matrices described in earlier sections by requiring that the sets of variables $\{m_{r,0,0}, \ldots, m_{r,p-1,p-1}\}$ satisfy the orthogonality requirement for all $r = 0, \ldots, n-1$. If this constraint is satisfied, then the matrix, H_n, of equation (3) or (8) becomes orthogonal and is a valid candidate for a kernel in a generalized spectral decomposition problem. For the power of two case the orthogonality constraint on the sets $\{A_r, B_r, C_r, D_r\}$ for all $r = 0, \ldots, n-1$ reduces to

$$A_r^2 + B_r^2 = 1 \tag{19a}$$

$$C_r^2 + D_r^2 = 1 \tag{19b}$$

$$A_r C_r + B_r D_r = 0 \tag{19c}$$

for each r. In this case equation (14) becomes the kernel of the transform and when the sets $\{A_r, B_r, C_r D_r\}$ are all identical, equation (15) becomes the kernel.

If it is desired to make the kernel matrix symmetric so that a transformation taken twice results in the original function again, then further simplifications result in the closed form representation of the matrix H_n. The requirement for symmetry and orthogonality for the case of identical sets $\{A_r, B_r, C_r, D_r\}$ for all r is

$$B = C \tag{20a}$$

$$A^2 + B^2 = 1 \tag{20b}$$

$$B^2 + D^2 = 1 \tag{20c}$$

$$(A + D)B = 0 \tag{20d}$$

The entries of the matrix H_n become

$$H_n(x,u) = A^{\sum_{r=0}^{n-1} \bar{u}_r \bar{x}_r} \; B^{\sum_{r=0}^{n-1} u_r \oplus x_r} \; D^{\sum_{r=0}^{n-1} u_r x_r} \tag{21}$$

Where $\oplus$ implies an exclusive "or" Boolean operation. Notice that the exponents can be determined by summing the result of parallel register operations (Boolean "and" and Boolean "exclusive or") on the variables u and x. However equation (21), under the constraint that B = 0 satisfying equation (20d), reduces to

$$H_n(x,u) = A^{\sum_{r=0}^{n-1} \bar{u}_r} \; D^{\sum_{r=0}^{n-1} u_r} \; \delta(u-x) \tag{22a}$$

or

$$H_n(x,u) = \left(\frac{D}{A}\right)^{\sum_{r=0}^{n-1} u_r} \; \delta(u-x) \tag{22b}$$

which is a diagonal matrix. The alternative constraint to equation (20d) is that A = -D in which case more interesting orthogonal symmetric matrices result.

$$H_n(x,u) = A^{\sum_{r=0}^{n-1} u_r \odot x_r} \; B^{\sum_{r=0}^{n-1} u_r \oplus x_r} \; (-1)^{\sum_{r=0}^{n-1} u_r x_r} \tag{23a}$$

or

$$H_n(x,u) = A^n \left(\frac{1-A^2}{A^2}\right)^{\frac{1}{2}\sum_{r=0}^{n-1} u_r \oplus x_r} \; (-1)^{\sum_{r=0}^{n-1} u_r x_r} \tag{23b}$$

where $\odot$ is the Boolean "coincidence" operation equivalent to the complement of the exclusive "or" operation. The class of orthogonal matrices described by equations (23) is a two parameter family of sets of kronecker matrices

subject to the constraint that $A^2 + B^2 = 1$. Consequently, valid 2-tuples satisfying this requirement are $\{\cos\theta, \sin\theta\}$, $\{3/5, 4/5\}$, $\{1/\sqrt{2}, 1/\sqrt{2}\}$ and many others.

Applications

Specific transformations which are readily implementable in the context of the above kronecker and matrix factorization techniques include the Fourier, Hadamard or Walsh, Generalized Walsh, and a variety of other unnamed transforms. Referring to equation (23b) it was indicated that the 2-tuple given by $\{\cos\theta, \sin\theta\}$ describes M_r for all r as

$$M_r = \begin{bmatrix} \cos\theta & \sin\theta \\ \sin\theta & -\cos\theta \end{bmatrix} \tag{24}$$

As θ varies from 0° to 45° the resulting kronecker matrix varies from a diagonal matrix to one in which the energy in each row (and column) is uniformly distributed over every entry. The matrix is given by

$$H_n(x,u) = \cos^n\theta \left(\frac{\sin\theta}{\cos\theta}\right)^{\sum_{r=0}^{n-1} u_r \oplus x_r} (-1)^{\sum_{r=0}^{n-1} u_r x_r} \tag{25}$$

as can be seen from equation (23b). When $\theta = 0^\circ$ we use the fact that zero raised to the zero power is one and when $\theta = 45^\circ$ this matrix reduces to the Hadamard matrix of order 2 which is equivalent to the discrete Walsh transform, [7]. The matrix of equation (25) then becomes

$$H_n(x,u) = \left(\frac{1}{2}\right)^{n/2} (-1)^{\sum_{r=0}^{n-1} u_r x_r} \tag{26}$$

The transformation described by equation (26) also describes a class of error correcting codes given by Hadamard matrices of order 2. Another example of a Hadamard matrix which can be easily represented in the above described lexicographic notation is the powers of four matrix generated by kronecker products of

$$M_r = \begin{bmatrix} 1 & 1 & 1 & -1 \\ 1 & 1 & -1 & 1 \\ 1 & -1 & 1 & 1 \\ -1 & 1 & 1 & 1 \end{bmatrix} \tag{27}$$

for all r. In this case equation (11) describes the matrix in closed product form where $p = 4$ and $m_{i,j} = -1$ for all $i+j = 3$ and $m_{i,j} = 1$, otherwise. Consequently, equation (11) reduces to

$$H_n(x,u) = \prod_{r=0}^{n-1} (-1)^{\delta(x_r + u_r - 3)} \tag{28a}$$

or

$$H_n(x,u) = (-1)^{\sum_{r=0}^{n-1} \delta(x_r + u_r - 3)} \tag{28b}$$

where x_r and u_r range from zero through 3. This particular transformation has the property that each orthogonal vector in the matrix H_n has approximately the same number of zero crossings. This is to be contrasted to the Hadamard (Walsh) transform which has $N = 2^n$ different number of zero crossings.

The Walsh transform, equation (26), has been generalized to a much larger class of orthogonal transformations by Chrestenson [8] who has described many of the convergence properties of this expanded class. In discrete matrix notation the generalized Walsh transforms of order p require M_r for all r to be given by

$$M_r(x, u) = [W^{ux}] \tag{27}$$

where $W = \exp\{2\pi j/p\}$ and simplifications can be made due to the fact that $W^{ux} = W^{ux \bmod p}$. For an N by N discrete generalized Walsh transform where $N = p^n$, the matrix is given by

$$H_n(x, u) = \prod_{i=0}^{p-1} \prod_{j=0}^{p-1} W^{ij \sum_{r=0}^{n-1} \delta(x_r - i)\, \delta(u_r - j)} \tag{28}$$

Note that the discrete generalized Walsh transform of order 2 reduces to the Hadamard transform. It is also interesting to note that the generalized Walsh transform core matrix, equation (27), performs a Fourier transform of resolution p. However, the kronecker product of the generalized Walsh transform, equation (28), no longer performs a Fourier transformation.

Conclusions

It has been the purpose of this paper to present a generalized closed product representation of certain classes of kronecker matrices. Vector or matrix multiplications have been shown to be implementable in fewer operations than normal matrix algebra requires. Powers of two kronecker matrices are described and are shown to be representable in simple product

forms which reduce to parallel binary register operations in the exponents of the core matrix entries. Orthogonal kronecker matrices are described and generalized spectral analysis technques are presented. Finally, some specific applications are presented in order to develop some practical results using the concepts presented earlier. While the applications are not exhaustive, they do present a few specific areas where processing with kronecker matrices occupy a pertinent role.

Acknowledgements

The assistance of Mr. K. Caspari and J. Lee of ITT-EPL, Hyattsville, Maryland in pointing out the generalized Walsh references is greatly appreciated.

