FINAL REPORT

18 December 1972 - 17 May 1973

SOURCE ENCODING OF IMAGES FOR EFFICIENT TRANSMISSION

NASA Grant NGR 05-003-538

Principal Investigators:

V. R. Algazi and D. J. Sakrison

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

The purpose of this grant was the continuation of some work on Source Encoding of Images for Efficient Transmission carried under sponsorship of the National Aeronautics and Space Administration for the last few years. The small Grant NGR-05-003-538 was important in assuring the continuity of our work which is now being pursued under sponsorship of the National Science Foundation. Two pieces of work have been completed with partial support of this grant and constitute our final report.

1. Comments on "Basic Restricted Transformation and Performance Measure for Spectral Representations," by V. R. Algazi and B. J. Fino, to appear in the IEEE Transactions on Information Theory in July 1973.
2. "A Unified Treatment of Discrete Unitary Transforms with a Fast Algorithm," by B. J. Fino and V. R. Algazi, submitted for publication.

Copies of these two publications are attached.

Comment on "Basis Restricted Transformations and Performance Measure for Spectral Representations".

V. R. ALGAZI
B. J. FINO

ABSTRACT: In this correspondence, we comment on some of the shortcomings, for source encoding problems, of the measure of distance between orthonormal representations proposed by Pearl [1].

We further examine, using rate distortion theory, whether rates achievable for a given distortion level by the use of basis-restricted transformations have some inherent dissimilarity independently of input statistics and of other implied parameters.

Research sponsored in part by the National Aeronautic and Space Administration, Grant NASA-NGR-05-003-538.

Comment on "Basis Restricted Transformations and Performance Measure for Spectral Representations".

Pearl [1] has proposed a distance measure to characterize "inherent dissimilarities" between orthonormal transformations. With this measure the distance between orthonormal transformations is "independent of input statistics". One of the applications mentioned for basis restricted transformations is to the source encoding of random processes. In this correspondence we comment on some of the shortcomings of Pearl's distance measure when applied to source encoding problems. We further examine, using rate distortion theory, whether rates achievable for a given distortion level by the use of basis-restricted transformations have some inherent dissimilarity independently of input statistics and of other implied parameters.

We note first that dissimilarity between two transformations is defined by Pearl for the class of random processes which have a diagonal correlation matrix for one of the transformations. The transformation is thus the Karhunen-Loeve expansion of a random process adapted to the specific transformation. Under these circumstances independence on input statistics achieved by averaging over a class of processes cannot be dissociated of the class of processes considered. Moreover this construction does not provide any insight into a common question of interest in the use of basis-restricted transformations for source encoding: Given a random process with known correlation matrix how do different transformations rank in the encoding of this random process? This is the specific question that Pearl refers to when he comments on the asymptotic distance between the Walsh-Hadamard and the Fourier transformation for large block size. However Pearl's distance measure also indicates a diverging distance between the Karhunen Loeve (which is then the Kronecker or natural transformation) and Fourier transformation for large block size. On this basis one cannot feel confident that Pearl's distance between Walsh and Fourier transforms provides an answer to a

common conjecture.

The evaluation of the relative performance of orthonormal transformations in source encoding problems can be done by assuming that the expansion coefficients are encoded independently and the rate distortion bound evaluated for each coefficient. This approach was used by Goblick and Holsinger [2] to evaluate the performance of the Fourier expansion for source encoding and has been extended to other "basis-restricted transformations" by Pearl.

For example we show in Figure 1 such rate distortion functions for a Gaussian Markov random process with correlation function $R(n) = e^{-\alpha|n|}$ (1) $n = 0, \pm 1, \pm 2$ --. These curves are for $\alpha = + 0.05$ and clock size $N = 32$. Similar curves are given by Pearl, Andrews and Pratt for $N = 256$ [3]. Note that the Haar and Walsh-Hadamard transformations behave similarly, in contradiction to Pearl's distance. The similarity of performance can be explained by exploiting the formal relations between the Haar and Walsh Hadamard transformations [4]. Note also that for the semi-logarithmic scale used for the mean square distortion the rate-distortion per sample curves are parallel straight lines. This occurs for mean-square distortions lower than the smallest variance of expansion coefficients since we have then

$$R(\varepsilon) = \frac{1}{2N} \log_2 \frac{\prod_{i=1}^{N} \sigma_i^2}{\varepsilon^N} \qquad (2)$$

Therefore $d_{R1,\,R2} = R_1(\varepsilon) - R_2(\varepsilon) = \frac{1}{N} \log_2 \prod_{i=1}^{N} (\sigma_{1i}/\sigma_{2i})$ in which subscripts 1 and 2 denote two distinct orthonormal transformations. Note that $d_{R1,R2}$ is no longer dependent on the specific ε.

To illustrate the dependence of source statistics we show in figure (2) d_R as a function of α, defined in (1) for the fixed block size $N = 32$. It is noteworthy that the curves for Walsh Hadamard and Fourier transformations intersect. Thus no general conclusion can be obtained on the relative merit

of these two orthonormal transformations in source encoding problems. Averaging on some set of statistics is possible but the use of the results is dependent on the specific application. As shown in Figure (3), when the block size increases, distance $d_{R1, R2}$ between the Fourier and Karhunen Loeve transforms decreases to zero as expected. It can be shown that the distance between the Haar and Karhunen-Loeve transforms increases to a limit and it is conjectured that the Walsh Hadamard transform has a similar limiting behavior.

V. Ralph Algazi
Department of Electrical Engineering
University of California, Davis

Bernard J. Fino
Department of Electrical Engineering
and Computer Science
University of California, Berkeley

References:

[1] J. Pearl, "Basis-Restricted Transformations and Performance Measures for Spectral Representations", IEEE Trans. on Inf. Theory, Vol. IT-17, November 1971, pp. 751-752.

[2] T. Goblick and J. Holsinger, "Analog Source Digitization: A Comparison of Theory and Practice", IEEE Trans. on Inf. Theory, Vol. IT-13, April 1967, pp. 323-326.

[3] J. Pearl, H. C. Andrews, W. K. Pratt, "Performance Measures for Transform Data Coding", IEEE Trans. on Communications, Vol. COM-20, No. 3, June 1972, pp. 411-414.

[4] B. Fino, "Relations between Haar and Walsh Hadamard Transforms", Proceeding of the IEEE, Vol. 60, May 1972, pp. 647-648.

