TRANSFORM CODING OF PICTORIAL DATA

By

W. K. CHAM

A Doctoral Thesis

Submitted in partial fulfilment of the requirements

for the award of

Doctor of Philosophy

of the Loughborough University of Technology

SYNOPSIS

By using transform coding, image transmission rates as low as 0.5 bit/pel can be achieved. Generally, the bit rate reduction is achieved by allocating fewer bits to low energy high order coefficients. However,to ensure reasonably good picture quality, a large number of bits has to be allocated to high energy dc coefficients for both fine quantization and good channel error immunity. A technique has been developed that, in some cases, allows the dc coefficients to be estimated at the receiver, thus eliminating a major source of difficulty with respect to channel errors. The computational requirement depends on the number of dc coefficients, M , which is equal to the number of blocks within the picture. In practice, M is large and so the computational load is substantial, and therefore, to make the method practical, three modified schemes called ELEMENT ESTIMATION, ROW ESTIMATION, and PLANE ESTIMATION are proposed, all requiring reduced computation time and memory. Results of simulations of these methods using different block sizes and different degrees of ac coefficient truncation are shown.

A new unified matrix treatment of Walsh transforms using the concept of dyadic symmetry is then developed. This treatment allows the straightforward derivation of a simple equation for the generation of Walsh matrices of different orderings, various re-ordering schemes and fast computational algorithms.

As the theory relates to a binary field with 'logical and' and modulo two addition as operations,it allows both the generation of Walsh matrices of different orderings, and reordering schemes, to be carried out using simple logic circuits.

The theory of dyadic symmetry is then used to generate two new transforms which can be used for image processing. The new transforms have virtually the same complexity and computational requirements as the Walsh transform, employing additions, subtractions and binary shifts only but with an improved efficiency, defined in terms of its ability to decorrelate signal elements, which lies between that of the Walsh transform and that of the discrete cosine transform.

ACKNOWLEDGEMENT

I would like to thank my supervisor, Mr R.J.Clarke for his help and assistance throughout the course of the research program and especially his patient reading and useful comments in the preparation of this thesis. I also wish to express my most sincere thanks to Mr R.J.Clarke and Dr K.N.Ngan for their efforts to obtain the studentship for me, and to the British Telecom Research Establishment for the financial support which made it possible for me to carry out this work.

Special thanks also to Mr D.Allott for checking the final draft of the thesis and to Mr P.Atkinson for preparing the photographic results. I am most grateful to all my colleagues in the Communications Research Laboratory of Loughborough University with whom I have enjoyed every moment of my past three years research.

Finally, I wish to express my deepest gratitude to my parents for their moral and financial support throughout the course of my whole education.

LIST OF PRINCIPAL SYMBOLS AND ABBREVIATIONS

a	:	A positive constant less than unity (Table 4.1)
A	:	Vector containing dc coefficients (eq.5.13)
a_i	:	The i th element in A
$a(i,j)$	:	The dc coefficient in the (i,j)th block (eq.5.5)
$[B]$	:	The binary Walsh matrix
b_{ij}	:	The (i,j)th element of the binary Walsh matrix (eq.3.9)
C	:	Vector in transform domain (c.f. X)
c_i	:	The i th transform coefficient in vector C
CQ	:	The quantized value of C
$[CC]$	:	Covariance matrix of C (eq.4.4)
$[CX]$	:	Covariance matrix of X (eq.4.2)
D	:	Distortion (eq.1.1)
$[D]$	:	Dyadic-ordered dyadic symmetry matrix (eq.3.15)
$D_{1,j}$	:	The j th vertical edge vector (eq.5.14a)
$D_{2,j}$	:	The j th horizontal edge vector (eq.5.14b)
$d_{p,j,q}$	:	The q th element of the vector $D_{p,j}$

DCT	:	The discrete cosine transform (Fig.2.4)
DFT	:	The discrete Fourier transform
DPCM	:	Differential pulse code modulation
DST	:	The discrete sine transform (Fig.2.5)
E[ ]	:	Expected value of the variable in [ ]
F	:	Field (Def.2.2)
HCT	:	The high correlation transform (Fig.4.3)
H(w)	:	Relative sensitivity of the human visual system to spatial light intensity distribution w (Fig.1.11)
[H]	:	The Walsh matrix (eq.3.7)
h_{ij}	:	The (i,j)th element of the Walsh transform
i	:	Row index or index running vertically
i_m	:	The m th bits of i (i_1 is the msb)
j	:	Column index or index running horizontally
j_m	:	The m th bit of j (j_1 is the msb)
KLT	:	The Karhunen-Loeve transform
LCT	:	The low correlation transform (Fig.4.4)
n	:	Block size
[N]	:	Natural-ordered dyadic symmetry matrix (eq.3.12)

NMSE	:	Normalized mean square error (eq.4.9)
pdf	:	Probability density function
PER_r	:	Percentage of energy packed into the first r+1 transform coefficients (eq.4.6)
$Rad_i(t)$	:	The i th Rademacher function (Fig.3.1)
R(D)	:	Rate distortion function
$[S]$	:	Dyadic symmetry matrix
S_i	:	The i th row vector in $[S]$
$[T]$	:	Transform kernel
T_i	:	The i th row vector or basis vector of $[T]$
T_i	:	The i th column vector of $[T]$
UHCT	:	The unnormalized HCT
ULCT	:	The unnormalized LCT
V	:	Edge vector of dc basis picture (eq.5.4)
V_n^r	:	Vector space of dimension r consisting of n-vectors over F
X	:	Vector in signal domain (c.f. C)
XQ	:	The quantized value of X
$[Z]$	:	Sequency-ordered dyadic symmetry matrix (eq.3.17)

γ	: Gamma, parameter of a camera tube (Fig.1.10)
ΔL	: Visual threshold (Fig.1.9)
σ	: Standard deviation
σ^2	: Variance
η	: Transform efficiency (eq.4.5)
ρ	: Adjacent element correlation
$[\ ]^t$	: Transpose of [ ]
$[\ ]^{-1}$	: Inverse of [ ]
*	: 'Logical and'
(+)	: 'Exclusive or' (i.e. binary two addition)

C O N T E N T S

CHAPTER FOUR -- NEW TRANSFORMS

CHAPTER SIX -- RECAPITULATION AND SUGGESTIONS FOR FUTURE WORK

CHAPTER ONE

INTRODUCTION

1.1 INTRODUCTION

The material contained in this thesis all relates to an image data compression technique called transform coding. The research into this area described here was initiated by the new development of the British viewdata system, Prestel. The next section describes briefly what a viewdata system is and how the work reported here relates to it, in particular to the Prestel system. A review of transform coding is then given in section 1.3. It begins with the general theory of picture coding and finally concentrates on the various aspects of transform coding which are discussed separately under four sections. Finally, in the last section, the organization of the thesis is outlined.

1.2 MOTIVATION FOR THE WORK

Using a viewdata system, television viewers need not wait for the 9 o'clock news to get the weather forecast, the latest development in an international crisis or the financial news. The system transmits the information requested by its users from its information retrieval centres via the telephone network to a special decoder in the users' home for display on the TV set.