References

1. Good, I.J., "The Interaction Algorithm and Practical Fourier Analysis", Journal of the Royal Statistical Society (London), Vol. B20, p. 361, (1958).

2. Cooley, J.W. and Tukey, J.W., "An Algorithm for the Machine Calculation of Complex Fourier Series", Math. Computation, Vol. 19, pp. 297-301, (April, 1965).

3. Yates, F., "The Design and Analysis of Factorial Experiments", Imperial Bureau of Soil Science, No. 35, Harpenden, (1937).

4. Pratt, W.K., Kane, J., and Andrews, H.C., "Hadamard Transform Image Coding", IEEE Proceedings, Vol. 57, No. 1, pp. 58-68, (January, 1969).

5. Pease, M.C., "An Adaptation of the Fast Fourier Transform for Parallel Processing", JACM, Vol. 15, No. 2, pp. 252-264 (April, 1968).

6. Whechel, J.E., and Guinn, D.F., "The Fast Fourier-Hadamard Transform and Its Use in Signal Representation and Classification", Eascon '68 Record, pp. 561-573.

7. Walsh, J.L., "A Closed Set of Normal Orthogonal Functions", Ann. J. Math., Vol. 45, pp. 5-24, (1923).

8. Chrestenson, H.E., "A Class of Generalized Walsh Functions", Pacific Journal of Mathematics, Vol. 5, pp. 17-31, (1955).

APPENDIX B

HADAMARD TRANSFORM IMAGE CODING

BY

W. K. PRATT, J. KANE, AND H. C. ANDREWS

Reprinted from the PROCEEDINGS OF THE IEEE
VOL. 57, NO. 1, JANUARY, 1969
pp. 58-68

Hadamard Transform Image Coding

WILLIAM K. PRATT, MEMBER, IEEE, JULIUS KANE, SENIOR MEMBER, IEEE, AND HARRY C. ANDREWS, MEMBER, IEEE

Abstract—**The introduction of the fast Fourier transform algorithm has led to the development of the Fourier transform image coding technique whereby the two-dimensional Fourier transform of an image is transmitted over a channel rather than the image itself. This development has further led to a related image coding technique in which an image is transformed by a Hadamard matrix operator. The Hadamard matrix is a square array of plus and minus ones whose rows and columns are orthogonal to one another. A high-speed computational algorithm, similar to the fast Fourier transform algorithm, which performs the Hadamard transformation has been developed. Since only real number additions and subtractions are required with the Hadamard transform, an order of magnitude speed advantage is possible compared to the complex number Fourier transform. Transmitting the Hadamard transform of an image rather than the spatial representation of the image provides a potential toleration to channel errors and the possibility of reduced bandwidth transmission.**

I. Introduction

IN PREVIOUS papers a new technique of digital image coding, Fourier transform coding, has been introduced [1]–[3]. By this technique a two-dimensional Fourier transform of a digitized image is performed by a digital computer using the fast Fourier transform [4]–[8]. If $f(x, y)$ represents the amplitude of image samples over a square array of N^2 points then the two-dimensional Fourier transform $F(u, v)$ is defined as[1]

$$F(u, v) = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) \exp\left\{-\frac{2\pi i}{N}(ux + vy)\right\}. \quad (1)$$

The Fourier transform of the image is quantized, coded, and transmitted over the channel. Then, at the receiver, the inverse Fourier transform

$$\hat{f}(x, y) = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} \hat{F}(u, v) \exp\left\{\frac{2\pi i}{N}(xu + yv)\right\}, \quad (2)$$

is taken of the decoded transform $\hat{F}(u, v)$ to reconstruct a close approximation of the original image. Fig. 1 contains photographs taken from a cathode ray tube of an original test scene, the logarithm of the magnitude of its two-dimensional Fourier transform, and the inverse Fourier transform of the Fourier transform. The logarithm of the magnitude of the Fourier transform is displayed rather than the magnitude itself because of dynamic range limitations of the recording film.

The Fourier transform image coding technique has been shown to provide good quality image transmission with the same number of bits as required to code the spatial domain of an image by conventional pulse code modulation. Furthermore, transmission of the Fourier transform of an image rather than the image itself offers a certain immunity to channel errors and the ability to achieve a bandwidth reduction [2].

The tolerance of channel errors is due to the averaging process of the Fourier transform as indicated by (2). Each point in a reconstructed image is a weighted sum of all points in the spatial frequency domain. For subjective image viewing the overall loss of resolution that occurs when channel errors are introduced in the Fourier transform is often less objectionable than discrete defects that occur when channel errors affect the spatial representation of an image.

Bandwidth reduction is possible because the image energy, which is usually uniformly distributed in the spatial domain, tends to be concentrated near the origin of the Fourier domain, as evidenced by Fig. 1(b). Many of the higher spatial frequency components are of very low magnitude, and need not be transmitted.

The Fourier transform was a natural operator to be applied to the image coding problem because of its widespread use in other fields and the fact that a very efficient computational algorithm exists. A question that naturally arises is: are there other applicable transforms beside the Fourier transform? The answer is affirmative. For the image coding application, strictly speaking, all that is required is a two-dimensional operator which has an inverse, possesses the averaging property, and redistributes the image energy properly. It is advantageous from the standpoint of implementation that a fast computational algorithm exists and that the operator be its own inverse. The symmetric Hadamard matrix transform fulfills all of these requirements, and as will be discussed later, the Hadamard transform is better in many respects than the Fourier transform for image coding. Both transforms, in fact, are special cases of a matrix multiplication of the function to be transformed by a general matrix multiplier. Which matrix multipliers may be factored to reduce computation remains an open question [9]. Because of this, the broader questions as to what other transforms exist and what is the best transform for image coding will not be considered here.

Manuscript received July 22, 1968; revised October 14, 1968. This work was supported by NASA Grant NGR-05-018-044, Jet Propulsion Laboratory Gran 952312, and U. S. Air Force Grant AFCRL-68-C-0342.

W. K. Pratt and H. C. Andrews are with the Department of Electrical Engineering, University of Southern California, Los Angeles, Calif. 90007

J. Kane was with the Department of Electrical Engineering, University of Southern California. He is currently with the Department of Mathematics and Mathematical Ecology, University of British Columbia, Vancouver, B. C., Canada.

[1] Some authors insert the multiplying factors $1/N$ or $1/N^2$ before the double summation for purposes of normalization. The multiplying factor is not inherent to the computational algorithm and will be omitted here for both the Fourier and Hadamard transform formulations in series form.