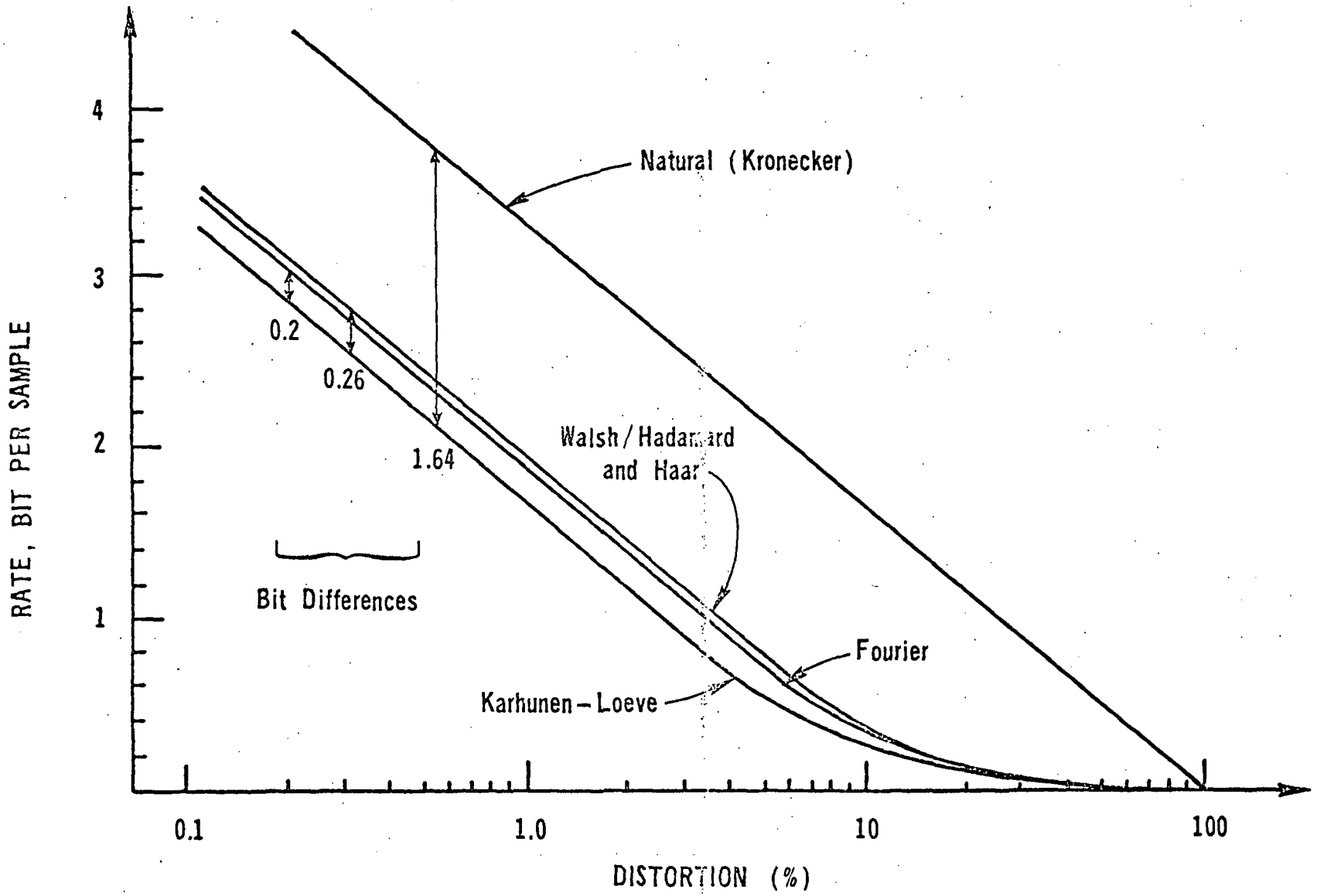


Figure 1. Rate Distortion Functions
$\alpha = 0.05$, Block Size = 32

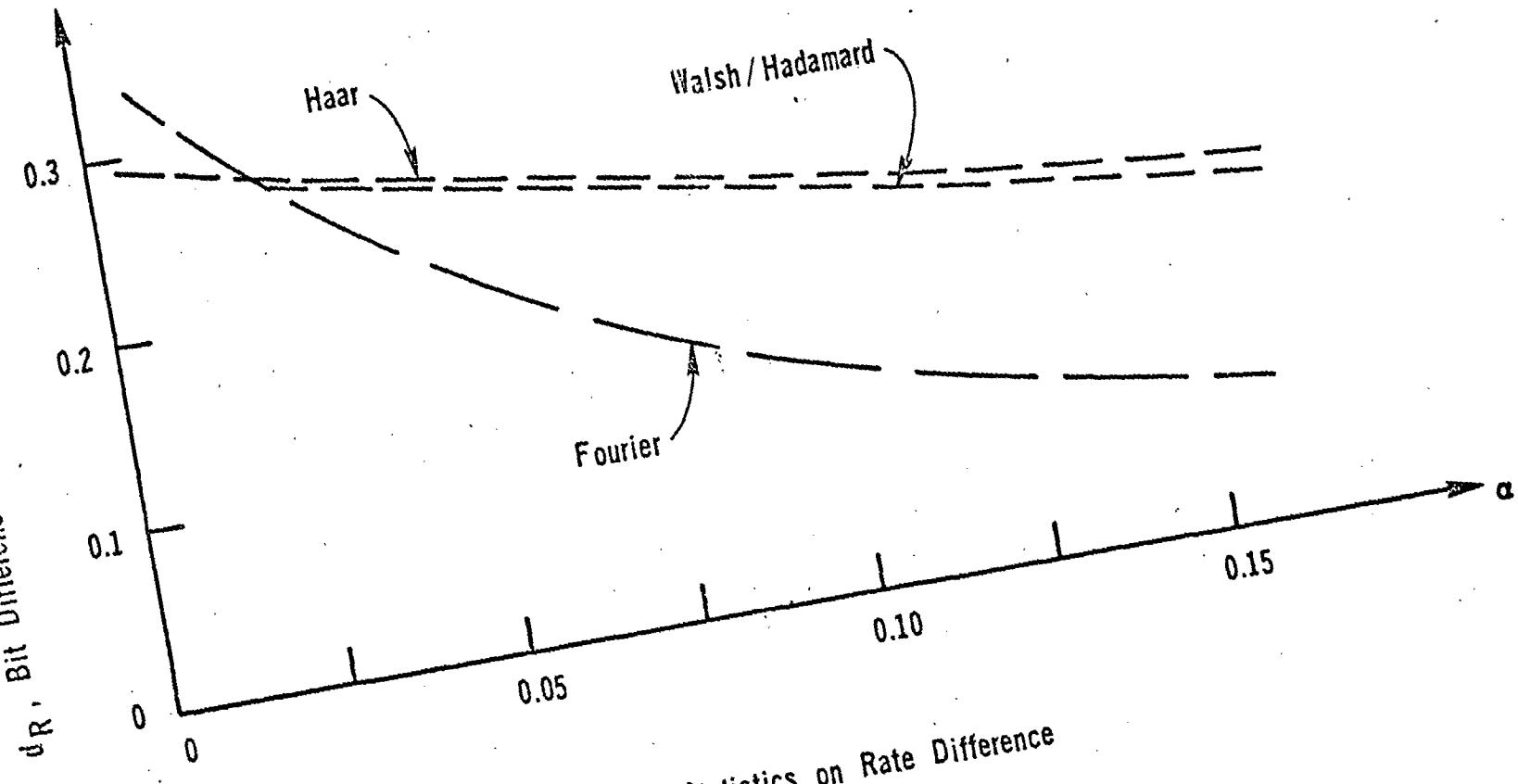


Figure 2. Effect of Statistics on Rate Difference
Block Size = 32

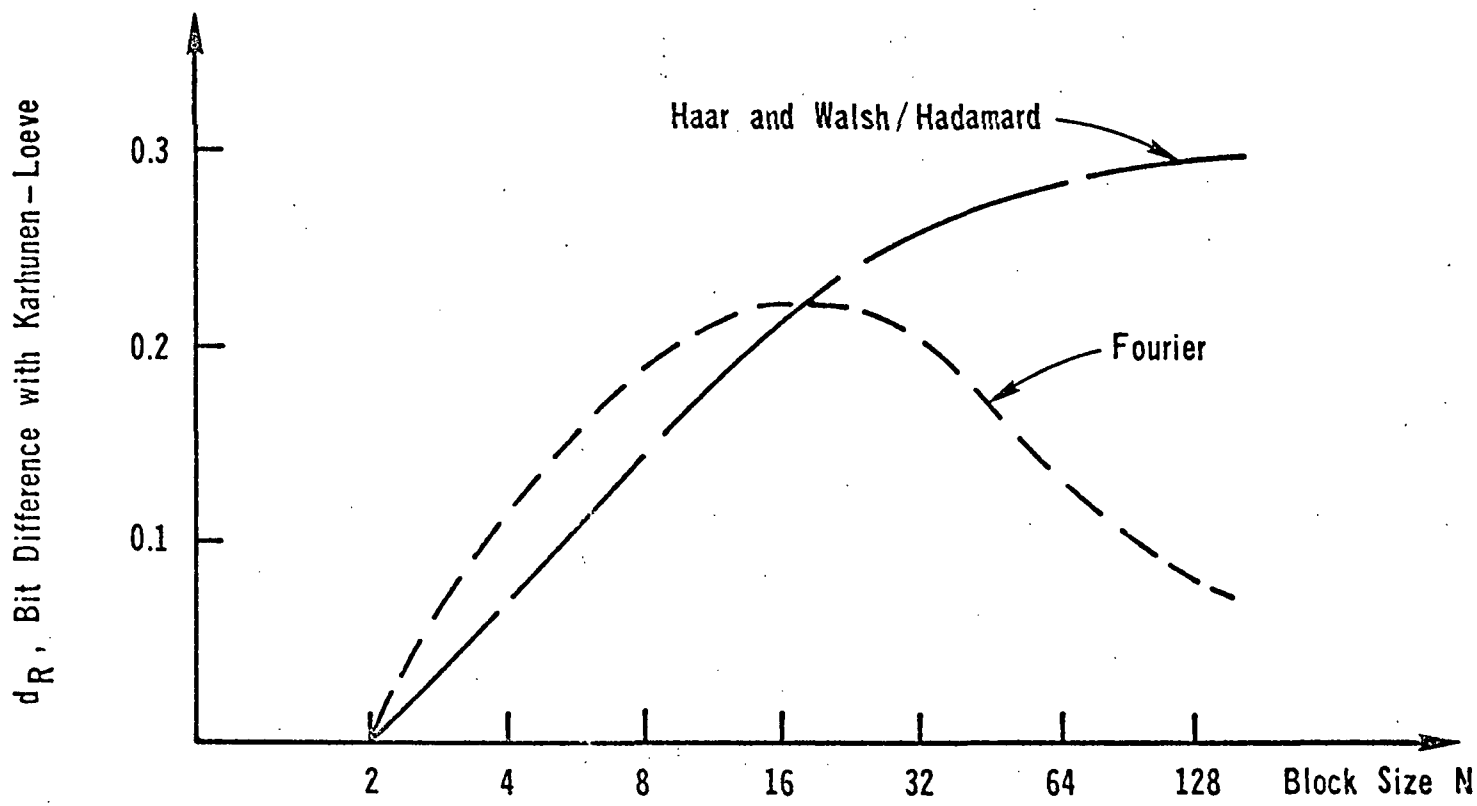


Figure 3. Effect of Block Size on Rate Difference
$\alpha = 0.05$

A UNIFIED TREATMENT OF DISCRETE UNITARY TRANSFORMS WITH A FAST ALGORITHM

by

Bernard J. Fino[†] and V. Ralph Algazi[††]

Abstract

A set of recursive rules which generate unitary transforms with a fast algorithm are presented. For each rule, simple relations gives the number of elementary operations required by the fast algorithm. The common Fourier, Walsh-Hadamard (W-H), Slant and Haar transforms are expressed with these rules. The framework developed allows the introduction of generalized transforms which include all common transforms in a large class of "identical computation transforms." A systematic and unified view is provided for unitary transforms which have appeared in the literature and for a number of new transforms of potential interest. Generalization to complex and multidimensional unitary transforms is considered.

Research sponsored in part by the National Aeronautic and Space Administration, Grant NASA-NGR-05-003-538, and in part by the National Science Foundation, Grant NSF-GK-37282

[†]Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, California 94720.

[††]Department of Electrical Engineering, University of California, Davis California 95616

A UNIFIED TREATMENT OF DISCRETE UNITARY TRANSFORMS WITH A FAST ALGORITHM

Contents

Index Terms:

Discrete transforms, fast algorithms, fast Fourier transforms, fast generalized transforms, generalized Kronecker product, Haar transform, identical computation transforms, slant transform, unitary transforms and matrices, Walsh-Hadamard transform.

Introduction

The dissemination of the Fast Fourier Transform algorithms, originally introduced by Good [1], and known as Cooley-Tukey [2] and Sande-Tukey [3] algorithms, has resulted in a large extension in the range of applications of the well known Fourier transform. Recently the Walsh-Hadamard transform, also with a fast algorithm [4] has drawn considerable interest [5]. The Haar transform although closely related to the Walsh-Hadamard transform [6] and potentially of interest [7], has received much less attention. These transforms have been used successfully for error free signal representation [8], pattern classification [4], [9], speech signal encoding [10] and above all for picture encoding [11], [12], [13]. Only a few transforms have been considered in these applications while many other transforms could be of interest. Some workers have considered the definition of generalized transforms and we mention the works by Andrews, et al [14], [15], [16], Rao, et al [17], [18] and Harmuth [19].

In this paper we present a unified view of discrete unitary transforms with a fast algorithm. A discrete unitary transform is characterized by a unitary matrix [T] such that $[T][T^{*t}] = [I]$ where * denote conjugate "t" transpose and [I] is the unity matrix of same order as [T], say N. For mathematicians a unitary matrix expresses a rotation of the orthonormal basis and preserves the Euclidian norm $\|V\|$ of a vector $\vec{V}$. In this N dimensional space, $\|V\| = \vec{V}.\vec{V}^{*t}$. In signal representation, this property means energy conservation and an easy expression of the mean square error when some components of the signal are ignored in the new base. The computation of the transformed vector $\vec{W}$ of $\vec{V}$ by the transform [T] such that $\vec{W} = [T]\vec{V}$ usually requires N^2 multiplications and N(N-1) additions.

For some specific transforms of interest such as the Fourier, Walsh-Hadamard transforms a fast algorithm has been found which requires fewer elementary operations. The analysis of these fast algorithms has been done by factorization of the matrix [T] into a set of largely sparse matrices, each expressing a stage of computation. This is the approach followed by Good [1] in his original paper which lead to the Fast Fourier Transform [2], [3] the Fast Walsh Transform [4] and other known fast transforms.

Here we consider recursive rules[1] for the generation of unitary transforms having a fast algorithm. These rules allow us to generate large classes of such transforms, many of which are new and possibly of interest, and to give general formulas for the number of elementary operations required by the corresponding fast algorithm.

1. Recursive Generative Rules:

We shall present three rules which generate a new unitary matrix from some original unitary matrices. For each rule we relate the number of elementary operations for the new transform to the number of elementary operations of the same type required by the original transforms. For rule 1 there is only one original matrix, for rule 2 two, and for rule 3 a set of original matrices.

[1] We denote by "rule" a set of operations performed in a prescribed order. We reserve the term "operation" for the elementary operations such as additions, multiplications, etc. which determine the computational complexity of a transform.

Rule 1: Operations on the columns of a unitary matrix:

Given a unitary matrix [T], two obvious operations on the columns yield another unitary matrix of some order:

a) permutation of the columns: This operation does not require any computation. In the computational process, this operation can be performed by applying the permutation to the coefficients of the input vector instead of the columns themselves.

b) multiplication of a column by a root of unity: This operation introduces a complex multiplication if the root of unity is not ± 1 or $\pm j$ ($j = \sqrt{-1}$) (see footnote 2).

These operations on the columns may be expressed by a matrix product [T] [D] with [D] such that $D_{ki} = e^{j\theta i}$ if column k is to be replaced by column i multiplied by the root of unity, $e^{j\theta i}$, and all other entries of [D] are null.

Rule 2: Rotation of rows by a unitary matrix

Consider a unitary matrix [T] of order N. The N row vectors form an orthonormal basis for S_N, the N dimensional space they span. m rows vectors of [T] form an orthonormal basis for subspace S_m. If these m vectors are rotated by a unitary matrix [U] of order m, we obtain a new orthonormal basis $\mathcal{B}$ for S_m. The remaining unchanged N-m rows of [T] are an orthonormal basis of the subspace S_{N-m} orthogonal to S_m and form with $\mathcal{B}$ a new orthonormal basis for S_N. Thus, the matrix [T'] obtained after rotation of the m rows by the unitary matrix [U] is unitary.

---

[2] Multiplications by ± 1 and $\pm j$ may be counted as operations if the hardware realization of the algorithm is not able to keep track of them. However, for the error analysis of the algorithm these multiplications, even if they are performed, do not introduce any error.

Some particular cases of interest are:

a) multiplication of the whole matrix by a unitary matrix of the same order

b) permutation of the rows (multiplication by a permutation matrix)

c) multiplication of a row by any root of unity.

The operation b) and c) can be represented by the matrix product [D] [T] where [D] is, as before, such that $D_{ik} = e^{j\theta i}$ if row i of T is replaced by row k multiplied by the root of unity, $e^{j\theta i}$, and all other entries of [D] are null.

Number of Elementary Operations:

If transforms T and U require respectively t and u elementary operations of a specific kind, it is obvious that the transform T' will require at most t' of these operations with

$$t' = t + u \qquad (1)$$

(It may happen that [T'] so generated has a simpler algorithm).

Equation (1) applies independently to any type of elementary operation, additions, real and complex multiplications as well as any other specific operation (e.g. shift, multiplication by $\sqrt{2}$ . . . etc.)

Rule 3: Generalized Kronecker Product:

Given two sets of unitary matrices, set $\{\mathcal{A}\}$ of m matrices $\left[A^i\right]$ (i=0,...,m-1) all of order n and set $\{\mathcal{B}\}$ of n matrices $\left[B^i\right]$ (i=0,...,n-1), all of order m, we define the generalized Kronecker product of the sets $\{\mathcal{A}\}$ and $\{\mathcal{B}\}$, denoted $\{\mathcal{A}\}\otimes\{\mathcal{B}\}$ to be the square matrix [C] of order (mn) such that

$$C_{i,j} = C_{um+w,\ u'm+w'} = A^{w}_{uu'} \cdot B^{u'}_{ww'} \tag{2}$$

with $i = um+w$ $\quad u, u'=0, \ldots, n-1$

$j = u'm+w'$ $\quad w, w'=0, \ldots, m-1$

a) <u>[C] is a unitary matrix:</u>

Proof:

$$\sum_{k=0}^{mn-1} C_{ik}\, C^{*}_{jk} = \sum_{v=0}^{n-1} \sum_{z=0}^{m-1} C_{um+w,\ vm+z}\ C^{*}_{u'm+w',\ vm+z}$$

with $k = vm+z$ $\quad v=0, \ldots, n-1$

$z=0, \ldots, m-1$

Using (2)

$$\sum_{k=0}^{mn-1} C_{ik}\, C^{*}_{jk} = \sum_{v=0}^{n-1} \sum_{z=0}^{m-1} A^{w}_{uv}\, A^{*\,w'}_{u'v}\ B^{v}_{wz}\, B^{*\,v}_{w'z}$$

$$= \sum_{v=0}^{n-1} A^{w}_{uv}\, A^{*\,w'}_{u'v} \underbrace{\sum_{z=0}^{m-1} B^{v}_{wz}\, B^{*v}_{w'z}}_{\delta_{w\,w'}}$$

$\delta_{w\,w'}$ by orthonormality of $\left[B^{v}\right]$

$$= \delta_{w,w'} \underbrace{\sum_{v=0}^{n-1} A^{w}_{uv}\ A^{*\,w}_{u'v}}_{\delta_{uu'}}$$

by orthonormality of $\left[A^{w}\right]$

$$= \delta_{ww'}\, \delta_{uu'} = \delta_{ij}$$

where δ_{ij} is the Kronecker delta $\delta_{ij} = 1$ if $i=j$

$= 0$ otherwise.

This proves that [C] is a unitary matrix. QED

In the particular case in which the matrices $\left[A^{i}\right] = \left[A\right]$ are all identical, and also the matrices $\left[B^{i}\right] = \left[B\right]$, then the generalized Kronecker

product $\{\mathcal{A}\} \otimes \{\mathcal{B}\}$ reduces to the usual Kronecker product of matrices [14]: $[A] \otimes [B]$.

b) Factorization of [C]: We now prove that

$$[C] = [P^t]\ [\mathrm{Diag}\{\mathcal{A}\}]\ [P]\ [\mathrm{Diag}\{\mathcal{B}\}] \qquad (3)$$

where $[\mathrm{Diag}\{\mathcal{A}\}]$ and $[\mathrm{Diag}\{\mathcal{B}\}]$ are block diagonal matrices formed with the matrices of the sets $\{\mathcal{A}\}$ and $\{\mathcal{B}\}$ (see Fig. 1) and [P] is the permutation matrix of order mn such that $P_{k\ell} = \delta_{vz'}\ \delta_{zv'}$ when $k = vn+z$, $\ell = v'm+z'$ and $z', v=0, \ldots, m-1; z, v'=0, \ldots, n-1$. Equation (3) is a generalization of the factorization of a simple Kronecker product into Good matrices [14].

Proof:

$$[\mathrm{Diag}\{\mathcal{A}\}]_{k'k} = \delta_{vv''}\ A^{v}_{z''z} \qquad \text{with} \quad k' = v''n+z''$$

$$[\mathrm{Diag}\{\mathcal{B}\}]_{\ell j} = \delta_{u'v'}\ B^{u'}_{z'w'} \qquad j = u'm + w'$$

$$[P^t]_{ik'} = \delta_{uz''}\ \delta_{wv''} \qquad i = um + w$$

We evaluate an element of the matrix on the right hand side of (3)

$$\sum_{k'=0}^{mn-1}\ \sum_{k=0}^{mn-1}\ \sum_{\ell=0}^{mn-1} [P^t]_{ik}\ [\mathrm{Diag}\{\mathcal{A}\}]_{k'k}\ [P]_{k\ell}[\mathrm{Diag}\{\mathcal{B}\}]_{\ell j} =$$

$$\sum_{v''=0}^{m-1}\ \sum_{z''=0}^{n-1}\ \sum_{v=0}^{m-1}\ \sum_{z=0}^{n-1}\ \sum_{v'=0}^{n-1}\ \sum_{z'=0}^{m-1} \delta_{uz''}\ \delta_{wv''}\ \delta_{vv''}\ A^{v}_{z''z}\ \delta_{vz'}\ \delta_{zv'}\ \delta_{u'v'}\ B^{u'}_{z'w'} =$$

$$A^{w}_{uu'}\ B^{u'}_{ww'} = C_{ij} \qquad \text{QED.}$$

c) Number of elementary operations:

With the computational blocks corresponding to the transforms $A^0, \ldots, A^{m-1}$ and $B^0, \ldots, B^{n-1}$, the factorization of equation (3) leads directly to the computational block of the transform C given in Figure 2.

From the structure of the algorithm of Figure 2 it is easy to see that if the matrices $[A^i]$ $(i=0, \ldots, m-1)$ and $[B^u]$ $(j=0,\ldots,n-1)$ have algorithms requiring respectively p_n^i and q_m^j elementary operations of a specific type, their generalized Kronecker product [C] will require P_{mn} of these operations with

$$P_{mn} = \sum_{i=0}^{m-1} p_n^i + \sum_{j=0}^{n-1} q_m^j \qquad (4)$$

In the particular case of a simple Kronecker product $p_n^i = p_n$ and $q_m^j = q_m$ so

$$P_{mn} = m\,P_n + n\,q_m \qquad (5)$$

Note that the use of rule 1 and 2 only increases the number of elementary operations while the order of the generated transform does not change. For rule 3, even if $[A^i]$ and $[B^j]$ do not have fast algorithms and thus require n^2 and m^2 elementary operations, [C] requires a maximum of $(m+n)mn \leq (mn)^2$ (for m, n > 1) elementary operations.