Simply, a viewdata system is a network of computer centres of two types, information retrieval centres (IRCs) and update centres (UDCs), interconnected together by high-speed data links as shown in Fig.1.1. A viewdata system has two types of user, information retrievers (IRs) and information providers (IPs). An information retriever is a telephone subscriber, who, with the help of a hardware interface (the special decoder), can make contact with one of the information retrieval centres and retrieve wanted information from a data base within the computer network and have it displayed on a TV screen. Another type of user is the information provider, who is either a person or an organization authorized to have the right to update information contained in particular pages or files within a viewdata database by means of a special editing terminal connected to an update centre.

Many countries, like the United Kingdom, Canada, France and Japan have already started developing their own viewdata systems. The United Kingdom started the venture before all other competitors and developed the world's first operational viewdata system, Prestel, which at present can deal with two types of information: alphanumeric and graphical. British Telecom is planning to upgrade the present viewdata system so that computer programs as well as images can also be handled. With the the present data transmission rate (1.2 kbps) over British public telephone network, transmission of a digital picture, occupying a quarter of a TV screen with a resolution of 256 x 256 pels, requires a transmission time of about seven and a half minutes. Therefore, data compression techniques have to be employed to shorten the transmission time. Also, to ensure good picture qua-

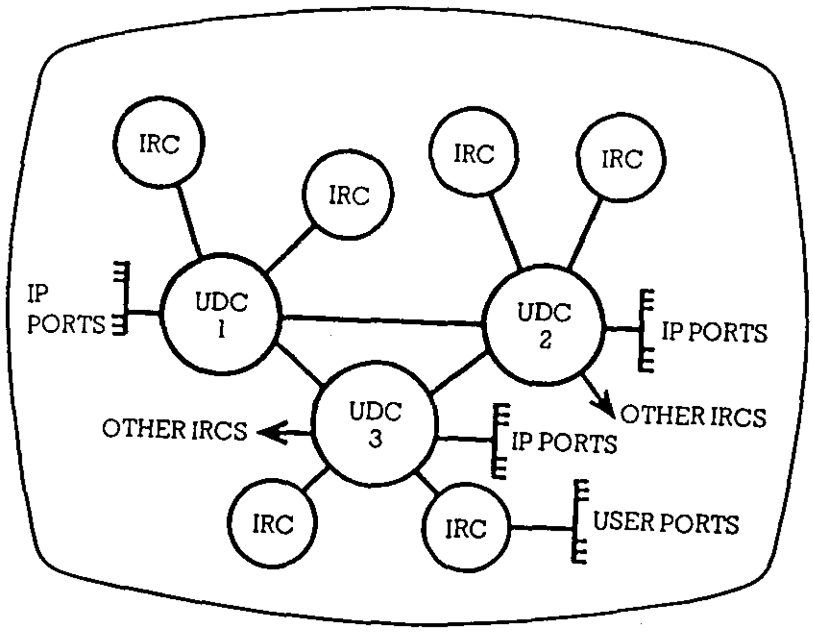


Fig.1.1 Network topography of a viewdata system.

lity, error detection and correction techniques have to be used to safeguard the data from corruption by channel noise. The research work described in this thesis examines various aspects of a data compression technique called transform coding, which (at present) can reduce the transmission time from seven and a half minutes (8 bit/pel) down to about twenty seven seconds (0.5 bit/pel).

1.3 TRANSFORM CODING ------- A REVIEW

A video system typically starts with a two-dimensional distribution of light intensity. The two-dimensional light intensity is usually raster scanned by a TV camera to provide a one-dimensional signal. The signal waveforms from most TV cameras are often companded, i.e. made a compressed nonlinear function of scene luminance. Since in most cases eight-bit uniform quantization of this companded signal gives imperceptible quantization noise, the companded waveform is usually represented as a two-dimensional array of eight-bit picture elements. Further, there are 287.5 visible lines in one field, so the array size is for convenience often taken as 256 x 256 and thus represents a huge amount of information.

However, members of this large number of picture elements are generally highly correlated and the image contains significant structure. For example, pictures may consist of many areas exhibiting a repetitive pattern analogous to the texture of cloth or the pattern of a tile floor. Studies and different classifications of texture [1-4] have been carried out over the past few years. In addition, pictures often contain a number of areas of nearly constant brightness. Statistics on the number of these areas, their brightness, sizes, etc, have been collected [5-6]. A definite structure also exists in the boundaries between these areas, which are usually sharp edges, and studies of these edges have also reported in the literature [7-9].

Ideally, one would like to take advantage of this redundancy and structure in pictorial data, so that pictures can be encoded or represented using fewer bits, hence needing less storage space and less transmission time. Encoding of such signals is performed by a myriad of different techniques which can be divided into two classes ---- waveform coding and parameter coding. The objective of waveform coding is simply replication of waveforms, whereas parameter coding attempts to represent the image using the basic features necessary in some specific applications. Parameters of these basic features are extracted at the transmitter, transmitted through the channel, and then used to synthesize the image.

So far, no universal model is able to represent all images sucessfully because of the immense variations between different image sources. However, models have been proposed to represent a restricted set of images with good results. For example, images can be modelled as random concatenations of textures [10-12]. Hence, a picture can be represented by parameters for the texture, and the position and orientation of edges. In another example, a system for the transmission of a 'head-and-shoulder' image builds and maintains a 3-D model of the object to be coded. Parameters of the facial expressions of the object are extracted at the transmitter, and sent through the channel to update the model at the receiver. The reproduced image at the receiver is a 2-D projection of this model [13].

The other class of coding technique, waveform coding, can again be divided into two main categories ---- predictive coding and transform coding. Fig.1.2 shows the block diagram of a predictive coding system which is often called differential pulse code modulation (DPCM). The sample to be encoded is predicted from the encoded values of the previously transmitted samples and only the prediction error is quantized for transmission. Research work in predictive coding is mainly concentrated on the improvement of the predictor and the quantizer by making them optimal for a particular type of source, or adaptive to local statistics by employing sophisticated algorithms. For example, various forms of switched predictors to deal with the sharp changes at the boundaries of textures, and motion compensated predictors to deal with moving objects, have been evolved to minimize the prediction error. On the other hand, various adaptive quantizers [15-18] have been designed to minimize the quantization noise of the prediction error. In general, all predictive coders achieve data compression by exploiting redundancy in the data.

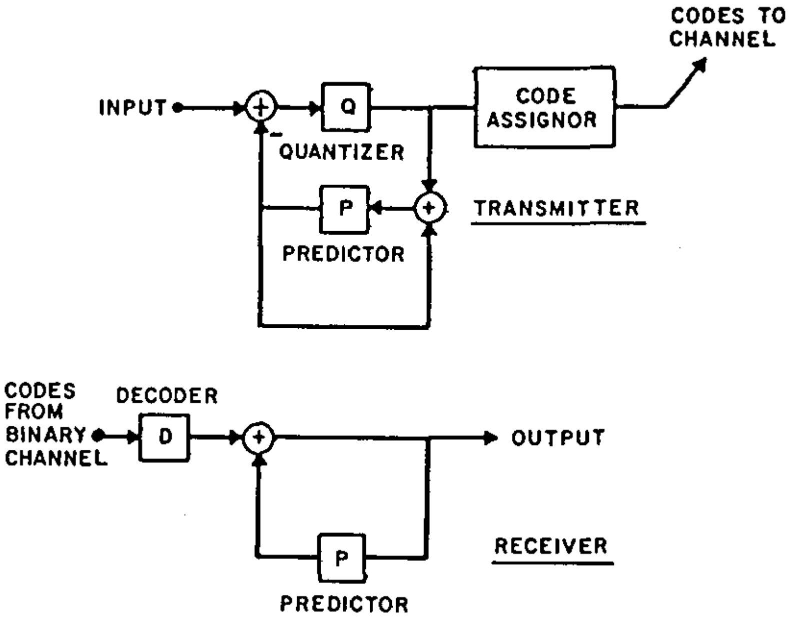


Fig.1.2 Block diagram of a DPCM system

In transform coding a completely different approach is used. Fig.1.3 shows the block diagram of of a transform coding system. The original image is divided into subpictures of a particular block size and transformed into sets of weakly correlated coefficients. The coefficients are then quantized and coded for transmission. At the receiver, the received bits are decoded into transform coefficients, and an inverse transform is applied to the coefficients to return to the picture domain.