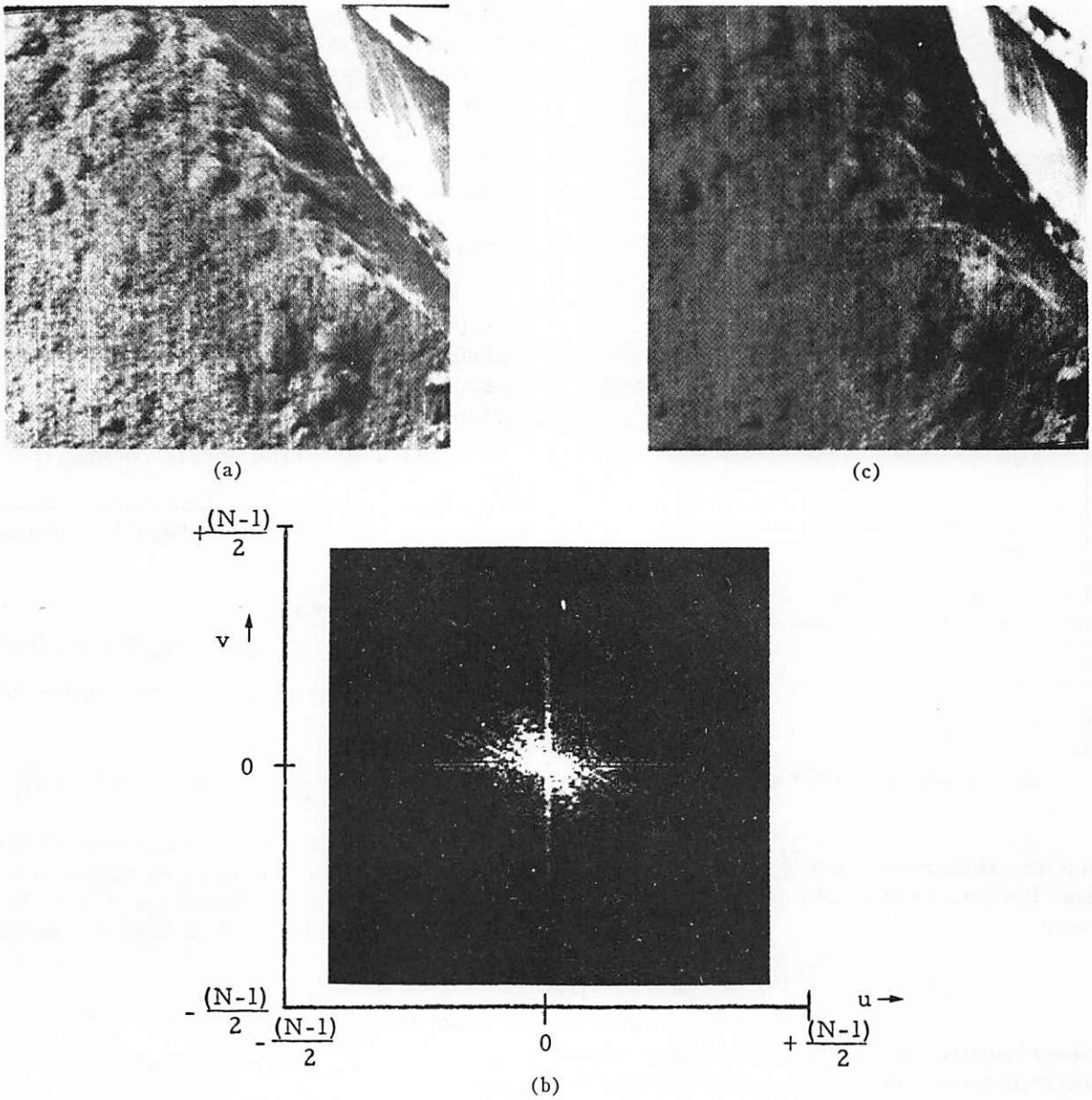


Fig. 1. Fourier transforms of Surveyor footpad; (a) footpad, (b) the logarithm of the magnitude of the Fourier transform of (a), and (c) the inverse Fourier transform of the Fourier transform of (a).

The Hadamard transform has been applied to the coding of vocoder speech signals by Crowther and Rader [10]. They have built coder and decoder networks whose designs are based upon a Hadamard matrix. Because of the correlation properties of the vocoder speech signals, it was found possible to reduce the number of quantization levels of the Hadamard transformed speech signals.

II. Hadamard Matrices [11], [13]

The Hadamard matrix[2] is a square array of plus and minus ones whose rows (and columns) are orthogonal to one another. If $\boldsymbol{H}$ is an $N \times N$ Hadamard matrix then the product of $\boldsymbol{H}$ and its transpose is the identity matrix. Thus,

$$\boldsymbol{H}\boldsymbol{H}^{+} = N\boldsymbol{I} \tag{3}$$

[2] Boldface symbols représent matrices.

where $\boldsymbol{I}$ is the identity matrix. If $\boldsymbol{H}$ is a symmetric Hadamard matrix then (3) reduces to

$$\boldsymbol{H}\boldsymbol{H} = N\boldsymbol{I}. \tag{4}$$

The rows (and columns) of a Hadamard matrix may be exchanged with one another without affecting the orthogonality properties of the matrix.

The lowest-order Hadamard matrix is of order two.

$$\boldsymbol{H} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}. \tag{5}$$

It is known that if a Hadamard matrix of order N exists ($N > 2$), then $N \equiv 0 \pmod 4$. The existence of a Hadamard matrix for every value of N satisfying this requirement has not been shown, but constructions are available for nearly all permissible values of N up to 200. The simplest con-

$$N = 2 \quad \begin{array}{c} \text{matrix} \\ \begin{bmatrix} + & + \\ + & - \end{bmatrix} \end{array} \quad \begin{array}{c} \text{Sequency} \\ 0 \\ 1 \end{array}$$

$$N = 4 \quad \begin{array}{c} \text{matrix} \\ \begin{bmatrix} + & + & + & + \\ + & - & + & - \\ + & + & - & - \\ + & - & - & + \end{bmatrix} \end{array} \quad \begin{array}{c} \text{Sequency} \\ 0 \\ 3 \\ 1 \\ 2 \end{array}$$

$$N = 8 \quad \begin{array}{c} \text{matrix} \\ \begin{bmatrix} + & + & + & + & + & + & + & + \\ + & - & + & - & + & - & + & - \\ + & + & - & - & + & + & - & - \\ + & - & - & + & + & - & - & + \\ + & + & + & + & - & - & - & - \\ + & - & + & - & - & + & - & + \\ + & + & - & - & - & - & + & + \\ + & - & - & + & - & + & + & - \end{bmatrix} \end{array} \quad \begin{array}{c} \text{Sequency} \\ 0 \\ 7 \\ 3 \\ 4 \\ 1 \\ 6 \\ 2 \\ 5 \end{array}$$

Fig. 2. Hadamard matrices of order $N=2^n$.

struction is for a Hadamard matrix of order $N=2^n$ where n is an integer. In this case if $\boldsymbol{H}$ is a Hadamard matrix of order N, the matrix

$$\boldsymbol{G} = \begin{bmatrix} \boldsymbol{H} & \boldsymbol{H} \\ \boldsymbol{H} & -\boldsymbol{H} \end{bmatrix} \tag{6}$$

is a Hadamard matrix of order $2N$. Fig. 2 contains several Hadamard matrices of order $N=2^n$. Another simple construction is possible if $\boldsymbol{A}$ and $\boldsymbol{B}$ are Hadamard matrices of orders M and N, respectively. Then there exists a Hadamard matrix of order $M \cdot N$ given by

$$\boldsymbol{H}_{M \cdot N} = \begin{bmatrix} a_{11}\boldsymbol{B} & a_{12}\boldsymbol{B} \cdots a_{1M}\boldsymbol{B} \\ a_{21}\boldsymbol{B} & \vdots \\ \vdots & \vdots \\ a_{M1}\boldsymbol{B} \cdots\cdots\cdots & a_{MM}\boldsymbol{B} \end{bmatrix} \tag{7}$$

Other constructions are given in [13]–[15]. The set of known Hadamard matrices is sufficiently numerous to satisfy almost all size requirements for image coding.

A frequency interpretation can be given to the Hadamard matrix. Along each row of the Hadamard matrix the frequency is called the number of changes in sign. Harmuth has coined the word "sequency" to designate the number of sign changes [16]. Fig. 2 gives the sequency interpretation for several Hadamard matrices of binary order. It is possible to construct a Hadamard matrix of order $N=2^n$ that has frequency components at every integer from 0 to $N-1$.

This frequency interpretation of the rows of a Hadamard matrix leads one to consider the rows to be equivalent to rectangular waves ranging between ± 1 with a subperiod of $1/N$ units. Such functions are called Walsh functions [17]–[21] and are further related to the Rademacher functions [22]. Thus, in this context the Hadamard matrix merely performs the decomposition of a function by a set of rectangular waveforms rather than the sine-cosine waveforms associated with the Fourier transform.