The results of equations (1), (4) and (5) are important: for every transform generated with the recursive rules presented, they give a simple and systematic way to estimate its computational complexity.

2. Identical Computation (IC) Family:

The generative rules defined above create a unified framework for the known fast unitary transforms, introduce new transforms, and allow assessment of the computational complexity of such transforms. In this paper, one large family of transforms is considered: the "identical computation transforms" that we discuss now.

We denote by $\{\mathcal{A}\} \otimes [B_q]$ the generalized Kronecker product of a set $\{\mathcal{A}\}$ of q matrices $[A_p^k]$ (K=0, ..., q-1) of order p and a set $\{\mathcal{B}\}$ of p identical matrices $[B_q]$ of order q. $[B_q]$ will be called a core matrix and $[A_p^k]$ a parent matrix. The IC transforms are recursively generated from a unique class, $\mathcal{C}$, of parent matrices of some order f and an original core matrix $[\mathcal{O}]$ of order q. An IC transform of order (qf^n) is then obtained from the original core matrix $[\mathcal{O}]$ by the recursive formulas:

$$[IC_{qf}] = [D_{qf}]\ [\{\mathcal{A}\} \otimes [O]]\ [D'_{qf}]$$

$$[IC_{qf^n}] = [D_{qf^n}]\ [\{\mathcal{A}_n\} \otimes [IC_{qf^{n-1}}]]\ [D'_{qf^n}] \qquad (6)$$

where the matrices [D] and [D'] express respectively a reordering followed by multiplications by roots of unity of the rows and the columns. All parent matrices of $\{\mathcal{A}_i\}$ .... $\{\mathcal{A}_n\}$ belong to $\mathcal{C}$.

The common characteristic of all the transforms of the IC family is that their algorithms only use in any computation intermediate results obtained from the input vector through identical computations (so the name of the family). This property provides a uniform treatment of successive components of the input vector if we consider that any parent matrix

treats uniformly its input vector. For this family, all the normalizations can be delayed to the last stage of computation.

We shall consider different choices for the original matrix $[\mathcal{O}]$, the class $\mathcal{C}$ of parent matrices, the matrices [D] and [D'] and the sets $\{\mathcal{A}_k\}$. We first show that the basic transforms, Fourier, Walsh-Hadamard and Haar, are IC transforms.

3. Basic Transforms: Fourier, Walsh-Hadamard, Haar:

In this section with the help of the generative rules, we examine the well known Fourier, W-H , and Haar transforms. This approach allows the derivation of some new results concerning the number of multiplications required by a FFT of composite order, a concise presentation of the different definitions of the W-H transform, and simple definitions of the Haar transform. In addition it makes apparent the common structure of these transforms. This will lead in the next section to the definition of families of transforms between the basic transforms.

In the following we emphasize specific orderings for the basic transforms: frequencies for the Fourier transform, zequencies[3] for the W-H transform and rank for the Haar transform. These orderings have proved to be useful in signal encoding because they concentrate the signal energy into the first transform coefficients, for some image models [21].

---

[3]This terminology has been introduced by Yuen [20]. The zequency is the number of zero crossings.

3.1 Generalized Fast Fourier Transform of Composite Order

a) decomposition theorem:

Given the Fourier matrices $[F_p]$ and $[F_q]$ of orders p and q respectively, the matrix $[F_{pq}]$ such that

$$[F_{pq}] = [\{[F_q^k]\} \otimes [F_p]]\ [P]^t \tag{7}$$

is the Fourier matrix of order pq. The set $\{[F_q^k]\}$, k=0, ... p-1 is such that

$$[F_q^k] = [F_q]\ [D_k] \text{ where} \tag{8}$$

$[D_k]$ is a diagonal matrix such that $(D_k)_{u'u'} = \exp(-2\pi j \frac{ku'}{pq})$

$[P]^t$ is the permutation matrix such that

$$P_{st} = \delta_{uz}\ \delta_{kw} \text{ with } s = uq + k \quad t = wp + z$$

$$u, z < p$$

$$k, w < q$$

Proof: We denote $[\{[F_q^k]\} \otimes [F_p]]$ by $[F'_{pq}]$.

$$(F'_{pq})_{ug+k,\ u'g+k'} = (F_p^k)_{uu'} \cdot (F_q)_{kk'}$$

$$= (F_p)_{uu'} \cdot e^{-2\pi j \frac{ku'}{pq}} \cdot (F_q)_{kk'} = \frac{1}{\sqrt{2^{pq}}} e^{-2\pi j (\frac{uu'}{p} + \frac{ku'}{pq} + \frac{kk'}{q})}$$

$$[F_{pq}] = [F'_{pq}] \cdot [P]^t \Rightarrow (F_{pq})_{ug+k,\ wp+z} = (F'_{pq})_{uq+k,\ u'q+k'} \cdot \delta_{zu'}\ \delta_{wk'}$$

$$= e^{-2\pi j\left(\frac{uz}{p} + \frac{kz}{pq} + \frac{kw}{q}\right)} = e^{-2\pi j \frac{(uq+k)\ (wp+z)}{pq}} \qquad \text{QED.}$$

Note that $[F_{pq}]$ is symmetric and orthonormal so that $([F_{pq}]^{-1})^* = [F_{pq}]$. Making use of (3) and (7) it is possible to derive a new expression for $[F_{pq}]$:

$$[F_{pq}] = [[F_p] \otimes \{[F_q'^k]\}]\ [P] \qquad (9)$$

$$\text{with } [F_2'^k] = [D_k]\ [F_q]$$

If $[F_p]$ and $[F_q]$ require respectively $\mathcal{A}_p$ and $\mathcal{A}_q$ complex additions, $\mathcal{M}_p$ and $\mathcal{M}_q$ complex multiplications, $[F_{pq}]$ will require by application of (4).

$$\mathcal{A}_{pq} = p\mathcal{A}_q + q\mathcal{A}_p \qquad \text{complex additions} \qquad (10)$$

$$\mathcal{M}_{pq} = p\mathcal{M}_q + q\mathcal{M}_p + C_{p,q} \qquad \text{complex multiplications} \qquad (11)$$

where $C_{p,q}$ is the number of complex multiplications introduced by (8).

$C_{p,q} = pq$ if all the factors including ± 1, $\pm j$ are considered.

$C_{p,q} = (p-1)\ (q-1)$ if the factors ± 1 are discarded

$C_{p,q} = (p-1)\ (q-1)-1$ if the factors $\pm j$ are also discarded (when (pq) is a power of 2).

b) Generalized FFT of composite order

If the order of the Fourier transform is composite, i.e. $N = \rho_1 \ldots \rho_n$, the previous decomposition theorem yields the well known FFT

algorithms [2] [3] detailed by Glassman [22] in the most general case. The recursive use of the formulae (10) and (11) gives the number of required operations. In the case of $N = r^n$ we can solve these recursive equations: this is the case of FFT of radix r.

$$\mathcal{A}_{r^n} = r^{n-1}\mathcal{A}_r + r\,\mathcal{A}_{r^{n-1}} \text{ or } \mathcal{A}_{r^n} = nr^{n-1}\mathcal{A}_r$$

$$\mathcal{M}_{r^n} = r^{n-1}\mathcal{M}_r + r\mathcal{M}_{r^{n-1}} + (r-\alpha)(r^{n-1}-\alpha) - \beta \text{ or} \tag{12}$$

$$\mathcal{M}_{r^n} = n\,r^{n-1}\mathcal{M}_r + (r-\alpha)\left[(n-1)\,r^{n-1} - \alpha\frac{(r^{n-1}-1)}{r-1}\right] - \beta\,\frac{(r^{n-1}-1)}{r-1} \tag{13}$$

(α, β depend on the value of $C_{p,q}$)

The radices 2, 4, 8 and 16 have been considered in the literature.

For the radix 2, which gives the most popular FFT, the recursive relations given by the decomposition theorem are

$$[F_{2^n}] = [\{[F_2^k]\} \otimes [F_{2^{n-1}}]]\,[P]^t \tag{14}$$

and $[F_{2^n}] = [[F_{2^{n-1}}] \otimes \{[F_2'^k]\}]\,[P]$

$$\text{with } [F_2^k] = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & e^{-2\pi j \frac{k}{2^n}} \\ 1 & -e^{-2\pi j \frac{k}{2^n}} \end{bmatrix} \text{ and } [F_2'^k] = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ e^{-2\pi j \frac{k}{2^n}} & -e^{-2\pi j \frac{k}{2^n}} \end{bmatrix} \tag{15}$$

The algorithm corresponding to the recursive formula (14) and obtained by recursive use of Fig. 2 is shown in Fig. 3a; it can be arranged equivalently with all operations "in place" as shown in Fig. 3b

which is the classical diagram of the Cooley-Tuckey [2] algorithm with decimation in time.

The algorithm corresponding to the formula (15) is the Sande-Tukey [3] algorithm with decimation in frequency and is shown in Fig. 3-c and d.

For these Figures the factors are

$$a_o = 1$$

$$a_1 = \exp(-2\pi j/8)$$

$$a_2 = \exp(-4\pi j/8) = -j$$

$$a_3 = \exp(-6\pi j/8)$$

We can compare the FFT with radices 2, 4, 8 and 16 for transforms of order $N = 2^n = r^{\frac{n}{\log_2 r}}$ (n is then a multiple of 12). The formulas (12) and (13) then give:

Radix	$\mathcal{A}_r$	$\mathcal{M}_r$	$\mathcal{A}_{r^n}$	$\mathcal{M}^1_{r^n}$ (all factors)	$\mathcal{M}^2_{r^n}$ (no factors $\pm$ 1)	$\mathcal{M}^3_{r^n}$ (no factors $\pm$ 1 $\pm$ j)
2	2	0	$n2^n$	$(n-1)\ 2^n$	$n2^{n-1} - 2^n + 1$	$n2^{n-1} - 3.2^{n-1} + 2$
4	8	0		$(\frac{n}{2}-1)\ 2^n$	$3n2^{n-3} - 2^n + 1$	$3n2^{n-3} - \frac{13.2^{n-2}-4}{3}$
8	24	1		$(\frac{3n}{8}-1)\ 2^n$	$\frac{n2^n}{3} - 2^n + 1$	$\frac{n2^n}{3} - \frac{57.2^{n-3} - 8}{7}$
16	64	6		$(\frac{11n}{32}-1)\ 2^n$	$\frac{21n2^n}{64} - 2^n + 1$	$\frac{21n2^n}{64} - \frac{241.2^{n-4}-16}{15}$

The column $\mathcal{M}^2_{r^n}$ has been given by Singleton [23]. In fact our approach allows the evaluation of the computational complexity for any composite order, in particular for mixed radix FFT.

The factors ± 1 are easy to track in the algorithms and for most realizations multiplications by ± 1 are not performed. The factors $\pm j$ appear in various places in the algorithms and in most realizations multiplications by $\pm j$ are performed; however, in an error analysis these multiplications do not introduce any rounding error and the column $\mathcal{M}^3_{r^n}$ is then of interest.

3.2 Walsh-Hadamard Transform:

The W-H functions are well known and the results presented in this section are explicit or implicit in many publications. Here, we wish to express these results in terms of our generative rules; we think that the following compact notation clarifies the relations between the various orderings of the W-H functions and the different algorithms. This approach also makes apparent the common structure of the W-H transform with the Fourier and Haar transforms.

Three distinct orderings or rows of the W-H transform are commonly used and are of interest (see the discussion by Yuen [20]). For each of these orderings there exists a recursive matrix definition:

a) "natural order" It is obtained by simple Kronecker product without any permutation

$$[WH_{2^n}\ \text{nat.}] = [F_2] \otimes [WH_{2^{n-1}}\ \text{nat.}] \tag{16}$$

with the original core matrix $[F_2]$. This relation gives directly, by recursive use of Fig. 2, the fast algorithm of Fig. 4a (without the reorderings). A different presentation of this algorithm with identical stages of computation is given in Fig. 4b.

b) Paley's ordering: Used originally by Paley [24] it seems more suitable for mathematical developments than the other orderings. The recursive relations introduced by Yuen [20] are expressed by the matrix relation

$$[WH_{2^n}\ \text{pal.}] = [[F_2] \otimes [WH_{2^{n-1}}\ \text{pal.}]]\ [P]^t \qquad (17)$$

with the original core matrix $[F_2]$ and $[P]$ as defined previously for the Fourier transform (see section 3.1). This relation gives the algorithm of Fig. 4c (without the bit-inversion reordering) by recursive use of Fig. 2. A different presentation of this algorithm can be found in [6].

c) zequency ordering: This is the original ordering by Walsh [25] and is the ordering of interest for signal encoding because it ranks the transform coefficients roughly according to their variances for signal statistics commonly encountered in practice. The generating process $\mathcal{W}$, introduced in [6], defines recursively the W-H matrices in zequency order. We can express it by the matrix relation

$$[WH_{2^n}\ \text{zeq.}] = [W]\ [[F_2] \otimes [WH_{2^{n-1}}\ \text{zeq.}]] \qquad (18)$$

where [W] denotes the reordering of the process $\mathcal{W}$. A zequency ordered algorithm has been investigated by Manz [26].

Although the zequency ordering could be generated recursively, the corresponding algorithm would not be simple and it may be preferable to obtain the W-H transform in natural or Paley's order and then perform a global reordering.

Given the transform coefficients in Paley's order a "bit inversion" reordering, denoted in matrix form by [Q], is necessary to put them in zequency order: in a bit-inversion permutation, consecutive coefficients with k^{th} bit of the binary representation of their indexes equal to 1 are put in reverse order. This operation is performed for all bits starting from the rightmost bit. With 8 coefficients to reorder this means the following permutation:

index of coefficients in Paley's order	binary representation	reverse order for middle bit	reverse order for leftmost bit	final order (see Fig. 4c)
0	000	000	000	0
1	001	001	001	1
2	010	011	011	3
3	011	010	010	2
4	100	100	110	6
5	101	101	111	7
6	110	111	101	5
7	111	110	100	4

Given the transform coefficients in natural order, a bit reversal ordering, denoted by the matrix [R], will order them in Paley's order and a bit-inversion will order them in zequency order. For 8 coefficients to reorder we have:

index of coefficients in natural order	binary representation	bit reversal		bit inversion	final ordering (compare with Fig.4a,b)
0	000	000	0		0
1	001	100	4		4
2	010	010	2		6
3	011	110	6		2
4	100	001	1		3
5	101	101	5		7
6	110	011	3		5
7	111	111	7		1

It is important to note that, as the W-H matrices are symmetric in any of the three orderings, these reorderings can be performed on the columns as well as on the rows. Hence we have the matrix relations:

$$[WH_{2^n}\ \text{zeq.}] = [Q]\ [WH_{2^n}\ \text{pal.}] = [WH_{2^n}\ \text{pal.}]\ [Q]^t$$

$$[WH_{2^n}\ \text{zeq.}] = [Q]\ [R]\ [WH_{2^n}\ \text{nat.}] = [WH_{2^n}\ \text{nat.}]\ [R]^t\ [Q]^t$$

Since the W-H matrices are their own inverses, we have also using (16) and (17) the following recursive relations

$$[WH_{2^n}\ \text{nat.}] = [P]^t\ [[WH_{2^{n-1}}\ \text{nat.}] \otimes [F_2]]\ [P]$$

$$[WH_{2^n}\ \text{pal.}] = [[WH_{2^{n-1}}\ \text{pal.}] \otimes [F_2]]\ [P]$$

These relations however do not give different algorithms. All these

algorithms differ only by reordering and so have the same number of additions given by

$$\mathcal{A}_{2^n} = 2 \cdot \mathcal{A}_{2^{n-1}} + 2,\ 2^{n-1} \text{ with } \mathcal{A}_2 = 2 \text{ which gives}$$

$$\mathcal{A}_{2^n} = n2^n, \text{ a well known result.}$$

3.3 Haar transform:

The Haar transform is usually defined from the Haar functions [11]. The Haar matrix of order 8 $[H_8]$ ordered by ranks is as follows

$$[H_8] = \frac{1}{\sqrt{8}} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} \\ 2 & -2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & -2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & -2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & -2 \end{bmatrix} \quad \begin{matrix} \text{Zones} \\ 0 \\ 1 \\ \\ 2 \\ \\ \\ \\ 3 \end{matrix}$$

Here we use the generative rules to define recursively the Haar matrices and we have found two definitions:

1) The Haar matrix of order 2^n is obtained from the Haar matrix of order 2^{n-1} by simple Kronecker product with $[I_2] = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ followed by rotation of the rows 0 and 2^n by $[I_2]$. This is the process $\mathcal{H}$ of [6],

in terms of generative rules.