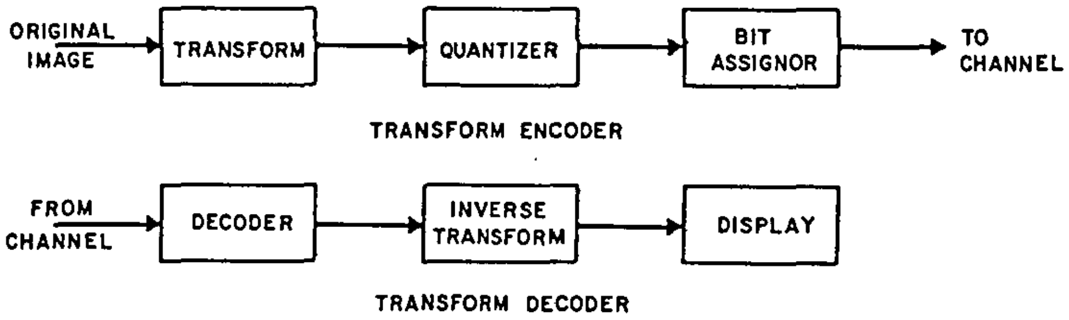


Fig.1.3 Block diagram of a transform coding system

Jain has shown that transform coding, although requiring more processing sophistication, achieves a higher degree of data compression than predictive coding for a one-dimensional Markov process [14]. On real images, the two schemes were found to perform quite closely at very low distortion, but transform coding is distinctly better at high values of distortion. In practice, only adaptive predictive coding algorithms can achieve the efficiency of even nonadaptive transform coding methods [14].

Data compression in transform coding is achieved by the transformation of the subpicture into another array such that maximum information is packed into a minimum number of coefficients. Therefore, the overall quantization error can be minimized by allocating more bits for transmission of coefficients having larger variances, and fewer bits for coefficients having smaller variances. The appropriate bit allocation can be derived from rate distortion theory which states that [19]:

the output of a source can be transmitted with average distortion D if the transmission rate is larger than R(D).

If D is the mean square error and the source has Gaussian probability distribution, then R(D) is found to be [20]

$$R(D) = \begin{cases} \log \sigma / \sqrt{D} & \sigma > \sqrt{D} \\ 0 & \sigma \leqslant \sqrt{D} \end{cases} \qquad \text{--------------(1.1)}$$

where σ is the standard deviation. Therefore, equation 1.1 can be used to determine the number of bits required for each transform coefficient [21-22].

Transform coding is a natural outgrowth of the principle of rate distortion. It was first applied to one-dimensional signals and later applied to picture coding [23-27,99]. Over the years, much effort has been devoted to the transform coding of pictorial data. Here, discussion of work on transform coding is grouped under the following four headings: 1) transformations, 2) adaptive schemes, 3) schemes exploiting interblock redundancy, and finally 4) schemes based on human psychovisual characteristics.

1.3.1 Transformations

This is the most important part of transform coding theory, and the detailed theory of transformation will be presented in chapter two. Basically, the primary purpose of the transformation is to convert statistically dependent picture elements into an array of uncorrelated coefficients such that maximum energy is packed into a minimum number of coefficients (the total energy in the transform domain remaining the same as that in the picture domain).

For a particular image, the optimal transformation which satisfies the criteria mentioned above is the Karhunen-Loeve transform (KLT) [28-29] (otherwise known as the Hotelling transform [30]) whose basis vectors are in fact the eigenvectors of the covariance matrix of the image. However, its practical application is beset by many problems.

Firstly, the KLT necessitates the computation of the eigenvectors which requires extra computational time, complicates implementation and, furthermore, sometimes the eigenvectors cannot be uniquely defined. In addition, there is no true fast computational algorithm for the KLT and extra bits are required for the transmission of either the basis vectors or the covariance matrix. All the above problems prevent the KLT from being used in practice.

Jain et al [31-32] have developed a fast KLT for a class of stochastic processes, which however, do not represent a typical image. On the other hand, the KLT for the first-order Markov process having the covariance matrix given by equation 2.32 (widely accepted as a good model for images), has no known fast computational algorithm [85]. All these problems can be eased by the application of a suboptimal transform. The first suboptimal transform to be investigated for image coding was the two-dimensional Fourier transform (Andrews and Pratt [99]). This was followed shortly by the discovery that the

Walsh transform could be utilized in place of the Fourier transform with a considerable decrease in computational requirement [101].

In 1971, investigation began into the application of the KLT [28-30] and Haar transform [34]. As mentioned above, application of the KLT in practice is prevented due to its complexity. On the other hand, the Haar transform has an extremely efficient computational algorithm, but results in a larger coding error. At about the same time, Enomoto and Shibata designed a new 8 x 8 transform to match typical image vectors [35]. Pratt generalized this transform [36] which is known as the slant transform, and later applied it to image coding with a fast computational algorithm [37] resulting in a lower mean square error for moderate block sizes in comparison to other unitary transforms. Many other transforms such as the DLB (Discrete Linear Basis) [38], Slant Haar Transform [39], SVD (Singular Value Decomposition) [40] and Modified Slant transform and Modified Slant Haar transform [41] have also been proposed for image coding. However, the discovery of the discrete cosine transform (DCT) in 1974 [42], its efficient fast computational algorithm in 1977 [43], and later its application in image coding via the fast computational algorithm [22] has generated much interest. Comparisons between the DCT and other suboptimal transforms using a stochastic image model have shown that the DCT results in the least mean square error [22] and in fact the DCT is asymptotically close to the KLT for the first-order Markov process of covariance matrix given by equation 2.32 [44-46]. Jain has suggested a sine transform with similar properties [31].

The DCT requires real number multiplications whilst the Walsh transform needs only additions and subtractions. In some cases, for example coding of moving pictures, a simple and efficient transform is still necessary. This is why a real time digital image coding system reported recently still adopts the Walsh transform [47]. In view of performance and simplicity, the choice of transformations lies very much between the Walsh transform and the DCT depending on whether or not processing speed is paramount.

In chapter four, two new transforms which can be used as substitutes for the Walsh transform are described. Both transforms have virtually the same complexity and computational requirements as those of the Walsh transform but their energy packing ability and decorrelation efficiency lie between those of the Walsh transform and of the DCT.

1.3.2 Adaptive schemes

Pictorial data is not homogeneous --- some regions of a picture consist of highly correlated pels and some regions contain a high degree of activity. Optimal nonadaptive coding schemes are matched to the average statistics of the whole picture. Adaptive coding schemes compute local statistics and then apply an algorithm that is efficient for those statistics. At the expense of increased complexity and computation time, adaptive coding schemes always outperform nonadaptive ones. Numerous adaptive schemes have been proposed. The main differences between them lie in the answer to one crucial question. How does the transmitter inform the receiver of the coding strategy it has employed for each particular section of the encoded picture?