III. Hadamard Transformation of Images

Let the array $f(x, y)$ represent the intensity samples of an original image over an array of N^2 points. Then the two-dimensional Hadamard transform, $F(u, v)$, of $f(x, y)$ is given by the matrix product

$$[F(u, v)] = [H(u, v)][f(x, y)][H(u, v)] \tag{8}$$

where $[H(u, v)]$ is a symmetric Hadamard matrix of order N. Pre- and post-multiplication of $[F(u, v)]$ by the Hadamard matrix gives

$$\begin{aligned} &[H(u, v)][F(u, v)][H(u, v)] \\ &\quad = [H(u, v)][H(u, v)][f(x, y)][H(u, v)][H(u, v)]. \end{aligned} \tag{9}$$

But, from (4), for a symmetric Hadamard matrix, $\boldsymbol{HH} = N\boldsymbol{I}$. Hence,

$$[f(x, y)] = \frac{1}{N^2}[H(u, v)][F(u, v)][H(u, v)] \tag{10}$$

and, aside from the constant scaling factor N^2, the arrays $f(x, y)$ and $F(u, v)$ are two-dimensional transform pairs.

For symmetric Hadamard matrices of order $N=2^n$, the two-dimensional Hadamard transform may be written in series form as

$$F(u, v) = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y)(-1)^{p(x,y,u,v)} \tag{11}$$

where

$$p(x, y, u, v) \equiv \sum_{i=0}^{n-1} (u_i x_i + v_i y_i).$$

The terms u_i, v_i, x_i and y_i are the binary representations of u, v, x, and y respectively. For example,

$$(u)_{\text{decimal}} = (u_{n-1}u_{n-2} \cdots u_1 u_0)_{\text{binary}} \tag{12}$$

where $u_i \in \{0, 1\}$. In (11) the summation in the exponent is performed modulo two. This representation of the Hadamard transform is for the Hadamard matrix in "natural" form as given by (6). Another series representation exists for a Hadamard matrix in "ordered" form in which the sequency of each row is larger than the preceding row. By this representation

$$F(u, v) = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y)(-1)^{q(x,y,u,v)} \tag{13}$$

where

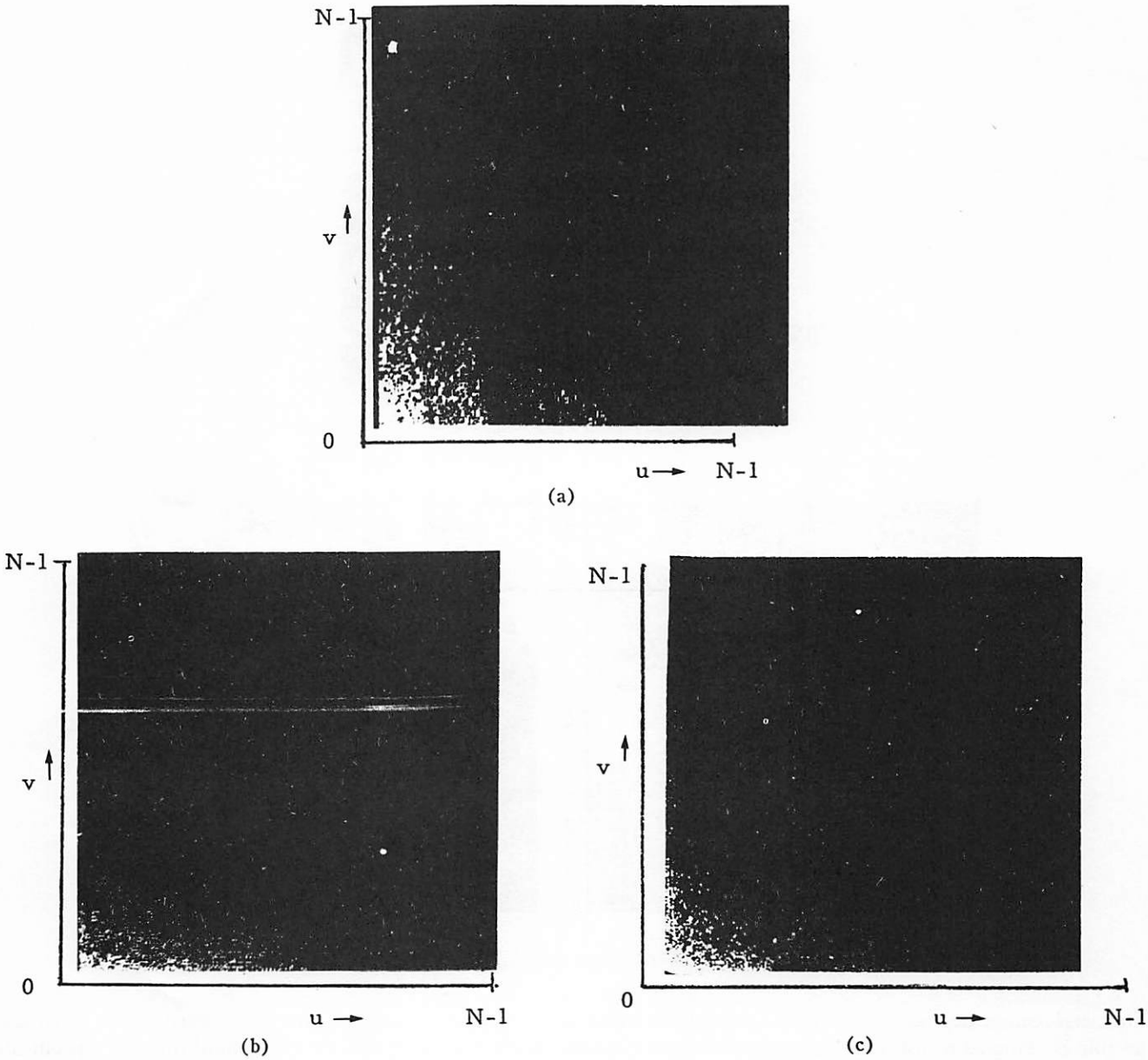


Fig. 3. Hadamard transforms of Surveyor scenes (displays of the logarithm of the magnitude of the Hadamard transform); (a) footpad, (b) boom, and (c) experimental box.

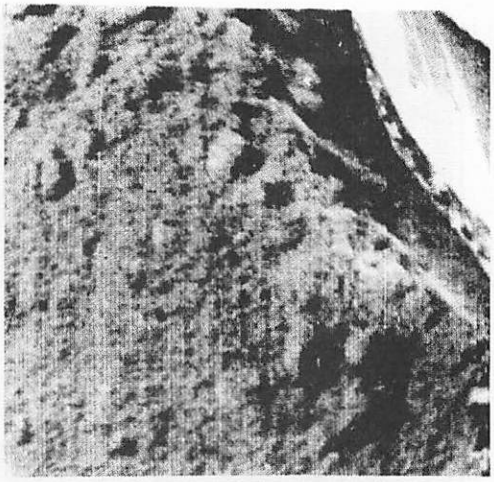
$$q(x, y, u, v) \equiv \sum_{i=0}^{n-1} [g_i(u)x_i + g_i(v)y_i]$$

and

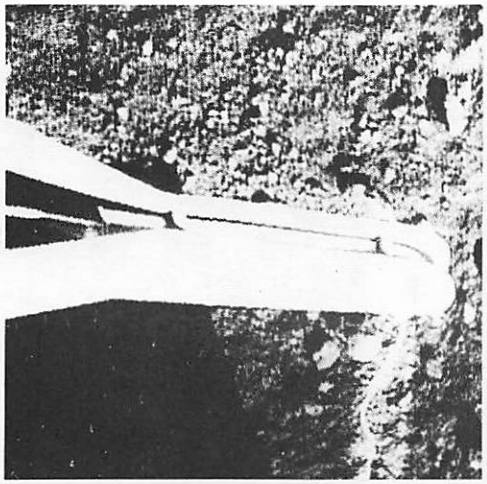
$$\begin{aligned} g_0(u) &\equiv u_{n-1} \\ g_1(u) &\equiv u_{n-1} + u_{n-2} \\ g_2(u) &\equiv u_{n-2} + u_{n-3} \\ &\vdots \\ g_{n-1}(u) &\equiv u_1 + u_0. \end{aligned}$$

The two-dimensional Hadamard transform may be computed in either natural or ordered form with an algorithm analogous to the fast Fourier transform computer algorithm. A computational algorithm for an ordered transform is considered later. All experimental results in this paper have been obtained with the Hadamard transform given by (13).