2) The Haar matrices are recursively defined by the relation:

$$[H_{2^n} \text{ nat.}]\ \{[F_2], [I_2], \ldots., [I_2]\} \circledast [H_{2^{n-1}} \text{ nat.}] \qquad (19)$$

The rows are obtained in "natural" order. To reorder them by their ranks, we need a "zonal bit reversal" ordering. A zone as defined in [6] is a set of coefficients with indexes between two successive powers of 2. A "zonal bit reversal" ordering is a bit-reversal followed by a reordering in the original order inside each zone. For 8 coefficients the zonal bit reversal ordering gives:

Index	Binary representation	Bit reversal	Reordering inside zones	final order (see Fig. 5a)
0	000	000	000	0
1	001	100	100	4
2	010	010	010	2
3	011	110	110	3
4	100	001	001	1
5	101	101	011	6
6	110	011	101	5
7	111	111	111	7

With both definitions we obtain by recursive application of the diagram of Fig. 2 the algorithm of Fig. 5a. This algorithm can be more conveniently organized as shown in Fig. 5b and give the rows directly ordered by their rank.

By application of (4) we obtain the following recursive formula for the number of additions:

$\mathcal{A}_{2^n} = 2 \cdot \mathcal{A}_{2^{n-1}} + 2$. Hence $\mathcal{A}_{2^n} = 2(2^n-1)$ with $\mathcal{A}_2 = 2$. 2^{n-1} normalizations are also required. A modified Haar transform obtained from the Haar transform by permutation of its columns is related to the Fourier transform (see later section 7.3): it is defined recursively by:

$$[MH_{2^n}] = [Z]\ [\{[F_2], [I_2], \ldots, [I_2]\} \otimes [MH_{2^{n-1}}]]\ [P]^t \qquad (20)$$

Globally the permutations [Z] perform a bit-reversal ordering inside each zone.

The modified Haar matrix of order 8 is as follows

$$[MH_8] = \frac{1}{\sqrt{8}} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ \sqrt{2} & 0 & -\sqrt{2} & 0 & +\sqrt{2} & 0 & -\sqrt{2} & 0 \\ 0 & \sqrt{2} & 0 & -\sqrt{2} & 0 & +\sqrt{2} & 0 & -\sqrt{2} \\ 2 & 0 & 0 & 0 & -2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & -2 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 & -2 & 0 \end{bmatrix}$$

and its algorithm is given in Fig. 5c.

4. Generalizations of the basic transforms

For the three basic transforms we have found recursive definitions with a matrix formula similar to (6). The direct comparison of these definitions suggests the generalization of the basic transforms to families between them. The simplest generalizations are two families between the

Fourier and W-H transforms, and between the W-H and Haar transforms. A larger generalization is the IC_2 family which includes all three basic transforms with parent matrices of order 2. A number of these generalized transforms has recently been discussed independently. We would like to show that they fall easily within the framework we have developed and that further generalizations are clearly possible.

4.1. Family between W-H and F:

If we compare the recursive generation of the Fourier matrices with radix 2 (14) and the W-H transform (17), we notice that they differ only by a set of factors. If we exclude the reordering of the rows, we see that we can easily generalize the W-H and Fourier transforms to a large family of unitary transforms given by the recursive formula

$$[GT_{2^n}] = \{[F_2(\theta_0),..,F_2(\theta_{2^{n-1}-1})]\} \otimes [GT_{2^{n-1}}]\ [P]^t$$

where $[F_2(\theta)] = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & \exp(-j\theta) \\ 1 & -\exp(-j\theta) \end{bmatrix}$

This family includes the W-H and Fourier transforms for the appropriate choices of the parameters $\theta_0 \ldots \theta_{2^{n-1}-1}$.

Two families have appeared in the literature for special choices of these parameters:

1) $\theta_k = \frac{2\pi\ kc}{2^n}$ where c is a real scalar varying from 0 (W-H transform) to 1 (Fourier transform). The corresponding transform has been called "general spectral analyzer" [14] [15].

2) $\theta_k = \frac{2\pi k}{2^n}$ if k mod $(2^{n-1-g}) = 0$

where g is an integer varying from 0 (W-H transform) to n-1 (Fourier transform), and $\theta_k = 0$ otherwise. The corresponding transform is the "Generalized Discrete Transform" [17]; we will denote it $[GT^g_{2^n}]$. Fig. 6 shows the matrix $[GT^2_{16}]$ and its fast algorithm. Many other choices for these factors are obviously possible and these two special choices do not seem to bear any exceptional importance.

As an example of use of our general formulas we compute now the required number of multiplications for the Generalized Discrete Transform. There are 2^g-1 factors different from 1.

$$\text{So } \mathcal{M}^g_{2^n} = 2\mathcal{M}^g_{2^{n-1}} + 2^g - 1 \qquad \text{for } n - 1 > g$$

$$\mathcal{M}^g_{2^n} = 2\mathcal{M}^g_{2^{n-1}} + 2^{n-1} - 1 \qquad \text{otherwise.}$$

$$\text{So that } \mathcal{M}^g_{2^n} = g\ 2^{n-1} - 2^g + 1$$

If we do not count the multiplications by $\pm j$, we find similarly

$$\mathcal{M}^g_{2^n} = (g-1)\ 2^{n-1} - 2^g + 2$$

4.2 Family between Haar and W-H

In [6] one of the authors has presented a family of transforms between the Haar and W-H transforms. This family was obtained by replacing W-H transforms of lower order by Haar transforms in the decomposition of the fast algorithm of a W-H transform. Now we have further decomposed the fast

algorithms of the two transforms up to similar recursive formulas (16) and (19) or (17) and (20). Obviously, if we choose any of the 2^{n-1} parent matrices needed to generate the matrix of order 2^n to be either $[F_2]$ or $[I_2]$, we obtain a large family of unitary matrices which includes the Haar, W-H, and unity matrices. There are

$$2^{2^{n-1}} \cdot 2^{2^{n-2}} \cdot \ldots \cdot 2^1 = 2^{1+ \ldots + 2^{n-1}} = 2^{2^n-1}$$

members of order 2^n in this family.

The number of additions is obviously twice the number of parent matrices equal to $[F_2]$. For the W-H transform we have $n2^{n-1}$ such matrices and therefore $n2^n$ additions. For the Haar transform we have $2^{n-1} + 2^{n-2} \ldots + 1 = 2^n-1$ such matrices $[F_2]$ and so $2(2^n-1)$ additions. The number of normalizations varies from 0(W-H) to 2^n-2 (the normalizing factors come by pairs and all pairs are different in the worst case). No multiplication is required during the computation. At the order 8, 2^7 matrices are in the family. We show in Figure 7 one of them with its fast algorithm.

Assume as a particular case that we choose the parent matrices of the recursive formulas to be

$$[F^k] \quad k = 0, \ldots, 2^{p-1}-1$$

with

$$[F^k] = [F_2] \text{ for } k = 0 \bmod (2^{p+h-n})$$

$$F^k = [I_2] \quad \text{otherwise}$$

where p is the stage of computation up to n when we generate a transform of

order 2^n and h an index lower than n.

Then, if the recursive formula used is similar to (16) and (19) we obtain a subclass of n transforms: for h = 0 we have the W-H transform and for h = n - 1 we have the Haar transform, both in natural order.

If the recursive formula is similar to (17) and (20) (with the permutation $[P]^t$ of the columns) we still obtain n transforms: for h = 0 we have the W-H transform in Paley's order and for h = n - 1 we have an unordered modified Haar transform. We denote these transforms $[WHH_2^k n]$. Fig. 8 shows $[WHH_{16}^2]$ and its fast algorithm.

4.3 IC_2 family:

To generate the family between W-H and W we have introduced a set of factors into the recursive formula for the W-H transform. To generate the family between W-H and Haar, we have replaced some parent matrices by the identity matrix $[I_2]$ in the same recursive definition of the W-H transform. If we allow simultaneously both operations we generate a larger family that we call IC_2.

More formally if $[T_{2^{n-1}}]$ is a member of IC_2 of order 2^{n-1}, a member of order 2^n is given by

$$[T_{2^n}] = [D_1]\ [\{[C_0], \ldots, [C_{2^{n-1}-1}]\} \otimes [T_{2^{n-1}}]]\ [D_2]$$

where $[D_1]$ and $[D_2]$ are permutation matrices and $C_0, \ldots, C_{2^{n-1}-1}$ are either

$$[I_2] \text{ or } [F_2(\theta)] = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & \exp(-h\theta) \\ 1 & -\exp(-j\theta) \end{bmatrix}$$

Let us call this class of parent matrices $\mathcal{C}_2$. For the order 2^n, $2^n - 1 (2^{n-1} + 2^{n-2} + \ldots + 1 = 2^n - 1)$ parent matrices have to be chosen independently in $\mathcal{C}_2$: we say that the family IC_2 has $2^n - 1$ degrees of freedom over $\mathcal{C}_2$ (see footnote 4).

The IC_2 family is very large and includes the families between W-H and F, W-H and Haar.

The number of required operations is given recursively by

additions: $\mathcal{A}_{2^n} = 2\ \mathcal{A}_{2^{n-1}} + 2A_n$

$$\text{Hence } \mathcal{A}_{2^n} = \sum_{k=1}^{n} 2^{n-k+1}\ A_k \tag{21}$$

multiplications: $\mathcal{M}_{2^n} = 2\,\mathcal{M}_{2^{n-1}} + L_n$

$$\text{hence } \mathcal{M}_{2^n} = \sum_{k=2}^{n} 2^{n-k}\ L_k \tag{22}$$

---

(4) This notion of degree of freedom is an extension of a concept introduced by Andrews and Caspari [16]. For them the degree of freedom of a class of matrices is the number of free parameters required to define this class. This definition is ambiguous when the constraints which define a class cannot be reduced to a set of free parameters. For example, the unitary matrices of order 2 are given 1 degree of freedom in [16] when in fact, on the real numbers, the most general matrix is

$$\begin{bmatrix} \cos\alpha & \sin\alpha \\ \varepsilon \sin\alpha & -\varepsilon \cos\alpha \end{bmatrix} \qquad \varepsilon = \pm 1 \qquad \alpha \in [0, 2\pi]$$

and on the complex numbers the general solution depends on 4 angles $\in [0, 2\pi]$ and 2 binary choices. Our approach is to track as far as possible the reduction to independent choices. If it can be reduced to a number of free parameters our degree of freedom will be the number of these parameters. Note that the relations (1) and (4) apply also to the recursive computation of the degree of freedom of a class. Note also that the degree of freedom has generally no relation with the computational complexity (which varies usually for the transforms of a class).

where A_k is the number of parent matrices different from $[I_2]$ at this k^{th} stage, and L_k the number of factors different from ± 1 (and maybe $\pm j$) at this stage.

For Haar $\quad A_k = 1$ for any k and $\mathcal{A}_{2^n} = 2(2^n-1)$

For W-H $\quad A_k = 2^{k-1}$ and $\mathcal{A}_{2^n} = n2^n$

For Fourier $\quad \mathcal{A}_{2^n} = n2^n$ and $L_k = 2^{k-1}$, $2^{k-1}-1$ or $2^{k-2}-2$, which yield the results of section 2.3 (radix 2).

We present now an example of interest in the IC_2 family: a class of transforms which make a discrete transition between the 3 basic transforms and which we call therefore the WFH class.

Each transform of this class is indexed by two positive integer parameters h and g such that $h + g < n$ when 2^n is the order of the transform and is denoted $[WFH^{g,h}_{2^n}]$.

$[WFH_{2^n}^{g,k}]$ is obtained recursively as the Fourier transform of radix 2 (formula 14) but with the parent matrices $[P_2^k]$ $k = 0, .., 2^{p-1}$ such that

$$[P_2^k] = [F_2(2\pi k/2^p)] \text{ for } k = 0 \pmod{2^{p-g-1}}$$

$$[P_2^k] = [F_2] \quad \text{for } k = 0 \pmod{2^{p+h-n}} \text{ but}$$

$$k \neq 0 \pmod{2^{p-g-1}}$$

$$[P_2^k] = [I_2] \text{ otherwise}$$

where p is the level of computation up to n.

We can represent then WFH transforms on a (g, h) plane as shown in Fig. 9. With appropriate permutation matrices, for h = 0 we have the n Generalized Discrete Transforms (see section 4.1), for g = 0 the n WHH transforms (see section 4.2). $WFH^{0,0}$ is the W-H transform, $WHH^{0,\,n-1}$ the Modified Haar transform and $WHH^{n-1,0}$ the Fourier transform. For h + g = n - 1 we have a set of n transforms in between the Fourier and Haar transforms which have been called the Modified Generalized Discrete Transforms and defined after much work in [18].

5. Other IC transforms:

Except for the Fourier transform, we have restricted ourselves so far to IC transforms obtained from original core matrix $[F_2]$ and parent matrices of order 2. The generative rules have given a unified approach of the usual unitary transforms. We now consider some examples with a different original core matrix and parent matrices of higher orders.

The matrices of order 2 are of practical interest for the fast algorithm as long as we perform the required operations (specially additions) with only two operands at a time. If fast additions involving, let us say, f operands, become available, the transforms with parent matrices of order f may be of interest.