There are two extreme ways of tackling this problem. In the first all the overhead information about adaptation is sent to the receiver; the other bases its adaptation completely upon previously transmitted data and no overhead information is sent.

The former way is perhaps best represented by theshold coding. Using this method one selects a threshold level and transmits only the transform coefficients that are larger than this threshold. This method is highly adaptive because the number and location of the coefficients that are larger than a threshold vary from one subpicture to another. Dillard [48] used this scheme for a 4 x 4 Walsh transform in which the dc coefficient and the largest ac coefficients are sent along with their addressing information. However, the addressing information without any compression could account for about 60% of the total bit rate [49], although run-length coding algorithm [50] and entropy coding [49] can be used to compress the addressing information. Good results have been reported at about 1.25 bit/pel [49].

Moving away from threshold coding, which uses up many bits on overhead information to achieve high adaptivity, is another type of adaptive scheme which will be called block classification coding in this thesis. Such schemes sort transform blocks into classes by the level of image activity present. Claire [51] and Gimlett [52] proposed a definition of 'activity index' using a weighted sum of the absolute values of the transform coefficients. Therefore, by allocating more bits to those blocks having a higher activity index, and fewer to those having a lower activity index, adaptation is achieved with addressing information considerably reduced. For example, Chen and Smith [22] divided transform blocks into four classes, each with equal numbers of transform blocks. Fig.1.4 and 1.5 illustrate respectively a typical classification map and bit allocation matrices for the four classes for the monochrome image 'girl' (Fig.4.8a) coded with an average of 1.0 bit/pel using Chen and Smith's system. In another example, Tasto and Wintz [33] classified image blocks into three categories according to the luminance activity.

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

Fig.1.4 A typical classification map for the 'Girl' picture in the Chen and Smith system.

In these systems, the addressing information required to be sent usually includes one classification map, and a set of variance matrices which are used to derive the corresponding bit allocation maps. Some systems, however, transmit the bit allocation matrices from which the variance matrices are estimated, with further bit rate reduction but at the cost of less accurate results [22]. Another group of schemes, which are sometimes known as recursive quantization techniques [50]

```
8666555444434333
6544443333322221
5443333222222210
4433332222111000
4443322211110000
4433222211100000
4333222111000000
4332322110000000
3222221100000000
3222211000000000
3221110000000000
3321100000000000
3221000000000000
2220000000000000
3200000000000000
2100000000000000
```

(a) The bit allocation map for class 4 (the higest activity).

```
8655554444343333
5443333322322221
4433222221111100
4333322222211000
4433222211110000
4332222211100000
4332221111000000
4322221110000000
3322211100000000
3222211000000000
2222110000000000
2221100000000000
2221100000000000
2210000000000000
2100000000000000
2100000000000000
```

(b) The bit allocation map for class 3.

```
8544433333222222
5433322222211110
4322211111110000
3332211111101000
3322111111000000
3222111100000000
3221111110000000
3221110000000000
2211100000000000
2111100000000000
2111100000000000
1111000000000000
1111000000000000
1100000000000000
1100000000000000
0000000000000000
```

(c) The bit allocation map for class 2.

```
8321001101100010
2111001001111100
2101111101100100
1101000111111000
1010101110110000
1001101111100000
1000010110000000
0111101100000000
0001011000000000
0001000000000000
1010000000000000
0011100000000000
1100000000000000
1110000000000000
1000000000000000
1000000000000000
```

(d) The bit allocation map for class 1 (the lowest activity).

Fig. 1.5 Typical bit allocation maps for the 'girl' picture with 1.0 bit/pel in the Chen and Smith system.

go further in eliminating the transmission of the variance matrices and classification map. Tescher et al [53-54], instead of sending the variance map of the DFT coefficient magnitudes, estimated them at the receiver using a predictor that predicts the variance of a given coefficient from the variances of a number of adjacent quantized elements. This system only needs the transmission of a few variances to start the estimation process at the receiver. Bits are then allocated to each coefficient proportional to the logarithm of the estimated coefficient variance, with the phase component having twice the number of quantization levels of the magnitude component. The same algorithm has also been investigated using the Hadamard transform.

In a different approach,Tescher and Cox [55], using a diagonal scanning pattern, converted the two-dimensional variance map into one--dimensional format as shown in Fig.1.6. Then, the i th coefficient variance is estimated as

$$\hat{\sigma}_i^2 = a_1 \hat{\sigma}_{i-1}^2 + (1 - a_1) \hat{x}_{i-1}^2 \qquad \text{---------(} 1.2 \text{)}$$

where $\hat{x}_{i-1}$ is the i-1th quantized coefficient and a_1 is a weighting factor which is chosen to be 0.75. The number of bits allocated for the quantization of x_i is proportional to the logarithm of the estimated variance $\hat{\sigma}_i^2$.

In another approach, Wong and Steele estimated the (r,s)th variance as

$$\hat{\sigma}(r,s)^2 = \exp\{ c_1 \ln[Q(r,s)] + c_2 \} \qquad \text{----(} 1.3 \text{)}$$

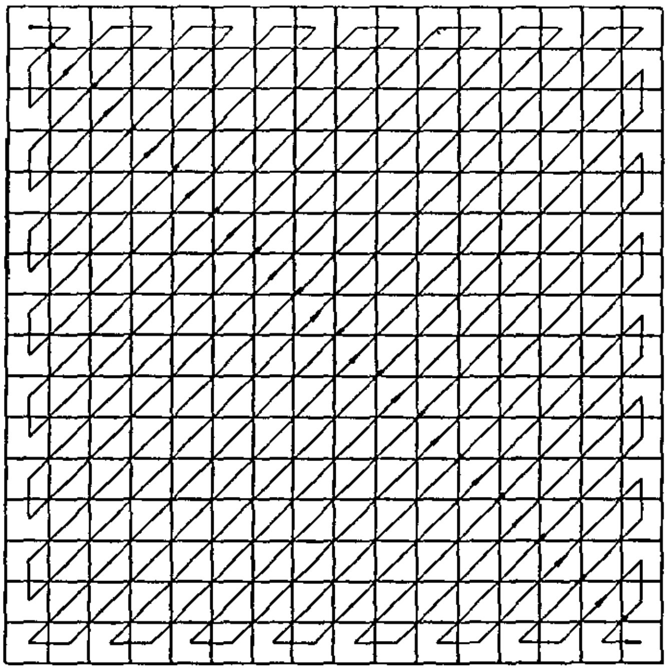


Fig.1.6 Scan path used in variance estimation

where Q(r,s) is a distance factor and c_1 and c_2 represent, respectively, the slope and intercept of the log-log relationship of Q(r,s) and $\sigma(r,s)$. Bits are allocated to each coefficient proportional to the logarithm of the estimated variance [56-57].

However, Recursive quantization suffers from one big drawback. A transmission error in one single coefficient will spread to the following coefficients. In view of probable channel error performance and coding efficiency, block classification coding seems to be the best choice.

1.3.3 Schemes exploiting interblock redundancy

In all of the foregoing transform coding techniques, it is assumed that successive blocks of data are independent. Indeed, if the block size is large, for example 32 or larger, the interblock correlation is negligible. However, transform coding using a large block size, although it can achieve a greater reduction in bit rate, suffers from two distinct disadvantages.