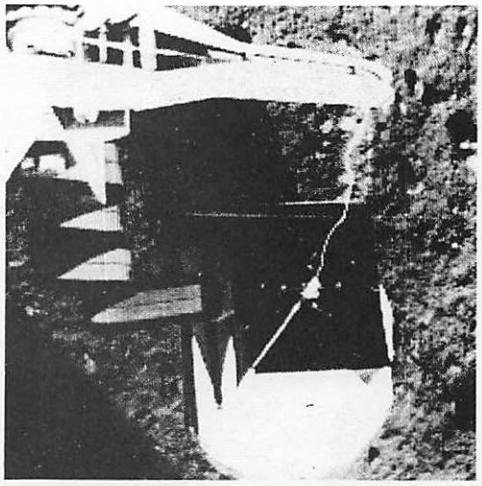
Experiments have been performed to determine the nature of the two-dimensional Hadamard transform of an image, and to assess the effects of a double Hadamard transformation. Fig. 3 contains cathode ray tube displays of the logarithm of the magnitude of the ordered Hadamard transform of several Surveyor scenes. In these photographs the origin (zero sequency) appears in the lower left corner. As the sequency increases, the magnitudes of the samples tend to decrease. This is an indication that there are relatively few high-amplitude brightness transitions between elements in the original scenes. Fig. 4 shows the double Hadamard transforms of the three scenes. There is no



(a)



(b)



(c)

Fig. 4. Double Hadamard transforms of Surveyor scenes; (a) footpad, (b) boom, and (c) experimental box.

noticeable image degradation between the originals and the double Hadamard transforms [compare, for example, Fig. 1(a) and Fig. 4(a)].

IV. Properties of Hadamard Image Transforms

The Hadamard transform has several interesting properties. The most important properties from the standpoint of image coding are dynamic range, conservation of energy, and entropy.

The zero sequency term

$$F(0,0) = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y) \tag{14}$$

is a measure of the average brightness of a scene. If $f(x, y)$ is a positive real function, then the maximum possible value for the zero sequency term is N^2A where A is the maximum value of $f(x, y)$. All Hadamard domain samples other than the zero sequency sample range between $\pm N^2A/2$. The magnitude of the zero sequency term is a bound for the magnitude of all other Hadamard domain samples.

A conservation of energy property exists between the spatial domain and the Hadamard domain. Specifically,

$$\sum_{x=0}^{N-1} \sum_{y=0}^{N-1} |f(x,y)|^2 = \frac{1}{N^2} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} |F(u,v)|^2. \tag{15}$$

This equation is analogous to Parseval's relationship for the Fourier transform [23]. The implication of this equation for image coding is that if a few of the Hadamard domain samples are of large magnitude, then many of the remaining samples must necessarily be of very low magnitude. Conceivably, the low-magnitude samples may be discarded, as is possible with the Fourier transform coding technique, to obtain a bandwidth reduction.

If $f(x, y)$ is considered to be a random two-dimensional function with a specified entropy, then the entropy of $F(u, v)$ is the same as the entropy of $f(x, y)$ since the Jacobian of the transformation matrix is unity [24]. Hence, with proper coding it is possible to transmit either a scene or its Hadamard transform over a channel with the same channel capacity.

V. Computational Algorithm

The computation of (13) is performed in two steps. First a one-dimensional Hadamard transform is taken along each row of the array $f(x, y)$ yielding

$$F(u, y) = \sum_{x=0}^{N-1} f(x, y)(-1)^{\sum_{i=0}^{n-1} g_i(u)x_i} \tag{16}$$

Then a second one-dimensional Hadamard transform is taken along each column of $F(u, y)$ giving the desired result

$$F(u, v) = \sum_{y=0}^{N-1} F(u, y)(-1)^{\sum_{i=0}^{n-1} g_i(v)y_i} \tag{17}$$

Computation of the one-dimensional Hadamard transform by brute force methods requires N^2 operations where an operation is either an addition or subtraction. An algorithm for obtaining the one-dimensional Hadamard transform in $N \log_2 N$ operations has been developed. The algorithm is quite similar to the fast Fourier transform algorithm; computational savings are realized by storage of intermediate results [8]. A "fast" Hadamard transform algorithm was outlined in 1937 by Yates [25]. In 1958, Good described a matrix decomposition technique which can be implemented to perform the Hadamard transform with $N \log_2 N$ operations [26].

Fig. 5 illustrates the computations performed for our one-dimensional Hadamard transformation with eight data points. The data points are arranged in a column at level 3 and then summed by pairs to produce intermediate results for level 2. A dotted line linking two modes indicates that the data point at the higher level is multiplied by minus one before addition, or equivalently, the data point forms the subtrahend of a subtraction operation. Operations follow the tree graph to level 0 which is the ordered Hadamard transform of $f(x)$. There are two operations performed at each node of levels 1, 2, and 3 yielding a total of 8 $\log_2 8 = 24$ operations.

The fast Hadamard transform algorithm performs all of the operations indicated in Fig. 5, but in a certain selected order. All operations of level k are not completed before proceeding to level $k-1$, but rather operations are performed according to a sieving sequence. Fig. 6 describes the basic sequence computations. The first sequence is the "S" sequence in which the sum of all data points is formed to produce $F(0)$. A "1" sequence subtracts the lower node from the upper node of a pair of nodes at level $k-1$ to produce a result at level k. In the "2" sequence, operations begin at level $k-2$ where pairs are subtracted from one another to produce the results of level $k-1$ which in turn are added together. The "3" sequence and higher sequences to the "n" sequence follow directly. Fig. 6 also indicates the storage requirements for the computational procedure.

Original data are stored in a block of N words corresponding to level n. Intermediate results are stored $n-1$ different blocks of sizes 2^{n-1} to 2^2 words. These storage locations correspond to levels $n-1$ to 1 in the computation procedure.

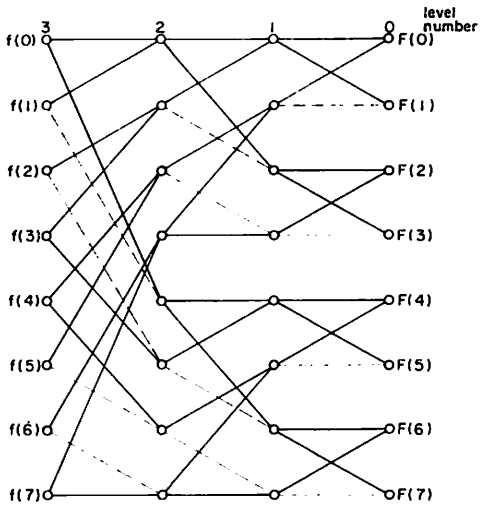


Fig. 5. Computations for one-dimensional third-order Hadamard matrix.

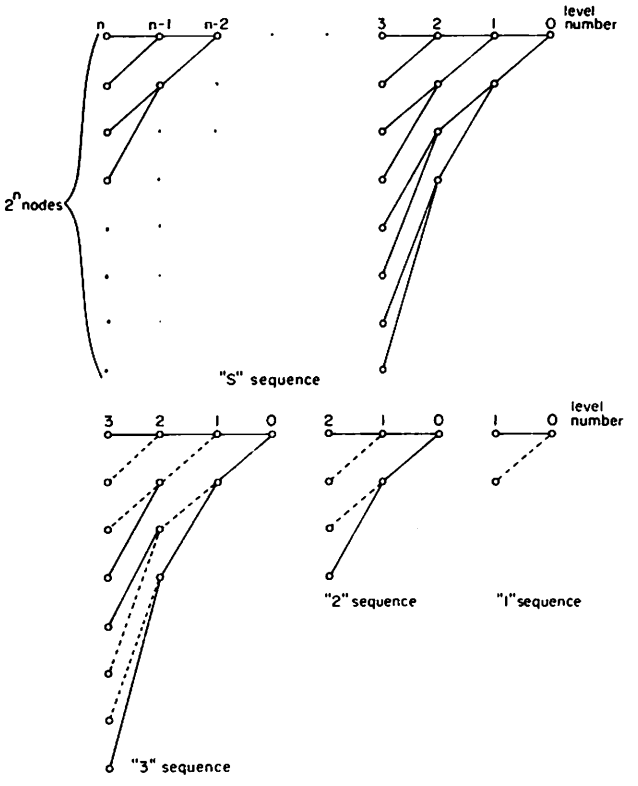


Fig. 6. Hadamard transform computational sequences.