Most of the recursive structures of the transforms presented in the previcus sections can be applied to parent matrices of higher orders than 2. We now give some examples:

a) different original core matrix

In the definition of the W-H transform the original core matrix $[F_2]$ may be replaced by the core matrix $\begin{bmatrix} \cos\theta & \sin\theta \\ \sin\theta & -\cos\theta \end{bmatrix}$ and we obtain a

transform considered by Andrews et al. [15]. This original core matrix can be used for all the recursive definitions considered.

b) Generalized 2 and 3 valued transforms:

In the definition of the W-H transform the role of $[F_2]$ as original core matrix and parent matrix can be performed by any unitary matrix [U] of order f. If [U] is an Hadamard matrix (its entries are $\pm 1/\sqrt{f}$) the generated matrix of order f^n will also be an Hadamard matrix. These matrices have been called "generalized 2-valued transforms" [19]. Similarly we can replace $[F_2]$ in the definition of the Haar transform by the same matrix [U] and we will generate a unitary matrix with entries 0 or $\pm 1/C_i$ where C_i is the normalizing factor of the i^{th} row: these matrices are the "generalized 3-valued transforms" [19]. More generally [U] can replace $[F_2]$ in the definition of the family of transforms between W-H and Haar.

c) IC_f family:

The IC_2 family was based on the set $\mathcal{C}_2$ for the parent matrices. We can define similarly the IC_f family based on the class $\mathcal{C}_f$ of parent matrices of order f which contains $[I_f]$ and $[F_f(\theta_1, \ldots, \theta_{f-1})]$ where k^{th} column of $[F_f(\theta_1, \ldots, \theta_{f-1})]$ = k^{th} column of $[F_f]$ x $\exp(e^{-j\theta k})$ ($[F_f]$ is the Fourier matrix of order f). The family IC_f has $\frac{f^n-1}{f-1}$ independent parent matrices chosen in $\mathcal{C}_f$: we say that IC_f has $\frac{f^n-1}{f-1}$ degrees of freedom over $\mathcal{C}_f$. The required number of additions and multiplication is computed recursively as done for (21) and (22):

$$\text{additions: } \mathcal{A}_{f^n} = f \cdot \mathcal{A}_{f^{n-1}} + \mathcal{A}_f A_n$$

$$\Rightarrow \mathcal{A}_{f^n} = \mathcal{A}_f \sum_{k=1}^{n} f^{n-k} A_k \qquad (23)$$

multiplications: $\mathcal{M}_{f^n} = f\,\mathcal{M}_{f^{n-1}} + \mathcal{M}_f\, A_n + L_n$

$$\mathcal{M}_{f^n} = \mathcal{M}_f \sum_{k=1}^{n} f^{n-k} A_k + \sum_{k-2}^{n} f^{n-k} L_k \quad (24)$$

where A_k is the number of parent matrices different from $[I_f]$ at the k^{th} stage and L_k the number of column multiplications with factors different from ± 1 (and *maybe* $\pm j$) at this k^{th} *stage*.

d) $WFH^{g,h}_{f^n}$ subfamily of IC_f

By analogy to the $WFH^{g,h}_{2^n}$ subfamily of IC_2 we can define the subfamily $[WFH^{g,h}_{f^n}]$ of IC_f as follows:

$[WFH^{g,h}_{f^n}]$ is obtained by successive generalized Kronecker products with the sets $\{[M^0], \ldots [M^k] \ldots, [M^{f^{p-1}-1}]\}$ of parent matrices such that

$[M^k] = [F_f^k]$ with column i of $[F_f^k]$ = column i of $[F_f]$ x $e^{\frac{-2\pi jki}{f^f}}$

for $k = 0 \bmod (f^{p-g-1})$

$[M^k] = [F_f]$ for $k = 0 \bmod (f^{p+h-n})$ and

$k \neq 0 \bmod (f^{p-g-1})$

$[M^k] = [I_f]$ otherwise

and at each level the permutation matrix $[P]^t$, $P_{st} = \delta_{uz}\,\delta_{kw}$ with $s = uf + k$, $t = wf^{p-1} + z$, is applied to reorder the columns.

It is easy to see that $[WFH^{n-1,\,0}_{f^n}]$ is the usual Fourier transform of order f^n; the matrices $[WFH^{0,0}_{f^n}]$ and $[WFH^{0,n-1}_{f^n}]$ have been introduced in the

literature respectively by Chrestenson [27] and Watari [28]. For these 2 matrices (23) and (24) reduce to the same recursive formulas and denoting by $\mathcal{P}$ the number of additions or multiplications:

for $WFH_{f^n}^{0,0}$ $\qquad \mathcal{P}_{f^n} = n \; \mathcal{P}_f \, f^{n-1}$

for $WFH_{f^n}^{0,n-1}$ $\qquad \mathcal{P}_{f^n} = \mathcal{P}_f \frac{f^n-1}{f-1}$

(This last result corrects the result given in [15], page 20.)
The other matrices of the family can be represented in the g-h diagram of Fig. 9.

6. Slant transform:

The Slant transform has been proposed by Enomoto et al. [29] for the order 8. Pratt et al. [30] have generalized this transform to any order 2^n and compared its performance with other transforms [31]. In this section we want to express the recursive generation of the Slant transform with our generative rules and compute the number of elementary operations required by its fast algorithm.

The Slant transforms of orders 4 and 8, $[S_4]$ and $[S_8]$, are as follows (in "natural" order).

$$[S_4] = \frac{1}{\sqrt{4}} \begin{bmatrix} 1 & 1 & 1 & 1 & \\ 1 & -3 & 3 & -1 & \times 1/\sqrt{5} \\ 3 & 1 & -1 & -3 & \times 1/\sqrt{5} \\ 1 & -1 & -1 & 1 & \end{bmatrix} \qquad \begin{matrix} \underline{\text{Zequencies}} \\ 0 \\ 3 \\ 1 \\ 2 \end{matrix}$$

$$[S_8] = \frac{1}{\sqrt{8}} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & & \text{Zequencies} \\ 1 & -3 & 3 & -1 & 1 & -3 & 3 & -1 & \times 1/\sqrt{5} & 7 \\ 7 & -1 & -9 & -17 & 17 & 9 & 1 & -7 & \times 1/\sqrt{5\times21} & 3 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & & 4 \\ 7 & 5 & 3 & 1 & -1 & -3 & -5 & -7 & \times 1/\sqrt{21} & 1 \\ 1 & -3 & 3 & -1 & -1 & 3 & -3 & 1 & \times 1/\sqrt{5} & 6 \\ 3 & 1 & -1 & -3 & -3 & -1 & 1 & 3 & \times 1/\sqrt{5} & 2 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 & & 5 \end{bmatrix}$$

The rows can be reordered by zequencies with the same permutation as the W-H transform in natural order.

The Slant transform of order 2^n in natural order is obtained from the Slant transform of order 2^{n-1} in natural order by simple Kronecker product with $[F_2]$ followed by rotation of the rows 2^{n-2} and 2^{n-1} by the matrix

$$\begin{bmatrix} \text{Sin } \theta_n & \text{Cos } \theta_n \\ \text{Cos } \theta_n & -\text{Sin } \theta_n \end{bmatrix}$$

with $\text{Sin } \theta_n = \sqrt{\dfrac{2^{2n-2}-1}{2^{2n}-1}}$

and $\text{Cos } \theta_n = \dfrac{2^{n-1}}{\sqrt{\dfrac{2^{2n}-1}{3}}}$

This choice of θ_n introduces in the Slant matrix $[S_{2^n}]$ the Slant vector $\vec{S}$

with components linearly decreasing:

$$S_i = \frac{(2^n - 1) - 2i}{\sqrt{\frac{2^n(2^{2n} - 1)}{3}}}$$

But some normalizations can be delayed to the last stage of computation and the rows 2^{n-2} and 2^{n-1} are rotated by the matrix

$$\begin{bmatrix} 2^{n-1} & -\frac{(2^{2n-2}-1)}{3} \\ 1 & 2^{n-1} \end{bmatrix}$$

requiring 2 shifts, 2 additions, 1 multiplication. The corresponding algorithm is shown in Fig. 10a.

Number of elementary operations:

Formulas (1) and (5) give:

for the number of additions:

$$\mathcal{A}_{2^n} = 2\,\mathcal{A}_{2^{n-1}} + 2^{n-1} \cdot 2 + 2 \text{ with } \mathcal{A}_2 = 2$$

$$\text{hence } \mathcal{A}_{2^n} = (n+1)\,2^n - 2$$

for the number of shifts:

$$\mathcal{S}_{2^n} = 2 \cdot \mathcal{S}_{2^{n-1}} + 2 \quad \text{with } \mathcal{S}_2 = 0$$

$$\text{hence } \mathcal{S}_{2^n} = 2^n - 2$$

for the number of multiplications:

$$\mathcal{M}_{2^n} = 2\mathcal{M}_{2^{n-1}} + 1 \quad \text{with } \mathcal{M}_4 = 0$$

$$\text{hence } \mathcal{M}_{2^n} = 2^{n-2}-1$$

Finally $2^n - 2^{n-2}-1$ normalizations are required at the last stage of computation.

However the algorithm at the order 4 can be performed with 8 additions, 2 multiplications as shown in Fig. 10b [30] instead of 10 additions and 2 shifts as above. The formulas (1) and (5) give then:

additions:

$$\mathcal{A}'_{2^n} = 2\,\mathcal{A}'_{2^{n-1}} + 2^{n-1} \cdot 2 + 2 \qquad \text{with } \mathcal{A}'_4 = 8$$

$$\text{hence } \mathcal{A}'_{2^n} = (2n+1)\,2^{n-1}-2 = \mathcal{A}_{2^n}-2^{n-1}$$

shifts:

$$\mathcal{S}'_{2^n} = 2 \cdot \mathcal{S}'_{2^{n-1}} + 2 \text{ with } \mathcal{S}'_4 = 0$$

$$\text{hence } \mathcal{S}'_{2^n} = 2^{n-1}-2 = \mathcal{S}_{2^n} - 2^{n-1}$$

multiplications:

$$\mathcal{M}'_{2^n} = 2\mathcal{M}'_{2^{n-1}} + 1 \text{ with } \mathcal{M}'_4 = 2$$

$$\text{hence } \mathcal{M}'_{2^n} = 32^{n-2}-1 = \mathcal{M}_{2^n} + 2^{n-1}$$

and as before $2^n - 2^{n-2} - 1$ normalizations.

7. Additional properties and generalizations of unitary transforms:

In this section we discuss briefly the complex extension of a real transform. We also point out some additional relations between transforms suggested by the unified framework presented.

7.1 Complex extension of a real transform:

From a real unitary matrix [RT] with rows $RT_0, \ldots, RT_{N-1}$, we construct a complex extension noted [CT] with rows $CT_0, \ldots, CT_N$ by creating two complex rows CT_p and CT_q from two real rows RT_m RT_n as follows

$$CT_p = \frac{1}{\sqrt{2}} (RT_m - j\ RT_n)$$

$$CT_q = \frac{1}{\sqrt{2}} (RT_m + j\ RT_n) \qquad (25)$$

Then the complex transform $\mathcal{V} = \mathcal{R} + j\mathcal{I}$ of a complex input vector V = R + j I is expressed uniquely from the real transforms of $\mathcal{R}$ and $\mathcal{I}$ denoted $\widetilde{\mathcal{R}}$ and $\widetilde{\mathcal{I}}$:

$$\mathcal{V}_p = CT_p \cdot V = \frac{1}{\sqrt{2}}(RT_m - RT_n)\ (R + j\ I) \text{ or}$$

$$\mathcal{V}_p = \frac{1}{\sqrt{2}}(\widetilde{\mathcal{R}}_m + \widetilde{\mathcal{I}}_n) + j(\widetilde{\mathcal{I}}_m - \widetilde{\mathcal{R}}_n) \qquad \text{and similarly}$$

$$\mathcal{V}_q = \frac{1}{\sqrt{2}}(\widetilde{\mathcal{R}}_m - \widetilde{\mathcal{I}}_n) + j(\widetilde{\mathcal{I}}_m + \widetilde{\mathcal{R}}_n)$$

With these relations the properties of complex transforms can be deduced from those of the real transform. In the literature, besides the real and complex Fourier transforms, the complex W-H transfoms (also called Complex BIFORE transform) [32][33] complex Haar transform (also called Complex Modified BIFORE transform) [34] have been defined.

Note that the complex W-H transform obtained by relations (25) would have entries $\pm 1 \pm j$; commonly the rows are then rotated by $\frac{1+j}{\sqrt{2}}$ to give a transform with entries ± 1 and $\pm j$. The rows of the complex W-H transform can be ordered according to a generalized frequency defined as the number of clockwise rotations around the origin when following cyclically the entries of a row.

7.2 Multidimensional transforms

The techniques presented for the one dimensional transforms extend to multidimensional separable transforms. Let us denote an input array of p dimensions by $A_{i_q, \ldots, i_p}$ and the p-dimensional separable transform by $T_{u_1, \ldots, u_p, i_1, \ldots, i_p} = T^1_{u_1 i_1} T^2_{u_2 i_2} \cdots T^p_{u_p i_p}$. Then the transformed array

$$B_{u_1, \ldots, u_p} = \sum_{i_1} \sum_{i_2} \cdots \sum_{i_p} A_{i_1, \ldots, i_p} T_{u_1, \ldots, u_p, i_1, \ldots, i_p}$$

can be written

$$B_{u_1, \ldots, u_p} = \sum_{i_p} T^p_{u_p i_p} \sum_{i_{p-1}} \cdots \sum_{i_1} A_{i_1, \ldots i_p} T^1_{u_1 i_1}.$$

If we express both arrays as 1 dimensional vectors A and B , for which indexes are obtained by lexicographic ordering of the indexes $(i_1, \ldots, i_p)$ and $(u_1, \ldots, u_p)$, the multidimensional transform can be expressed as a 1-dimensional transform:

$$A = [[T^1] \otimes [T^2] \ldots \otimes [T^p]]\ B$$

$$A = [T]\ B$$

The multidimensional transform has been reduced to a 1 dimensional transform. This expression now allows the evalution of the number of elementary operations and other generalizations discussed previously.

7.3 Relations between transforms

Two transforms with similar structures will often be related by matrix relations or energy invariants between the two sets of transformed coefficients.

a) matrix relations between transforms of same order:

In [6], matrix relations between the W-H and Haar transforms were proved. More generally, for WFH families, similar relations hold for all transforms lying on the same vertical line in the g-h graph of Fig. 9. These transforms only differ by the number of parent matrices $[F_2]$ they include. Therefore a multiplication by all the missing $[F_2]$ matrices will generate one transform from the other. Note that these relations only involve computations in zones as defined in 3.3 or subzones (zonal divisions of a zone).

b) energy invariants:

By Parseval's theorem the total energy of the transform coefficients of a same vector with different transforms is preserved. However, it may happen that the energy of a subset of coefficients is the same for some transforms: we say then there is an energy invariant between these transforms. Energy invariants are most likely when the transforms have an identical structure with different multiplicators. For example, by direct comparison of the

algorithms for the Fourier (Fig. 3b) W-H (Fig. 4c) and modified Haar (Fig. 5c), it is clear that the transformed coefficients before respective reorderings have identical energies in the zones defined in 3.3. This leads to the following energy invariants for the order 8.

Zone	Fourier (frequencies)	W-H (zequencies)	Mod. Haar (rank)
0	0	0	0
1	4	7	1
2	2,-2	3,4	2,3
3	1,3,-1,-3	1,2,5,6	4,5,6,7

For the WFH families, the transforms with same sets of invariants form nested triangles as shown in Fig. 9: the introduction of additional factors leads to additional smaller subsets of coefficients of a same subzone over which energy is invariant; the relations between transforms which exist along vertical lines of the diagram of Fig. 9 preserves this energy invariance in zones. The invariants between the Generalized Discrete Transforms and the Modified Discrete Transforms have been studied by Rao et al. [18].

Conclusions

In this work we have presented a unified treatment of unitary transforms having a fast algorithm. The use of recursive rules to describe unitary transforms allows a systematic way to view known transforms, to generate new transforms and provide a general approach to the evaluation of the computational complexity of transform algorithms. Among transforms

which are clearly related, we have studied the IC_f families and the WFH subfamilies which include most of the transforms considered in the literature.