1) It requires more computation time, more complex implementation and storage of large amounts of data both at the transmitter and the receiver, and consequently produces a delay in transmission.

2) Image statistics may vary widely within a block if the block size is large. Adaptive coding to match statistics within a block is then difficult to accomplish.

These drawbacks can be solved by choosing a small block size and then applying coding schemes which exploit interblock redundancy. One natural way is to apply predictive coding to exploit the redundancy between the transform coefficients of different blocks. This type of scheme, comprised of transform coding and predictive coding, is called hybrid coding. Fig.1.7 shows a block diagram of a hybrid coder.

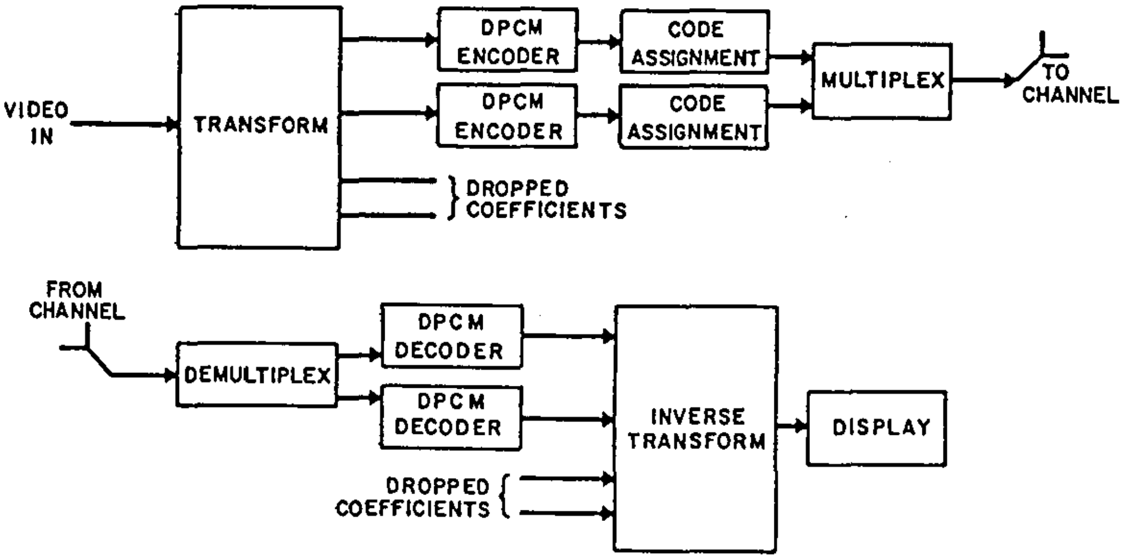


Fig.1.7 Block diagram of a hybrid coder.

Specifically, three types of schemes have been examined, 1) a one-dimensional block along a horizontal line with DPCM in the vertical direction; 2) a small two-dimensional block, and DPCM using coefficients of the previous horizontal block for prediction; 3) a two--dimensional block, and DPCM in the temporal direction. Variations of these three systems have been examined by many researchers [58-62]. Habibi showed both theoretically and experimentally that the performance of hybrid coding systems (1) and (2) surpasses that of both DPCM and a non-adaptive two-dimensional Hadamard transform coding system. Also, performance of the hybrid coding systems was found to be reasonably independent of the block size, the performance improving only very slightly for block sizes larger than eight.

Netravali et al [59], using a small two-dimensional block, showed that if the optimum transform (KLT) is not used, correlation exists between the coefficients of a given block.This implies that a better predictor can be designed by using not just the corresponding coefficient of the previous block, but also all other coefficients of the previous block, as well as those of the present one. Such a predictor was shown to be 25% more efficient in terms of data rate, for the same picture quality, than a predictor which based its prediction only on the corresponding coefficient of the previous block. Comparison between a hybrid coding system (3) and transform coding using three-dimensional blocks was carried out by Roese et al [63]. Both experimental and theoretical results indicated that the simpler hybrid coding system performs as well as the three-dimensional transform coder.

Another technique to exploit interblock redundancy is recursive block coding (RBC) [14] which encodes (n+p) samples at a time when a n x n transform is used. The theoretical background of RBC is given in section 2.4.2.2. For example for p=1, the coding algorithm for the block of data [x_0 ,x_1 ,...,x_{n+1}] proceeds as follows (Fig.1.8):

(i) Boundary point x_{n+1} is encoded, transmitted and stored at the receiver for the present as well as the next block.

(ii) At both the transmitter and the receiver, quantized boundary points xq_{-1} and xq_{0} xq_{n+1} are passed through a noncausal FIR filter α [Q] and a quantizer to produce a quantized n-vector XQ_b , called the boundary response.

(iii) A residual process $\tilde{X}^o_b$ is obtained by subtracting the quantized boundary response XQ from the original data X. This is then encoded and transmitted using the sine transform, which was found to be the KL transform of vector X^o if vector X is a sample of first-order Markov process [31].

Using a non-adaptive zonal coding technique, comparison between the sine transform with RBC and the DCT in both one and two-dimensions [59] has shown that RBC results in a smaller mean square distortion. In addition, recursive block coding, while producing sharper images, suppresses the objectionable block-boundaries which exist in the DCT coded picture. Meiri and Yudilevich have also developed a very similar algorithm called the pinned sine transform [65].

1.3.4 Psychovisual coding

Apart from the effort to match local statistics of inhomogeneous

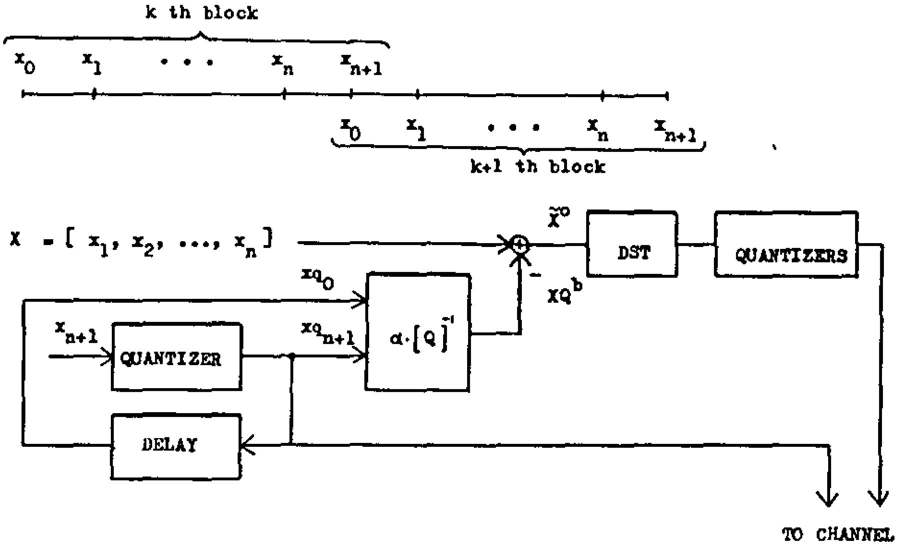


Fig.1.8 Block diagram of a recursive block coding system.

pictorial data with various adaptive schemes, human psychovisual considerations have also been used to reduce the bit rate. Much effort has been made to model the human visual response as a linear system [68-74]. This is probably because of the completeness and simplicity of linear systems theory. However, the work of Stockham [73] (1972) has shown that the human visual system is nonlinear and also rotationally variant. Mannos and Sakrison suggested that [75], after an initial nonlinear transformation, the remainder of the human visual response may be considered linear over a moderate range of light intensities. They therefore proposed a model for human psychovisual system consisting of a cascaded nonlinear and linear system, which will be described later in this section.