Computation of the Hadamard transform begins with the "S" sequence which computes $F(0)$. Subsequent calculations are controlled by the following sieving sequence of integers building up to integer n:

$$\{1, 2, 1, 3, 1, 2, 1, 4, 1, 2, 1, 3, 1, 2, 1, \cdots, 1, n, 1, 2, 1, \cdots, 1, 2, 1, 3, 1, 2, 1\}.$$

For example, after the "S" sequence, the "1" sequence computes $F(1)$ using the intermediate results stored in level 1. Then, the "2" sequence computes $F(2)$ using the intermediate results in level 2.

	Level 3	Level 2	Level 1	Level 0	
f(0)	A	I	M	F(0)	P
f(1)	B	J	N	F(1)	Q
f(2)	C	K		F(2)	R
f(3)	D	L		F(3)	S
f(4)	E			F(4)	T
f(5)	F			F(5)	U
f(6)	G			F(6)	V
f(7)	H			F(7)	W

Fig. 7. Storage locations for computation of one-dimensional third-order Hadamard transform.

Sequence number	Level number	Operation	Contents of storage locations	Result
—	3	—	f(0) A f(1) B f(2) C f(3) D f(4) E f(5) F f(6) G f(7) H	
S	2	A+B → I	f(0) + f(1) I	
		C+D → J	f(2) + f(3) J	
		E+F → K	f(4) + f(5) K	
		G+H → L	f(6) + f(7) L	
	1	I + J → M	f(0) + f(1) +f(2) +f(3) M	
		K+L → N	f(4) + f(5) +f(6) +f(7) N	
	0	M+N → P	f(0) +f(1) +f(2) +f(3) +f(4) +f(5) +f(6) +f(7) P	F(0)
1	0	M-N → Q	f(0) +f(1) +f(2) +f(3) -f(4) -f(5) -f(6) -f(7) Q	F(1)
2	1	I -J → M	f(0) +f(1) -f(2) -f(3) M	
		L-K → N	-f(4) -f(5) +f(6) + f(7) N	
	0	M+N → R	f(0) +f(1) -f(2) -f(3) -f(4) -f(5) +f(6) +f(7) R	F(2)
1	0	M-N → S	f(0) +f(1) -f(2) -f(3) +f(4) +f(5) -f(6) -f(7) S	F(3)
3	2	A-B → I	f(0) -f(1) I	
		D-C → J	-f(2) +f(3) J	
		E-F → K	f(4) -f(5) K	
		H-G → L	-f(6) +f(7) L	
	1	I + J → M	f(0) -f(1) -f(2) +f(3) M	
		K+L → N	f(4) -f(5) -f(6) +f(7) N	
	0	M+N → T	f(0) -f(1) -f(2) +f(3) +f(4) -f(5) -f(6) +f(7) T	F(4)
1	0	M-N → U	f(0) -f(1) -f(2) +f(3) -f(4) +f(5) +f(6) -f(7) U	F(5)
2	1	I -J → M	f(0) -f(1) +f(2) -f(3) M	
		L-K → N	-f(4) +f(5) -f(6) +f(7) N	
	0	M+N → V	f(0) -f(1) +f(2) -f(3) -f(4) +f(5) -f(6) +f(7) V	F(6)
1	0	M-N → W	f(0) -f(1) +f(2) -f(3) +f(4) -f(5) +f(6) -f(7) W	F(7)

Fig. 8. Computer operations for computation of one-dimensional third-order Hadamard matrix.

Fig. 7 gives the storage locations for computation of a one-dimensional third-order Hadamard transform. The computer operations for this example are listed in Fig. 8.

Since computation of the Hadamard transform requires only real additions and subtractions, whereas the Fourier transform computations are composed of complex multiplications, additions, and subtractions, considerable computational savings are afforded with the Hadamard transform. Both the Fourier and Hadamard transforms have been programmed on a TRW-530 digital computer. For a 256 by 256 point scene, the Fourier transform can be computed in 20 minutes and the Hadamard transform in 3 minutes.

VI. Hadamard Domain Quantization

The Hadamard transform of an image must be quantized for subsequent digital coding and transmission over a channel. In order to quantize Hadamard domain samples it is necessary to determine the number and placement of quantization levels. Both of these factors affect the degree of image degradation caused by quantization errors and the entropy of the quantized symbols. The approach taken here has been to select quantum levels to maximize the source entropy, and then to subjectively evaluate image quality.

Two models for the probability density of Hadamard domain samples that have been considered are shown in Fig. 9. The first is the linear quantization model, commonly used for spatial domain quantization, in which the quantum levels are equispaced over some maximum range. The second model is the Gaussian quantization model in which the quantization levels are chosen to partition the Gaussian curve into equal areas.

In the Gaussian quantization model the probability density of Hadamard domain samples is assumed to be of the form

$$P_H(Z) = [2\pi\sigma_H^2(u,v)]^{-\frac{1}{2}} \exp\left\{-\frac{Z^2}{2\sigma_H^2(u,v)}\right\} \tag{18}$$

where the variance function $\sigma_H^2(u, v)$ controls the placement of quantization levels as a function of the Hadamard spatial frequency. Referring to Fig. 3, it is evident that the variance function should be highest at the origin in the Hadamard domain, be circularly symmetric over one quadrant, and decrease monotonically toward the higher Hadamard dimensions. A convenient two-dimensional function possessing these properties is the Gaussian curve described by

$$\sigma_H^2(u,v) = S \exp\left\{-\frac{u^2+v^2}{p}\right\} \tag{19}$$

where S is an amplitude scaling constant and p is a spread control constant.

Reconstructions of the Hadamard transform of the footpad scene with the linear and Gaussian quantizer are shown in Fig. 10. In both cases, 64 quantization levels have been employed. Results with the linear quantizer are poor because of the large quantization errors at high sequencies. The reconstruction using the Gaussian quantizer with the variance parameter $p=1500$, on the other hand, shows negligible image degradation. Fig. 11 contains additional examples of reconstructions with the 64 level Gaussian quantizer with $p=1500$. Tests have been conducted to determine the effect of fewer quantization levels. Fig. 12 shows reconstructions of the footpad scene with the Gaussian quantizer with $p=1500$ for 32 and 16 quantization levels. The loss of resolution apparent in these pictures is due to the quantization errors at high sequencies.