In addition to allowing the introduction of new transforms with properties of interest, the framework provided can be used in several other studies and applications of unitary transforms. In particular an error analysis of unitary transforms is being carried out.

REFERENCES

1. I. J. Good, "The Interaction Algorithm and Practical Fourier Analysis," J. Roy. Statistical Soc., Vol. B 20, pp. 361 - 372, 1958, and addendum Vol. B 22, pp. 372 - 375, 1960.
2. J. W. Cooley and J. W. Tukey, "An Algorithm for the Machine Computation of Complex Fourier Series," Math. Comput., Vol. 19, pp. 297 - 301, Apr. 1965.
3. W. M. Gentleman and G. Sande, "Fast Fourier Transform - for Fun and Profit," AFIPS, 1966 Fall Joint Com. Conf., pp. 563 - 578.
4. J. E. Whetchel and D. F. Guinn, "The Fast Fourier-Hadamard Transform and its use in Signal Representation and Classification," Eascon '68 Rec., pp. 561 - 573.
5. Proceedings, "Application of Walsh Functions," National Technical Information Service, U.S. Department of Commerce, Springfield, Virginia 22151, 1970: AD-707 431, 1971: AD-727 000, 1972: AD-744 650.
6. B. J. Fino, "Relations Between Haar and Walsh/Hadamard Transforms," Proc. IEEE (Letter), Vol. 60, No. 5, pp. 647-648, May 1972.
7. J. E. Shore, "On the Application of Haar Functions," IEEE Trans. on Comm., Vol. COM-21, No. 3, pp. 209-216, March 1973.
8. J. Duan and P. A. Wintz, "Error Free Coding," to be published.
9. H. C. Andrews, An Introduction to Mathematical Techniques in Pattern Recognition, New York: Wiley, 1972.
10. S. J. Campanella and G. S. Robinson, "A Comparison of Orthogonal Transformations for Digital Speech Processing," IEEE Trans. on Comm., Vol. COM-19, No. 6, pp. 1045-1050, Dec. 1971.

11. H. C. Andrews, Computer Techniques in Image Processing, New York: Academic Press, 1970.

12. T. S. Huang, W. F. Schreiber and O. J. Tretiak, "Image Processing," Proc. IEEE, Vol. 59, pp. 1586-1609, Nov. 1971.

13. P. A. Wintz, "Transform Picture Coding," Proc. IEEE, Vol. 60 No. 7, pp. 809-820, July 1972.

14. H. C. Andrews and J. Kane, "Kronecker Matrices Computer Implementation and Generalized Spectra," J. Ass. Comput. Mach., Vol. 17, pp. 260-268, Apr. 1970.

15. H. C. Andrews and K. L. Caspari, "A Generalized Technique for Spectral Analysis," IEEE Trans. Comput. Vol. C-19, pp. 16-25, Jan. 1970.

16. H. C. Andrews and K. L. Caspari, "Degrees of Freedom and Modular Structure in Matrix Multiplication," IEEE Trans. on Computers, Vol. C-20, No. 2, pp. 133-141, Feb. 1971.

17. N. Ahmed, K. R. Rao and R. B. Schultz, "A Generalized Discrete Transform," Proc. IEEE, Vol. 59, No. 9 , pp. 1360-1362, Sept. 1971.

18. K. R. Rao, N. Ahmed and R. B. Schultz, "A Class of Discrete Orthogonal Transforms," to be published.

19. H. F. Harmuth, Transmission of Information by Orthogonal Functions, New York: Springer, Second Edition, 1972, pp. 30-36.

20. C-K. Yuen, "Remarks on the Ordering of Walsh Functions," IEEE Trans. Comput., Corresp., Vol. C-21, No. 12, pp. 1452, Dec. 1972.

21. P. Y. Schwartz, J. Poncin and B. Fino, "Statistical Properties of Orthogonal Transforms," Proc. Conf. on Digital Processing of Signals in Communications, Vol. 23, pp. 151-174, London, Apr. 1972.

22. J. A. Glassman, "A Generalization of the Fast Fourier Transform," IEEE Trans. on Computers, Vol. C-19, No. 2, pp. 105-113, Feb. 1970.

23. R. Singleton, "An Algorithm for Computing the Mixed Radix Fast Fourier Transform," IEEE Trans. on Audio and Electroacoustic, Vol. AU-17, No. 2, pp. 93-103, June 1969.

24. R.E.A.C. Paley, "On Orthogonal Matrices," J. Math. Phys. Vol. 12, pp. 311-320, 1933.

25. J. L. Walsh, "A Closed Set of Normal Orthonormal Functions," Amer. J. Math., Vol. 55, pp. 5-24, 1923.

26. J. W. Manz, "A Sequency-Ordered Fast Walsh Transform," IEEE Trans. on Audio and Electroacoustics, Vol. AU-20, No. 3, pp. 204-205, Aug. 72.

27. H. E. Chrestenson, "A Class of Generalized Walsh Functions," Pacific J. Math., Vol. 5, pp. 17-31, 1955.

28. C. Watari, "A Generalization of Haar Functions," Tohoku Math. J., Vol. 8, pp. 286-290, 1956.

29. H. Enomoto and K. Shibata, "Orthogonal Transform Coding System for Television Signals," Proc. 1971 Symp. on Appl. of the Walsh Functions, pp. 11-17. AD-727 000.

30. W. K. Pratt, L. R. Welch and W. H. Chen, "Slant Transform for Image Coding," Proc. 1972 Symp. on Appl. of the Walsh Functions, pp. 229-234. AD-744 650.

31. W. K. Pratt, "Walsh Functions in Image Processing and Two Dimensional Filtering," Proc. 1972 Symp. on Appl. of the Walsh Functions, pp. 14-22. AD-744 650.

32. N. Ahmed and K. R. Rao, "Complex Bifore Transform," Electron. Lett., Vol. 6, No. 8, pp. 256-258, 16th Apr. 1970.

33. F. R. Ohnsorg, "Application of Walsh Functions to Complex Signals," Proc. 1970 Symp. on Applications of the Walsh Functions, pp. 123-127. AD-707 431.

34. K. R. Rao and N. Ahmed, "Modified Complex BIFORE Transform," Proc. IEEE, Vol. 60, No. 8, pp. 1010-1012, Aug. 1972.

FIGURE CAPTIONS

Fig. 1. Block diagonal matrix.

Fig. 2. Generalized Kronecker product: algorithm

Fig. 3. Fast Fourier Transform - radix 2, order 8

a. Algorithm from Fig. 2 (decimation in time)

b. Cooley-Tukey algorithm (decimation in time-in place)

c. Algorithm from Fig. 2 (decimation in frequency)

d. Sande-Tukey algorithm (decimation in frequency-in place)

Fig. 4. Walsh-Hadamard transform (order 8)

a. Algorithm - natural order

b. Algorithm with identical stages

c. Algorithm - Paley's order

Fig. 5. Haar transform (order 8)

a. Algorithm - natural order

b. Algorithm - rank order

c. Modified Haar transform.

Fig. 6. Generalized discrete transform $[GT_{16}^{2}]$

a. matrix

b. algorithm

Fig. 7. Example of a member of the family between Haar and Walsh-Hadamard transforms.

a. matrix

b. fast transform (16 adds, 4 normalizations)