For pel-domain waveform coding, such as predictive coding, knowledge about the tolerable error at each pel and how the errors at adjacent points combine is very useful. The error ΔL, known as the visual threshold, is defined as the point at which a perturbation or distortion just becomes visible or ceases to be visible. It was found that ΔL depends on the following factors [66]:

1) L_S , the overall luminance of the surroundings,
2) L_B , the background luminance adjacent to the perturbation,
3) the presence of sharp luminance changes adjacent to the perturbation.

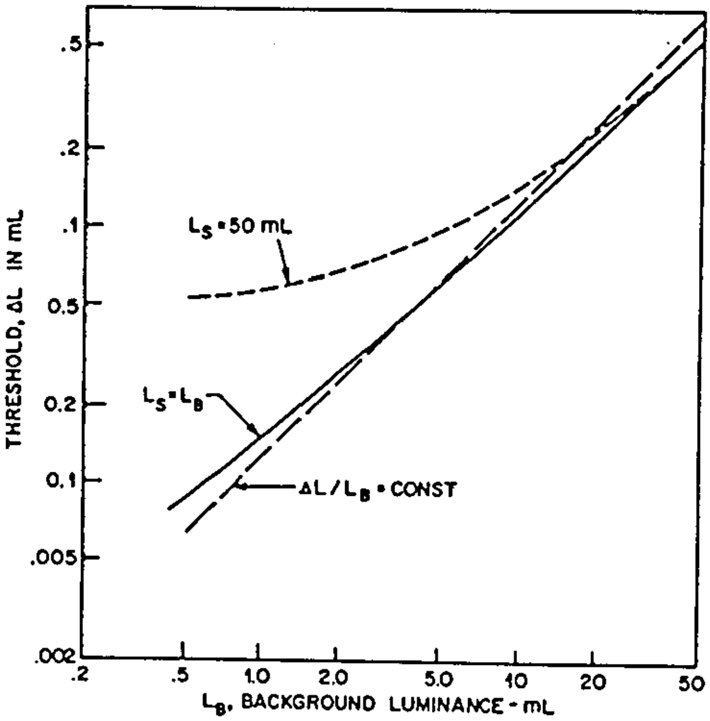


Fig.1.9 The relation between visual threshold (ΔL), background luminance (L_B) and surrounding luminance (L_S).

The relation between ΔL, L_S and L_B is shown in Fig.1.9. The long-dashed line represents the condition $\Delta L/L_B$ =constant (Weber's Law) when $L_S = L_B$. However, when L_S is much larger than L_B , Weber's

Law is nò lònger valid. The actual relation between ΔL and L_B is that given by the short-dashed line. If there are large changes in luminance adjacent to the perturbation, ΔL increases on both the dark side and the bright side of the luminance change. Details of visual threshold properties can be found in [66,67].

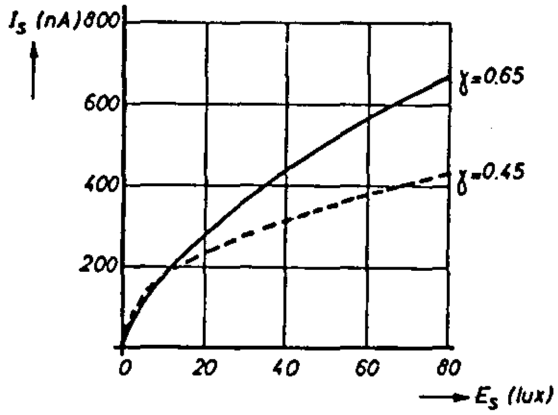


Fig.1.10 The relation between signal current (I_S) and illumination(E_S) for a camera tube with a γ of 0.65 and 0.45 respectively.

Whilst a human observer is more sensitive to a perturbation in a dark area than in a bright area, a video camera also has similar characteristics. Fig.1.10 shows the relation between the illumination (E_S) and the corresponding signal current (I_S) in the camera, which is

$$I_S = E_S^{\gamma}$$

The constant gamma (γ) is a parameter of the camera tube and is about 0.65 for most vidicons. However, a picture tube has also a curved characteristic with a γ of 2.2. To reach the ideal situation of

$$\gamma_{\text{vid.}} \times \gamma_{\text{picture tube}} = 1 \qquad \text{------------------(1.4)}$$

gamma correction is usually provided in the camera using a circuit to reduce $\gamma_{vid.}$ from 0.65 to 0.45 to satisfy equation 1.4. Therefore, the signal waveforms from most video cameras are already companded with γ equal to 0.45. Uniform quantization of these companded signals means that more bits are allocated to signals representing dark areas than those representing bright areas. The fact that the human visual response is more sensitive in dark areas than in bright areas is also reflected in the nonlinear part of the human visual system model proposed by Mannos and Sakrison, which is a nonlinear transformation $f(u)=u^{0.33}$.

For transform coding, a knowledge of H(w), the sensitivity or spatial frequency response of the human visual response is more useful. The linear part of the model proposed by Mannos and Sakrison [75] is a filter transform function which indicates the relative sensitivity of the human visual system H(w) to spatial light intensity distribution (w) as follows:

$$H(w) = 2.6 [0.0192 + 0.114 w] \exp\{ -(0.114 w)^{1.1} \} \qquad \text{-----------------------------(1.5)}$$

As depicted in Fig.1.11, H(w) has a maximum at w=8.0 cycle/degree with a rapid decrease on either side.

Hall [76] has claimed that the incorporation of a human visual system model in a Fourier transform coding scheme can improve the compression of still pictures by a factor of almost 10. He subsequently extended his techniques to code color signals with good results at 1.0 bit/pel and with acceptable quality down to 0.25 bit/pel. Ngan, in his

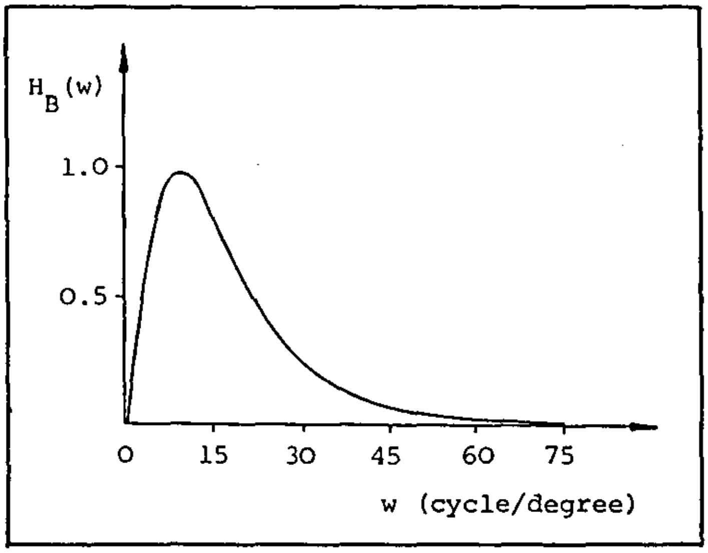


Fig.1.11 The Mannos and Sakrison human visual system model (after ref. [68]).

comparison of five adaptive schemes, also found that the scheme which has a bit assignment based on the human visual model given by equation 1.5 gave the best performance in terms of subjective quality [96].