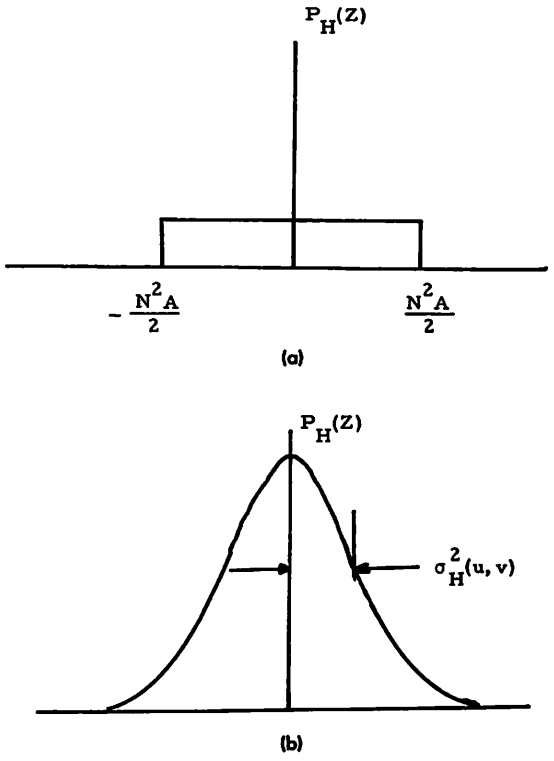
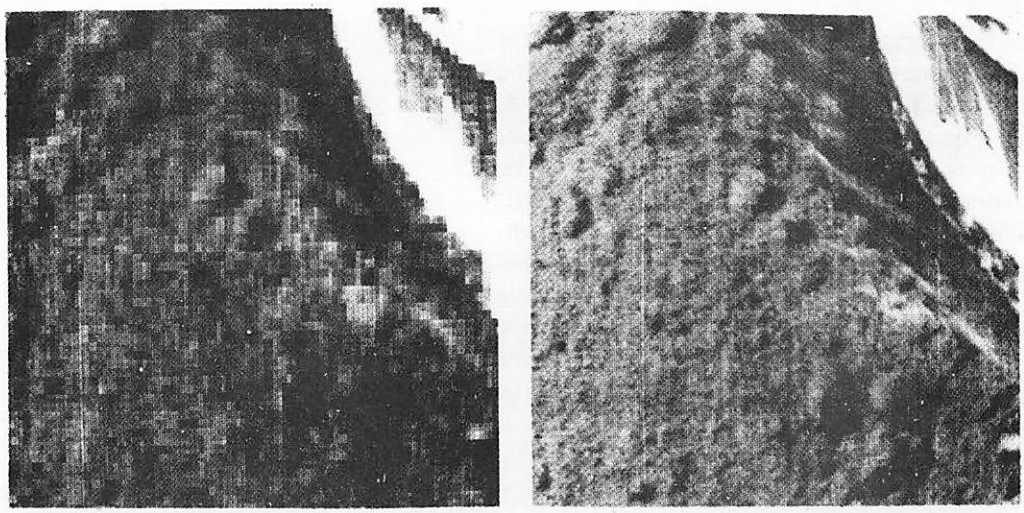


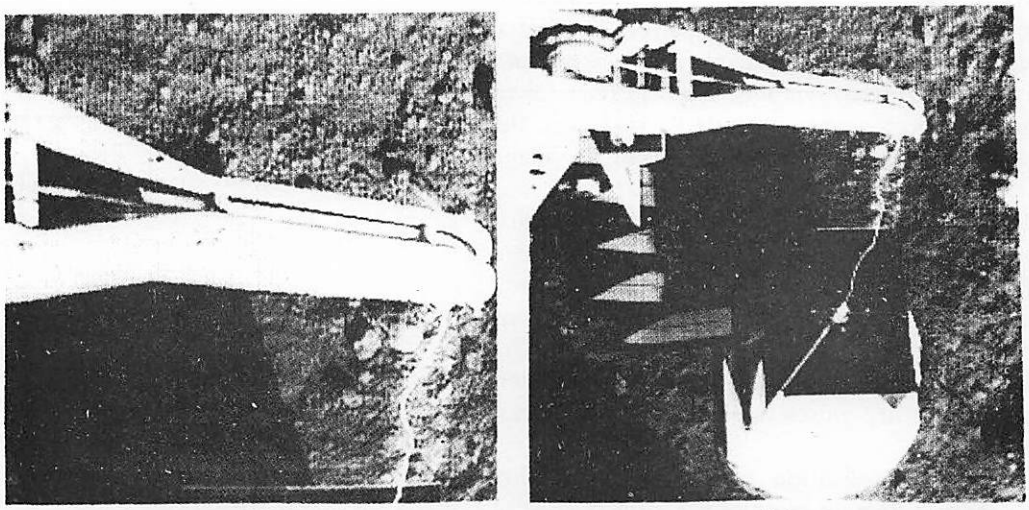
Fig. 9. Hadamard domain quantization rules; (a) linear, and (b) Gaussian.

If the Gaussian quantization rule is to be practical it is imperative that a variance function can be chosen for a wide class of scenes without detailed knowledge of the content of these scenes. Figs. 10 and 11 show that the Gaussian variance function, with the same variance parameter ($p=1500$), provides satisfactory reconstructions for three different scenes. It is also of interest to determine the effect of changes in the variance parameter. Fig. 13 contains reconstructions using the Gaussian quantizer with $p=500$, 1000, 2000, and 5000 and with 64 quantization levels. For $p=1000$ and $p=2000$ the reconstructions are quite satisfactory. At $p=500$ a distinct checking effect appears. This checking is due to quantization errors of large-magnitude low-sequency Hadamard samples. The quantizer



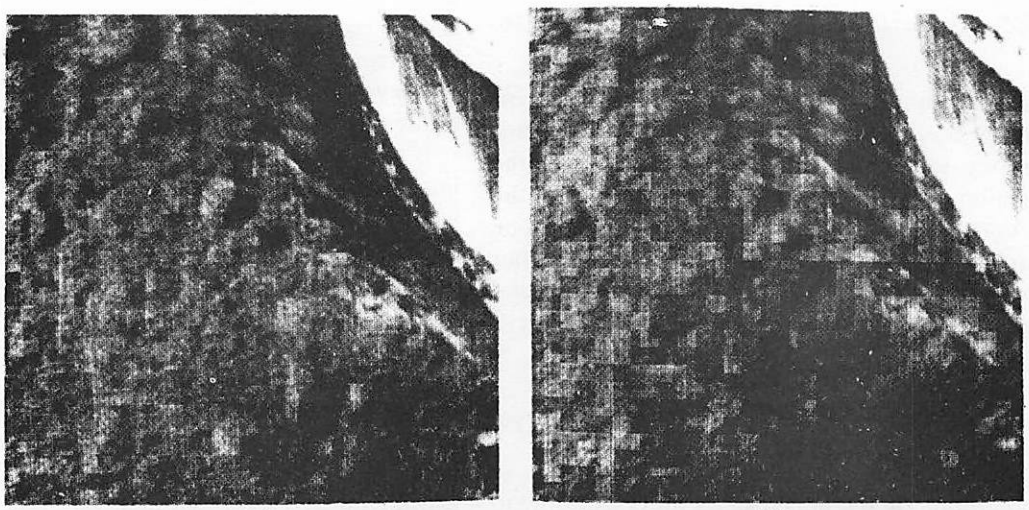
(a) (b)

Fig. 10. 64-level quantization of footpad; (a) linear quantization, and (b) Gaussian quantization, $p=1500$.



(a) (b)

Fig. 11. 64-level Gaussian quantization of boom and experimental box, $p=1500$; (a) boom, and (b) experimental box.



(a) (b)

Fig. 12. 32- and 16-level Gaussian quantization of footpad, $p=1500$; (a) 32-level quantization, and (b) 16-level quantization.