Fig. 8. Generalized transform $[WHH_{16}^{2}]$

a. matrix

b. algorithm

Fig. 9. WFH family

Fig. 10. Slant transform

a. algorithm - order 8 -

b. modified algorithm - order 4 -

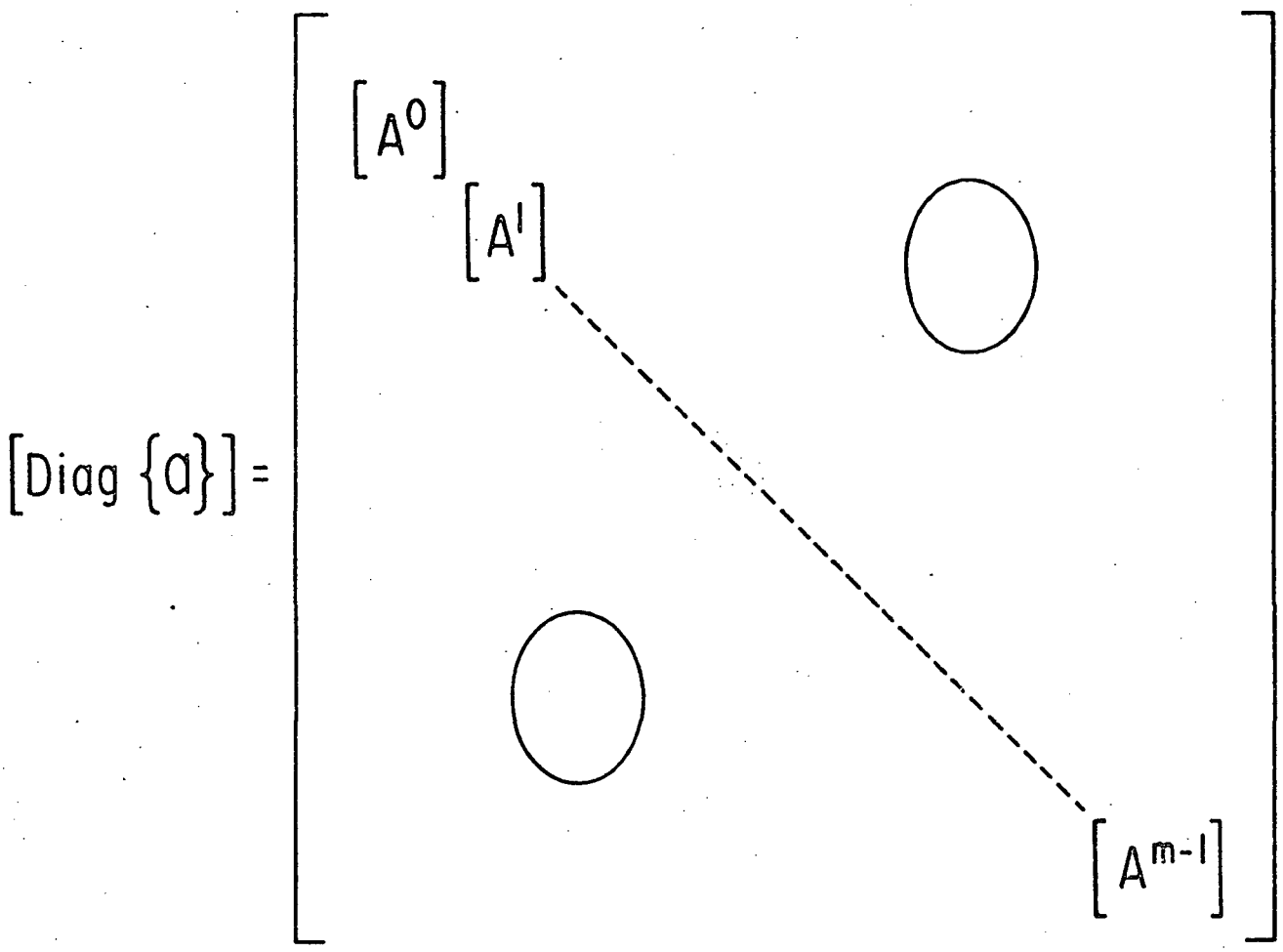


Fig. 1 : Block diagonal matrix

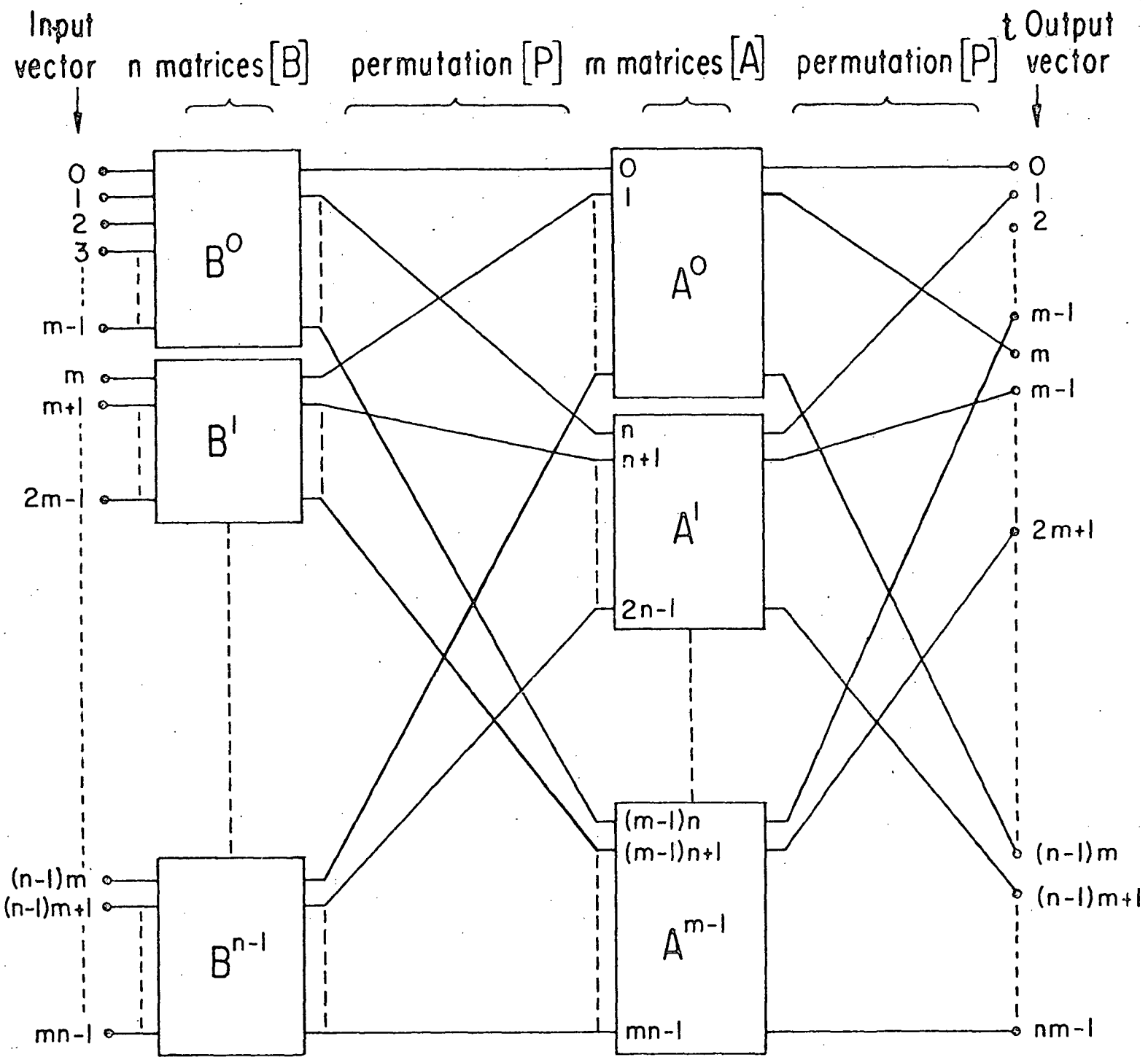
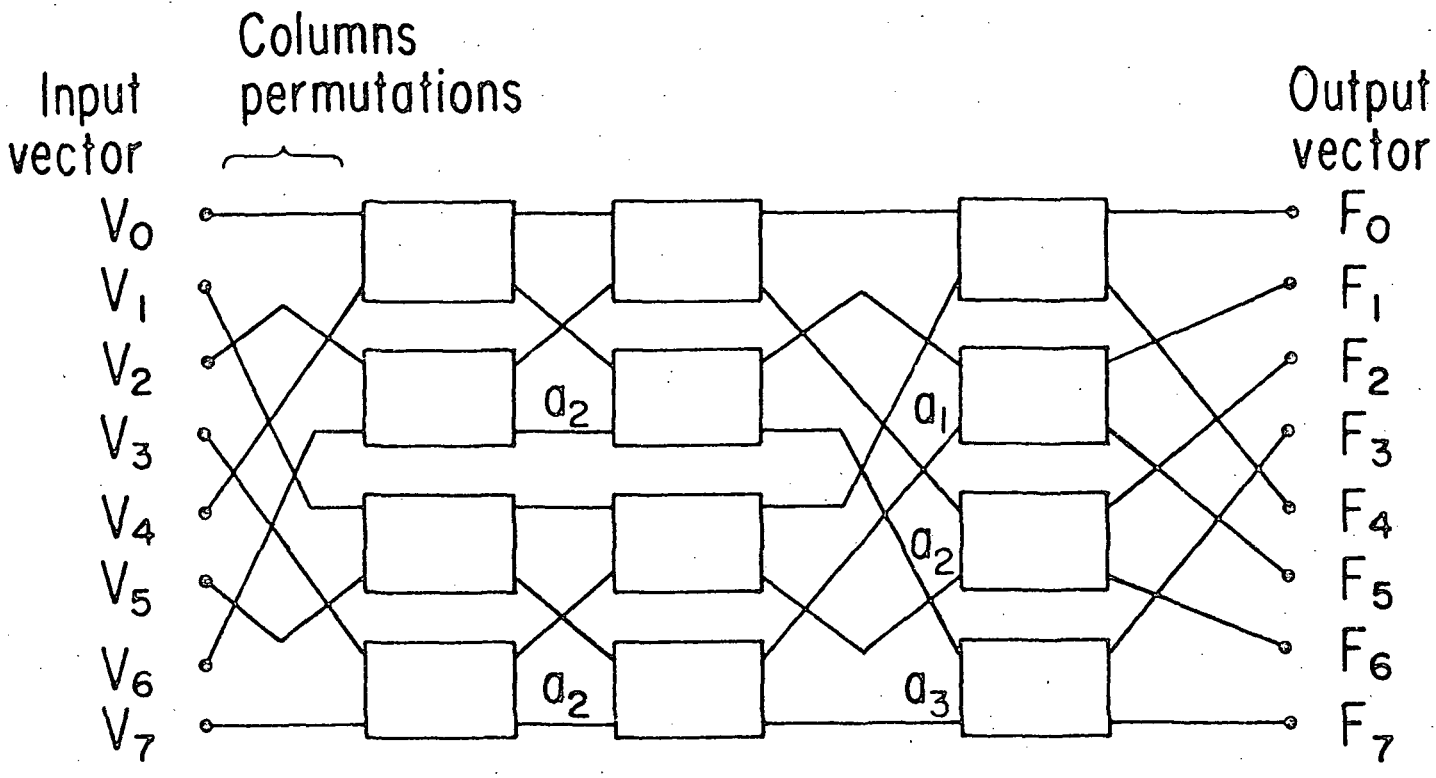
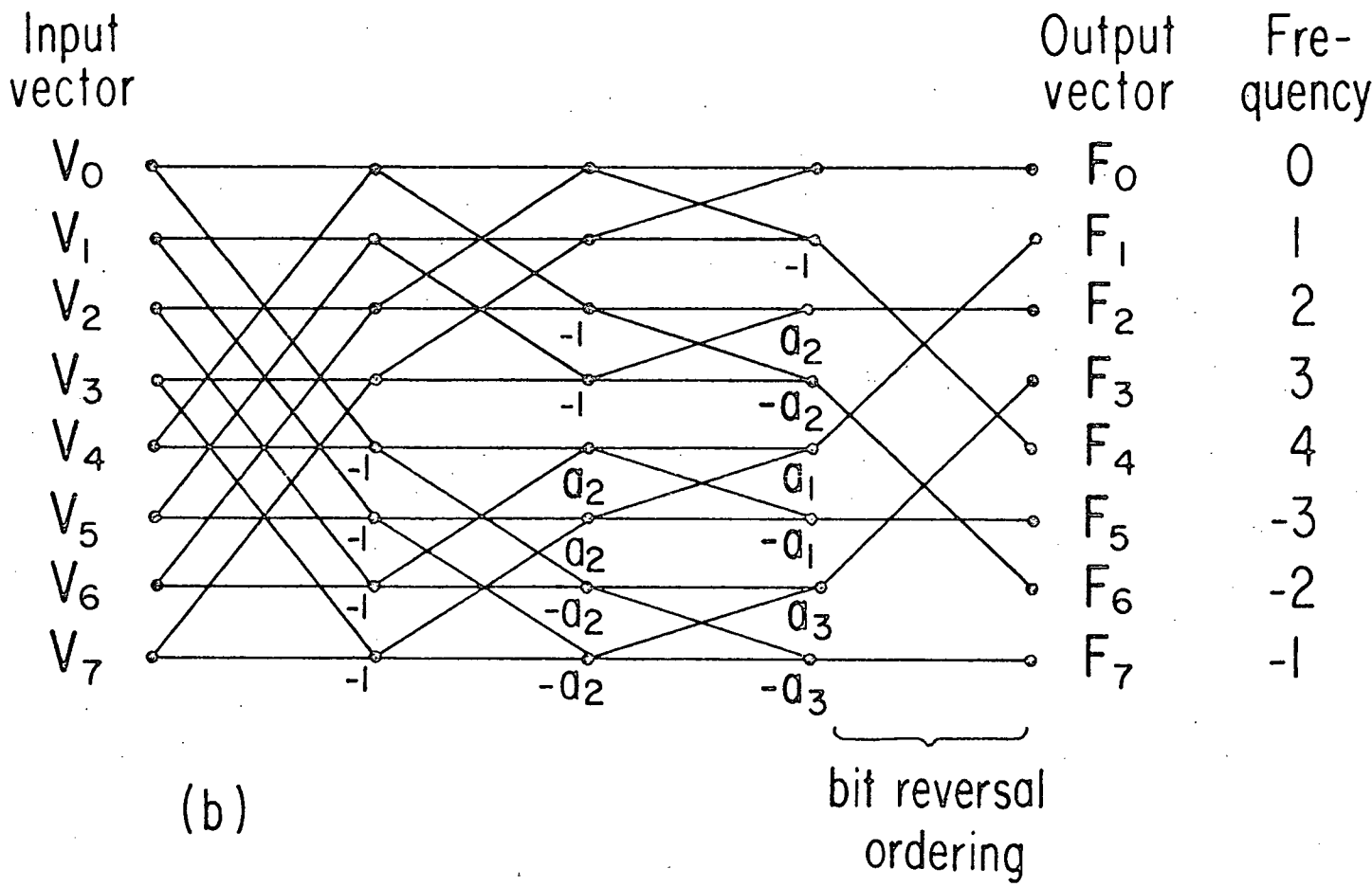


Fig. 2 : Generalized Kronecker product : algorithm



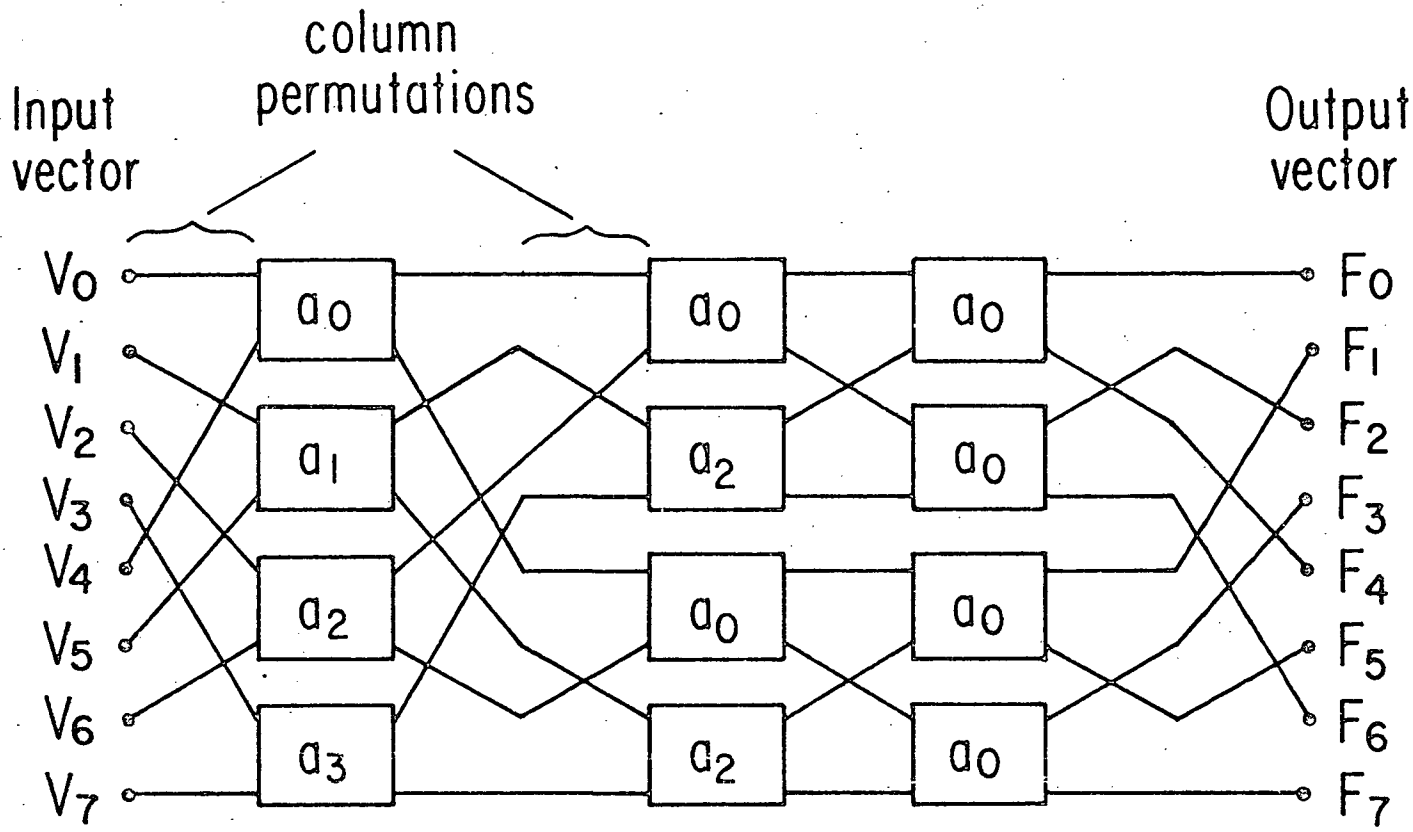
(a) Algorithm from Fig. 2

Fig. 3 : Fast Fourier Transform - radix 2, order 8-



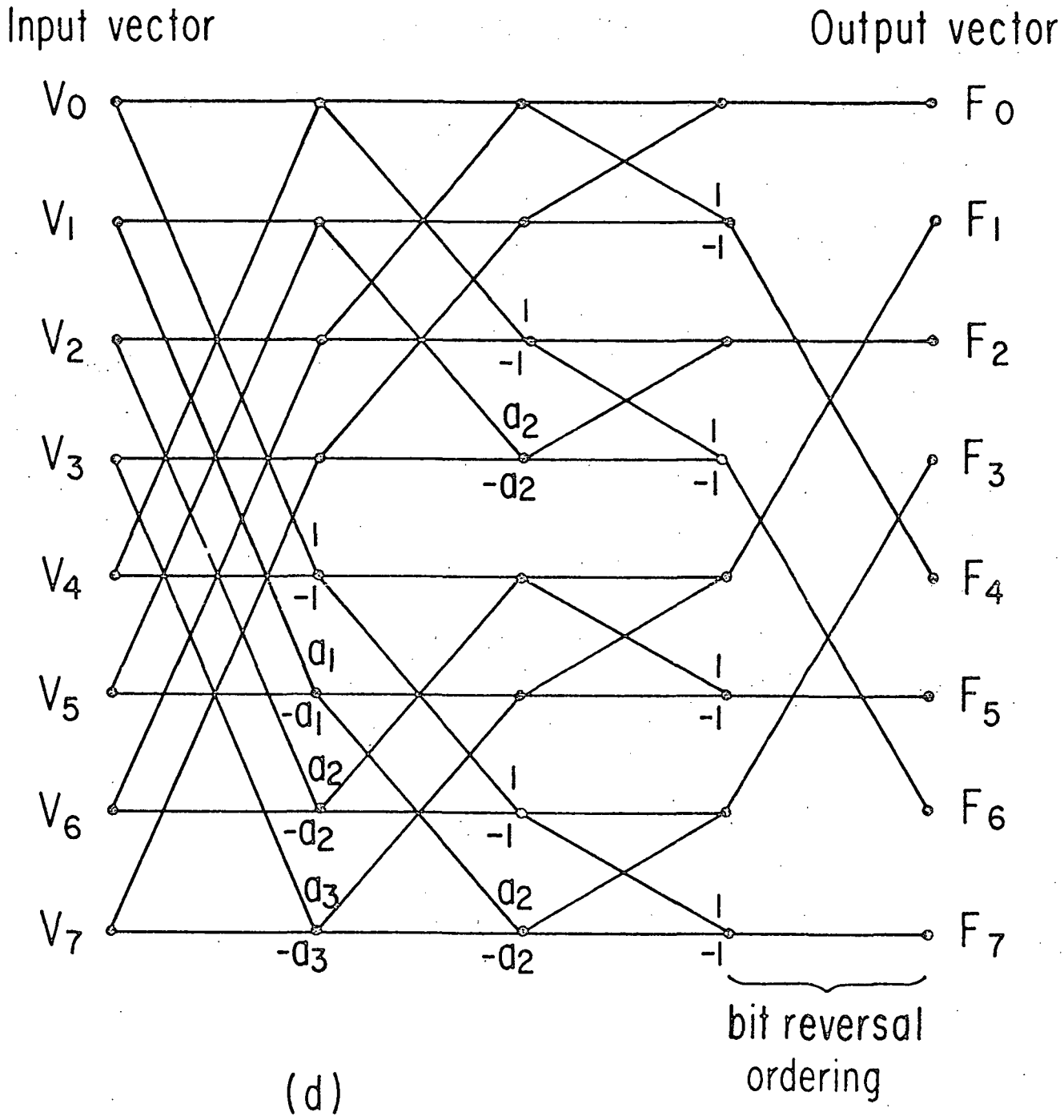
Cooley-Tukey algorithm - decimation in time, in place -