1.4 ORGANISATION OF THE THESIS

Following this introductory chapter, chapter two provides the basic theory that will be used throughout the thesis, and begins with the abstract concepts of fields, vector spaces and bases, followed by transform coding theory, and finally the optimization of parameters in a practical transform coding system. In chapter three, attention is concentrated on the Walsh transform. A unified matrix treatment for the Walsh matrix using the concept of dyadic symmetry is presented. This unified treatment allows a straightforward derivation of a simple equation for the generation of Walsh transforms of different orderings, various re-ordering schemes and various fast computational algorithms.

It is believed that this will provide a better understanding of the Walsh transform, and hence, allow further fast computational algorithms and new properties to be found. The whole theory relates to a binary field with 'logical and' and modulus two addition as operations and thus allows both the generation of Walsh matrices of different orderings, and re-ordering schemes, to be implemented using simple logic circuits.

The simplicity and ease of implementation of the Walsh transform have resulted in a wide range of applications. However, the performance of the Walsh transform is inferior to that of the more complicated examples such as the discrete Fourier and cosine transforms. Chapter four demonstrates the use of the theory of dyadic symmetry to generate two new transforms which can be used as substitutes for the Walsh transform. The new transforms have virtually the same complexity and computational requirements as the Walsh transform, employing additions, subtractions and binary shifts only but with an improved efficiency, defined in terms of ability to decorrelate signal elements, which lies between that of the Walsh transform and that of the discrete cosine transform.

A conventional transform coder employing an efficient transform exploits largely, if not entirely, the redundancy between pels within the same block. However, the correlation between pels in different blocks is completely neglected. Chapter five describes a technique that utilizes this interblock redundancy to allow the dc coefficients to be estimated at the receiver, thus allowing reductions in bit rate as well as eliminating a major source of difficulty with respect to channel errors. Three schemes called ELEMENT ESTIMATION, ROW ESTIMATION, and PLANE ESTIMATION are proposed. Results of simulations of these methods using different block sizes and different pictures are shown.

The thesis concludes with chapter six which collates the discoveries and work that has been performed in the course of the research programme, and makes suggestions for future work.

CHAPTER TWO

TRANSFORM CODING THEORY

2.1 INTRODUCTION

This thesis essentially contains three new contributions to transform processing: a unified treatment of the Walsh matrix using dyadic symmetry (chapter three), new simple and efficient transforms (chapter four) and dc coefficient restoration schemes (chapter five). These discoveries, all relating to transform coding theory, were developed using the theory of vectors and matrices, ranging from the abstract concept of fields to matrix algebra.

This chapter links all the theory together, establishing a common background of notation and terminology that will be used throughout the thesis. To do this, in the next section a brief review of groups and fields is first given, which is then followed by definitions of vectors and linear independence of vectors, and finally of vector spaces and bases. The unified matrix treatment of the Walsh matrix was essentially developed using the theory discussed in this section.

In section 2.3, attention is concentrated on vectors in vector spaces over a number field, within which the basic concept of image transform coding theory is established. Section 2.4 then describes the well known orthogonal transformations. The last section discusses the optimization of parameters in a transform coding system, including the choice of transformation, transform block size and quantization strategy. The matrix algebra required for the dc coefficient restoration schemes will be presented in chapter five and in the appendices and so is not included in this chapter.

2.2 VECTORS IN A VECTOR SPACE

2.2.1 Groups and fields

We are very familiar with the arithmetic of addition and multiplication of a set of real numbers. A modern mathematical point of view sees this as a special case of a large class of more general relationships. These general relationships are considered in an abstract way to save the trouble of proving analogous theorems in each case, as well as to obtain a better understanding of those relationships. Groups and fields, for example, are two of these abstract relationships [78-80].

Definition 2.1:

A group is a set $\{ a,b,c,\dots \}$ and an operation + which has the following properties:

1) Closure : $a+b$ is also a legitimate element of the set.

2) Commutative law : $a+b = b+a$

3) Associative law : $(a+b)+c = a+(b+c)$

4) Identity : there is some element, denoted 0, such that for any element a, $0+a=a$.

5) Inverse : corresponding to every element, a, there is another element, $-a$, such that $a+(-a)=0$.

The variety of sets and relationships which are groups is very large. A special class of groups is called a field.

Definition 2.2:

A field is a set $\{ a,b,c,... \}$ and two different operations + and * , satisfying the following rules:

1) Closure : $a+b$ as well as $a*b$ are valid elements of the set.

2) Commutative law : $a+b = b+a$
$a*b = b*a$

3) Associative law : $(a+b)+c = a+(b+c)$
$(a*b)*c = a*(b*c)$

4) Identity : There is some element, denoted 0, such that $0+a=a$.
There is some element, denoted 1, such that $1*a=a$.

5) Inverse : For any element, a, there is an element $-a$ such that $a+(-a)=0$. For any element, a, except 0, there is an element, a^{-1}, such that $a*a^{-1}=1$.

6) Distributive law: $a*(b+c) = (a*b)+(a*c)$

Property 5 allows us to define inverse operations for + and *, which are denoted as - and / respectively, using the following equations.

$$a-b = a+(-b) \quad \text{------------------------------------}(2.1)$$

$$a/b = a*(b^{-1}) \quad \text{------------------------------------}(2.2)$$

For example, the set $\{ 1,0 \}$ with operations "exclusive or" $\{ (+) \}$ and "logical and" $\{ * \}$ is a field because it satisfies all the six rules mentioned above. In chapters three and four, vectors over this field are used to represent a quantity called dyadic symmetry. Properties of dyadic symmetry are then derived from the well known properties of vectors over a field.

However, the fields that we are most familiar with may be number fields. A set of complex numbers, consisting of more than the element 0, is called a number field if the operations of addition { + } and multiplication { x } on any two of the numbers yield a number of the set. Examples of number fields are a) the set of all rational numbers, b) the set of all real numbers, c) the set of all complex numbers.

Vectors over the field of real numbers will be used to represent a block of pictorial data (in the spatial or transform domains). Properties of such blocks of data can be derived from the well known properties of vectors over a field.

2.2.2 Vectors over a field [81-83]

A point X in a plane can be denoted by an ordered pair of real numbers. This point X or (x_1 , x_2) can be represented as a two-dimensional vector or 2-vector and written as $[x_1 , x_2]$. In general, an n-vector over F can be defined by definition 2.3. (In this section, F is used to denote a general field with operations * and +)

Definition 2.3:

An n-dimensional vector or n-vector X over F is an ordered set of n elements x_i of F , thus

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \cdot\cdot \\ \cdot\cdot \\ x_n \end{bmatrix} \quad \text{---------------------}(2.3)$$

The elements x_1, x_2, ..., x_n are called respectively the first, second, ... nth components of X.

Definitions of addition, subtraction, scalar multiplication and dot product of vectors, as well as definitions of matrix and matrix algebra can be found in every text book on vectors and matrices, and will not be repeated here. Attention will be concentrated on the concept of linear dependence of vectors which is essential for the derivation of the concept of dyadic symmetry as well as of the two new transforms. For convenience, vectors are column vectors as in equation 2.3, unless specified otherwise.