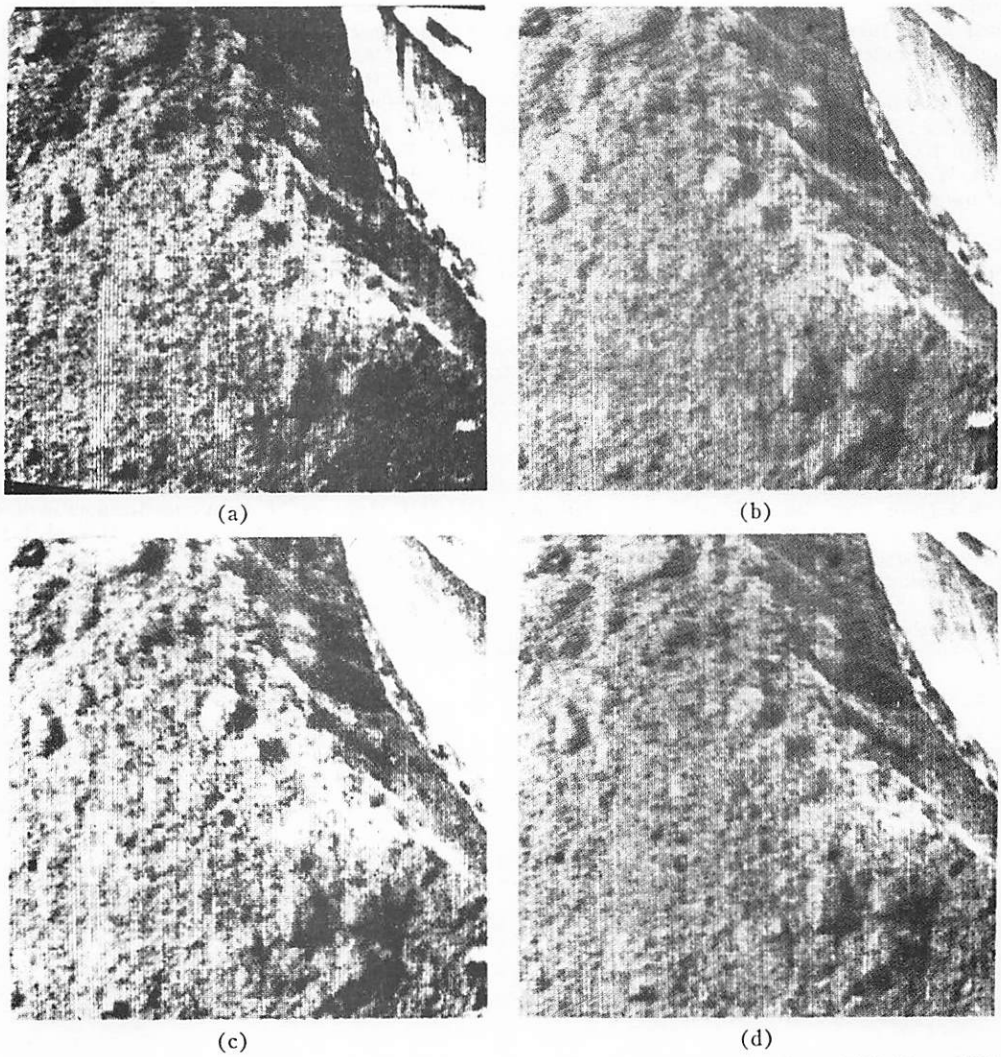


Fig. 13. 64-level Gaussian quantization of footpad with different variance functions; (a) $p=500$, (b) $p=1000$, (c) $p=2000$, (d) $p=5000$.

has clipped the magnitudes of these samples to a relatively low level. At the other extreme, for $p=5000$, the high-sequency samples, which are of low magnitude, are all given the same magnitude of the first quantum level. The result is a loss of resolution.

Conclusions

It has been shown that it is possible to transform an image by a Hadamard matrix operator which is similar to the two-dimensional Fourier transform. Use of the Hadamard rather than the Fourier transform results in an order of magnitude speed increase. The Hadamard samples can be quantized to as few as 64 levels, using a Gaussian quantization rule. Reconstruction of the quantized transform by a second two-dimensional Hadamard transform introduces negligible image degradation. With conventional PCM coding it is possible to transmit the Hadamard transform of an image at the same bit rate as the image itself. The potential advantages of the Hadamard image coding technique are a tolerance of channel errors and the possibility of bandwidth reduction.

Acknowledgment

Professor L. R. Welch of the University of Southern California suggested series representations of the Hadamard transform given by (11) and (13). The assistance of D. Ketcham of USC, who developed many of the experimental computer programs, and J. Pepin also of USC, who carried out the experiments, is greatly appreciated. Original digitized images were provided by F. Billingsley of the Jet Propulsion Laboratory.

References

[1] H. C. Andrews and W. K. Pratt, "Fourier transform coding of images," *Hawaii Internat'l Conf. on System Sciences*, pp. 677–679, January 1968.

[2] H. C. Andrews and W. K. Pratt, "Television bandwidth reduction by Fourier image coding," presented at the Soc. of Motion Picture and Television Engrs., 103rd Tech. Conf., May 1968.

[3] H. C. Andrews and W. K. Pratt, "Television bandwidth reduction by

encoding spatial frequencies," to be published in *J. Soc. Motion Picture and Television Engrs.*

[4] J. W. Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex Fourier series," *Math. of Comp.*, vol. 19, pp. 297–301, 1956.

[5] Subcommittee on Measurement Concepts, Audio and Electroacoustic Group, "What is the fast Fourier transform?" *Proc. IEEE*, vol. 55, pp. 1664–1673, October 1967.

[6] J. W. Cooley, P. A. W. Lewis, and P. D. Welch, "Historical notes on the fast Fourier transform," *Proc. IEEE*, vol. 55, pp. 1675–1677, October 1967.

[7] E. O. Brigham and R. E. Morrow, "The fast Fourier transform," *IEEE Spectrum*, vol. 4, pp. 63–70, December 1967.

[8] H. C. Andrews, "A high speed algorithm for the computer generation of Fourier transforms," *IEEE Trans. Computers* (*Short Notes*), vol. C-17, pp. 373–375, April 1968.

[9] W. M. Gentlemen, "Matrix multiplication and fast Fourier transformations," *Bell Sys. Tech. J.*, vol. 47, pp. 1099–1103, July–August 1968.

[10] W. R. Crowther and C. M. Rader, "Efficient coding of vocoder channel signals using linear transformation," *Proc. IEEE* (*Letters*), vol. 54, pp. 1594–1595, November 1966.

[11] J. Hadamard, "Resolution d'une question relative aux determinants," *Bull. Sci. Math.*, ser. 2, vol. 17, pt. 1, pp. 240–246, 1893.

[12] H. J. Ryser, *Combinatorial Mathematics*. New York: Wiley, 1963.

[13] R. E. A. C. Paley, "On orthogonal matrices," *J. Math. Phys.*, vol. 12, pp. 311–320, 1933.

[14] J. Williamson, "Hadamard's determinant theorem and the sum of four squares," *Duke Math. J.*, vol. 11, pp. 65–81, 1944.

[15] L. Baumert, S. W. Golomb, and M. Hall, Jr., "Discovery of an Hadamard matrix of order 92," *Bull. Am. Math. Soc.*, vol. 68, pp. 237–238, 1962.

[16] H. F. Harmuth, "A generalized concept of frequency and some applications," *IEEE Trans. Information Theory*, vol. IT-14, pp. 375–382, May 1968.

[17] J. L. Walsh, "A closed set of orthogonal functions," *Am. J. Math.*, vol. 55, pp. 5–24, 1923.

[18] N. J. Fine, "On the Walsh functions," *Trans. Am. Math. Soc.*, vol. 65, pp. 372–414, 1949.

[19] N. J. Fine, "The generalized Walsh functions," *Trans. Am. Math. Soc.*, vol. 69, pp. 66–77, 1950.

[20] G. W. Morgenthaler, "On Walsh-Fourier series," *Trans. Am. Math. Soc.*, vol. 84, pp. 472–507, 1957.

[21] K. W. Henderson, "Some notes on the Walsh functions," *IEEE Trans. Electronic Computers* (*Correspondence*), vol. EC-13, pp. 50–52, February 1964.

[22] H. Rademacher, "Einige Sätze von allgemeinen Orthogonal-Funktionen," *Math. Ann.*, vol. 87, pp. 122–138, 1922.

[23] A. Papoulis, *The Fourier Integral and Its Applications*. New York: McGraw-Hill, 1962, p. 27.

[24] H. C. Andrews, "Entropy considerations in the frequency domain," *Proc. IEEE* (*Letters*), vol. 46, pp. 113–114, January 1968.

[25] F. Yates, *The Design and Analysis of Factorial Experiments*. Harpenden: Imperial Bureau of Soil Analysis, 1937.

[26] I. J. Good, "The interaction algorithm and practical Fourier analysis," *J. Royal Stat. Soc.*, ser. B, vol. 20, pp. 361–372, 1958.