Fig. 3 Fast Fourier Transform - radix 2, order 8 -



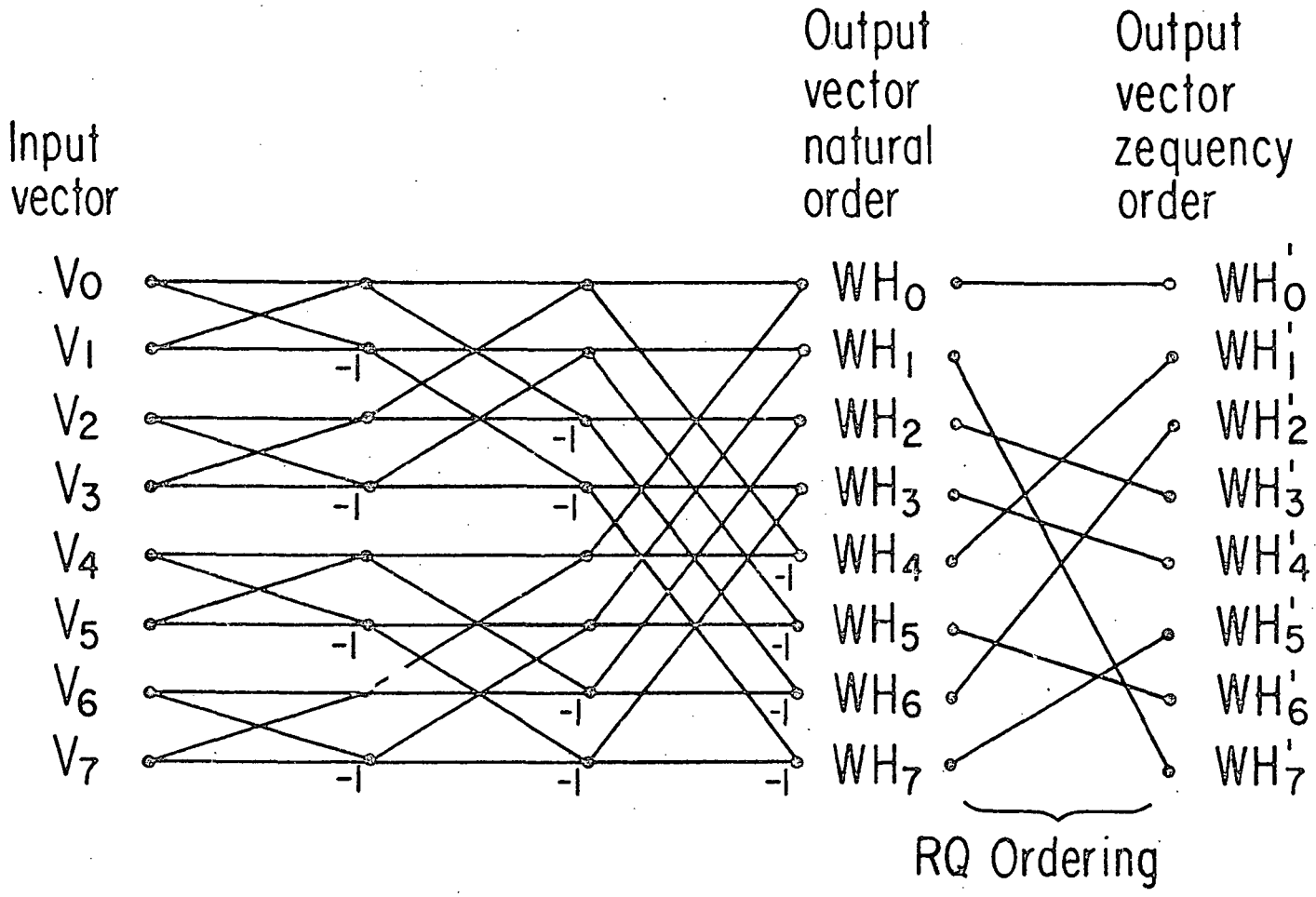
(c) Algorithm from Fig. 2 - decimation in frequency -

Fig. 3 : Fast Fourier Transform - radix 2, order 8 -



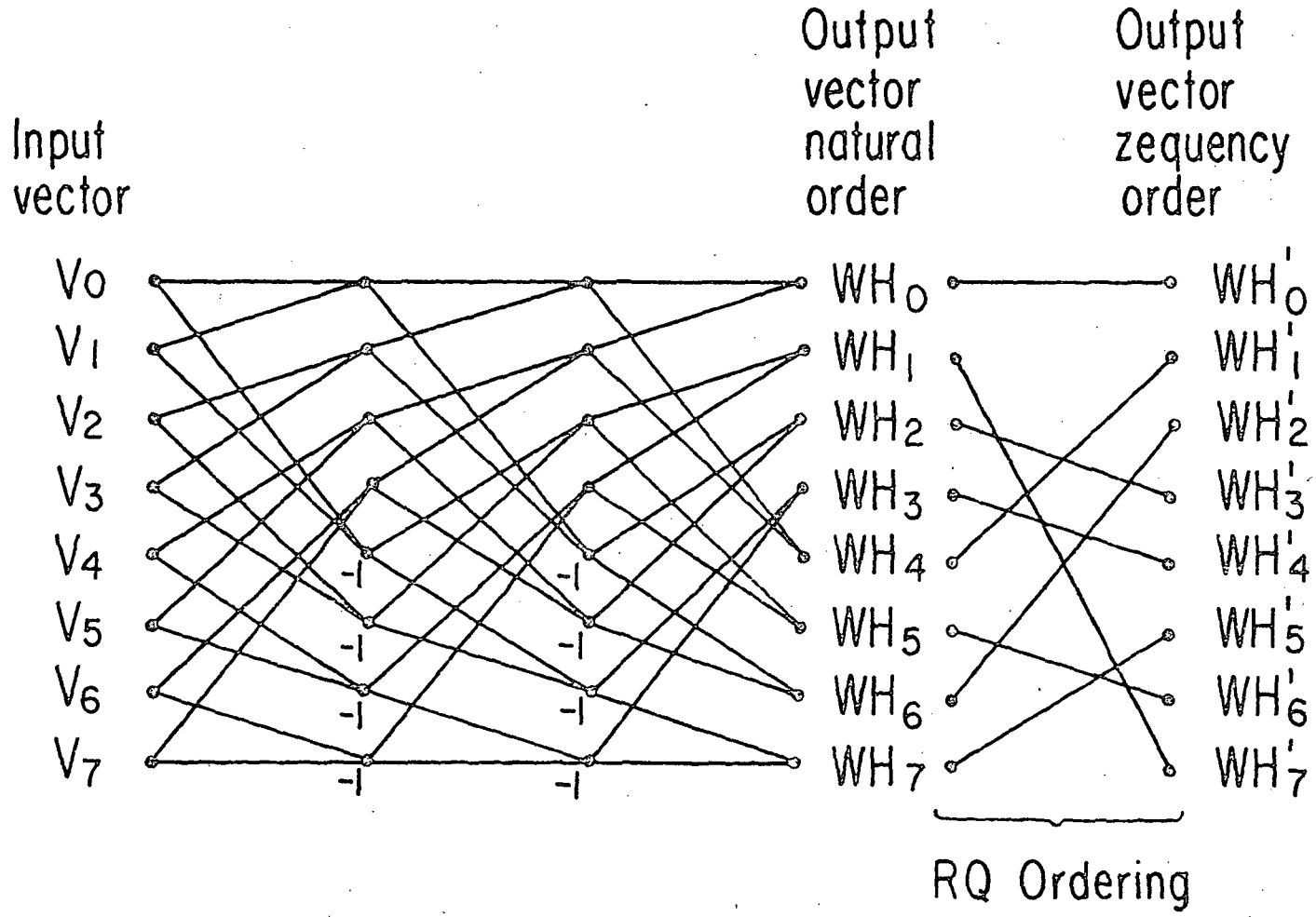
Sande-Tukey algorithm -decimation in frequency, in place -

Fig. 3 : Fast Fourier Transform -radix 2, order 8 -



(a) Algorithm - natural order -

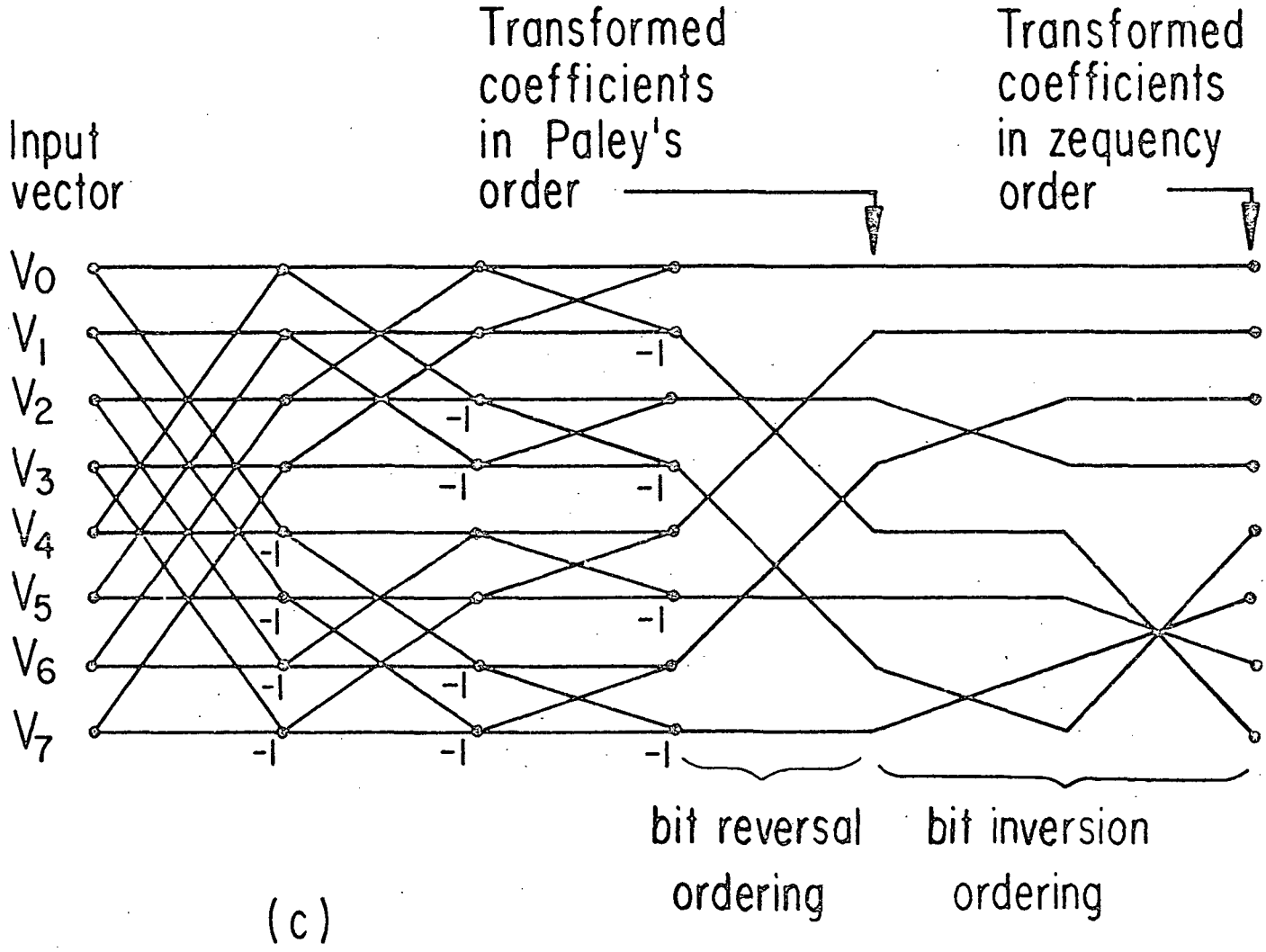
Fig. 4 : Walsh-Hadamard Transform - order 8 -



(b)

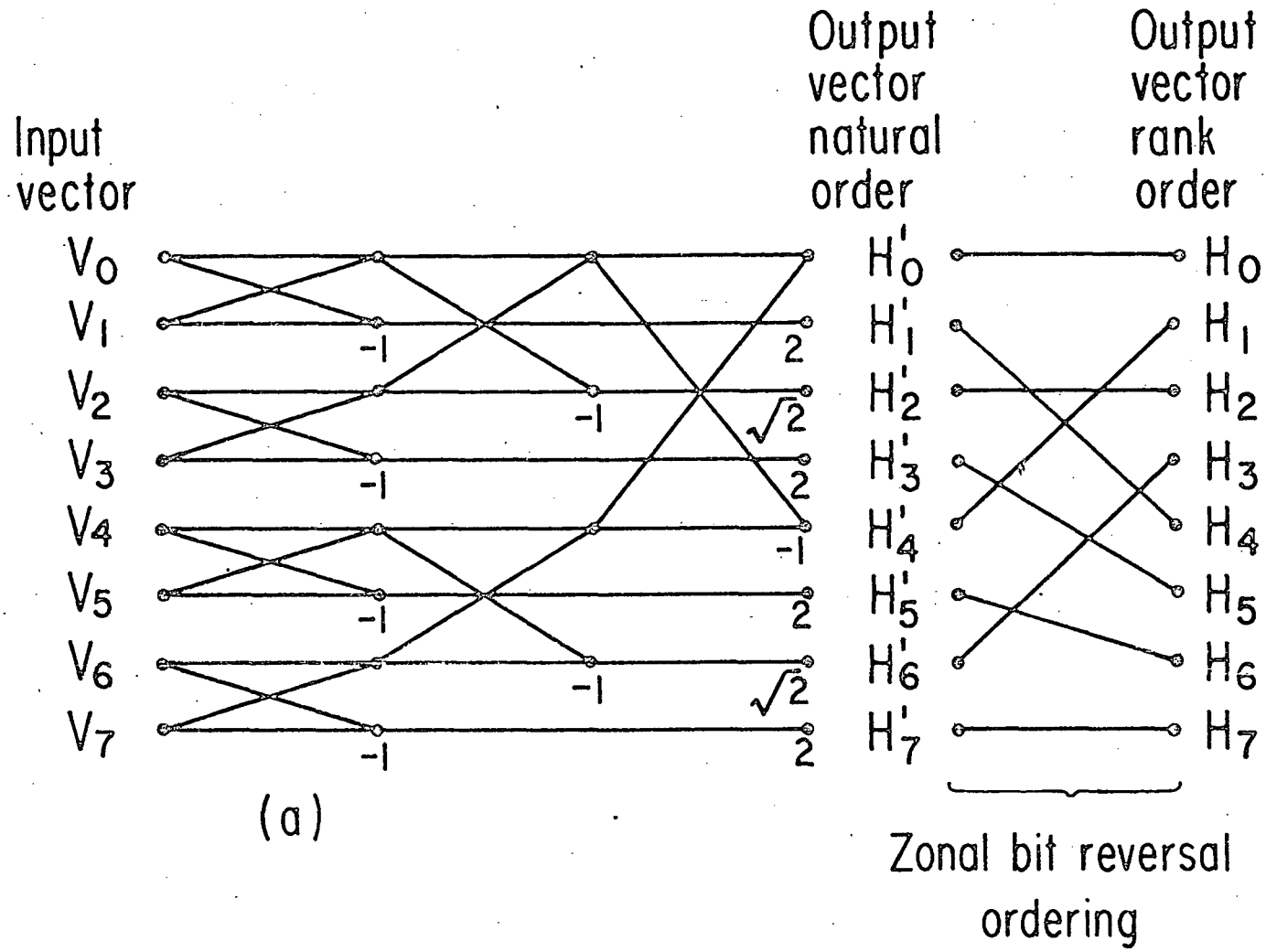
Algorithm with identical stages

Fig. 4 : Walsh-Hadamard Transform - order 8 -



Algorithm - Paley's order

Fig. 4 : Walsh-Hadamard Transform - order 8 -



Algorithm - natural order -

Fig. 5 : Fast Haar Transform - order 8 -