Definition 2.4:

The m n-vectors over F

$$\begin{aligned} X_1 &= [x_{11} , x_{12} , ..., x_{1n}]^t \\ X_2 &= [x_{21} , x_{22} , ..., x_{2n}]^t \\ \cdot\cdot \quad & \cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot \\ X_m &= [x_{m1} , x_{m2} , ..., x_{mn}]^t \quad \text{------------------}(2.4) \end{aligned}$$

are said to be linearly dependent over F if there exist m elements k_1, k_2, ..., k_m of F, not all zero, such that

$$k_1 {*} X_1 + k_2 {*} X_2 + ... + k_m {*} X_m = 0 \quad \text{---------------}(2.5)$$

Otherwise, the m vectors are said to be linearly independent. If in equation 2.5, $k_i \neq 0$, we may solve for

$$X_i = - \{ k_1 * X_1 + \dots + k_{i-1} * X_{i-1} + k_{i+1} * X_{i+1} + \dots + k_m * X_m \} / k_i \qquad \text{-----------------------(2.6)}$$

Therefore, the following properties exist

(a) If r vectors are dependent, any of them may always be expressed as a linear combination of the others.

(b) If r vectors are independent then none of them may be expressed as a linear combination of the others.

(c) If r vectors are independent while the set obtained by adding another vector X_{r+1} is dependent, then X_{r+1} can be expressed as a linear combination of $X_1, X_2, \dots\dots, X_r$

Furthermore, the following well known properties will be stated without proof.

(d) If among the m vectors $X_1, X_2, \dots, X_m$ (equation 2.4), there is a subset of $r < m$ vectors which are linearly dependent, the vectors of the entire set are linearly dependent.

(e) If the set of vectors (equation 2.4) is linearly independent so also is every subset of them.

(f) A necessary and sufficient condition that the vectors (equation 2.4) be linearly dependent is that the matrix

$$[\, X \,] = \begin{bmatrix} x_{11} & \dots\dots & x_{1n} \\ \cdot\cdot & \dots\dots & \dots \\ \cdot\cdot & \dots\dots & \dots \\ x_{m1} & \dots\dots & x_{mn} \end{bmatrix} \quad m \leqslant n \qquad \text{-----------(2.7)}$$

of the vectors (equation 2.4) be of rank $r < m$. If the rank is m, the vectors are linearly independent.

If m is greater than n, the m vectors (equation 2.4) must be linearly dependent as the rank r of the matrix (equation 2.7) must be less than m. In other words, we have property (g).

(g) There are at most n linearly independent n-vectors.

An $n \times n$ matrix has an inverse if its rank r equals n. Property (f) with $m=n$ can therefore be rewritten as

(h) The matrix (equation 2.7) has an inverse if and only if the vectors (equation 2.4) are linearly independent.

2.2.3 Vector spaces and vector bases

Definition 2.5:

Any set of n-vectors over F which is closed under both addition and scalar multiplication is called a vector space.

Therefore, if $X_1, X_2, \ldots, X_m$ are n-vectors over F, the set of all linear combinations

$$k_1{*}X_1 + k_2{*}X_2 + \;\ldots.. + k_m{*}X_m \qquad k_i \in F \qquad \text{-------}(\ 2.8\ )$$

is a vector space over F.

Definition 2.6:

By the dimension of a vector space V is meant the maximum number of linearly independent vectors in V or, (what amounts to the same thing), the minimum number of linearly independent vectors required to span V.

A vector space of dimension r consisting of n-vectors over F will be denoted by $V_n^r(F)$. When $r=n$, $V_n(F)$ will be used in place of $V_n^n(F)$ for simplicity.

Definition 2.7:

A set of r linearly independent vectors of $V_n^r(F)$ is called a basis of the space.

Any r linearly independent vectors of a space will serve as a basis and each vector of the space is a unique linear combination of the basis vectors of $V_n^r(F)$.

The n n-vectors

$$\begin{aligned} E_1 &= [\,1, 0, \ldots, 0, 0\,]^t \\ E_2 &= [\,0, 1, \ldots, 0, 0\,]^t \\ \cdot\cdot &\quad \cdots\cdots\cdots\cdots \\ E_n &= [\,0, 0, \ldots, 0, 1\,]^t \text{---------------------(2.9)} \end{aligned}$$

are called elementary or unit vectors over F. The unit vectors E_1, E_2, ..., E_n constitute an important basis, known as the unit basis, for $V_n(F)$. Every vector $X = [\,x_1, x_2, \ldots, x_n\,]^t$ of $V_n(F)$ can be expressed as

$$X = x_1E_1 + x_2E_2 + \ldots. + x_nE_n \qquad \text{---------(2.10)}$$

The components $x_1, x_2, \ldots, x_n$ of X are now called the coordinates of X relative to the unit basis. Unless stated otherwise, a vector is always given relative to the unit basis.

Let $T_1, T_2, \ldots, T_n$ be the basis vectors of another basis of $V_n(F)$ and

$$X = c_1 T_1 + c_2 T_2 + \ldots\ldots + c_n T_n \qquad \text{---------(2.11)}$$

Then the scalars $c_1, c_2, \ldots., c_n$ are called the coordinates of X relative to the T-basis and are represented by vector C :

$$C = [\ c_1, c_2, \ldots, c_n\ ]^t \qquad \text{------------------(2.12)}$$

Equation 2.11 now can be written as

$$X = [\ T_1, T_2, \ldots, T_n\ ]\ \ C$$

or in a more concise form

$$X = [\ T\ ]^t C \qquad \text{----------------------------(2.13)}$$

$$\text{where} \qquad [\ T\ ] = \begin{bmatrix} T_1^t \\ T_2^t \\ \cdot\cdot \\ T_n^t \end{bmatrix} \text{----------------------------(2.14)}$$

Let $W_1, W_2, \ldots, W_n$ be yet another basis of $V_n(F)$, and the coordinates of X relative to the W-basis be represented by the vector

$$X_W = [\ w_1, w_2, \ldots, w_n\ ]^t \text{-------------------(2.15)}$$

Therefore, we have

$$X = [\ W\ ]^t X_W \qquad \text{---------------------------(2.16)}$$

Equations 2.13 and 2.16 imply

$$[\ T\ ]^t\ C \ = \ [\ W\ ]^t\ X_W$$

or

$$C \ = \ \left([\ T\ ]^t\right)^{-1}[\ W\ ]^t\ X_W \text{ -----------------------(2.17)}$$

2.3 LINEAR ORTHOGONAL TRANSFORMS

2.3.1 One-dimensional linear orthogonal transforms

Consider a block of n pels or a subpicture. If we represent it as a vector X in a vector space $V_n(F)$, then the vector space $V_n(F)$ contains all the possible subpictures. From now on, unless specified otherwise, F refers to the field of real numbers.

In transform coding, vector X will be transformed into vector C of n coefficients by a transform [T] at the transmitter

$$C \ = \ [\ T\ ]\ X \text{ -------------------------(2.18)}$$

then each of the coefficients will be separately coded and sent through the channel. At the receiver, X is obtained by taking the inverse transform of C.

$$X \ = \ [\ T\ ]^{-1}\ C \text{ ---------------------(2.19)}$$

The transformations between X and C in equations 2.18 and 2.19 are in fact simply changes of coordinates between the unit basis and basis [T] as given by equation 2.14. The elements of the vector X are the coordinates of the subpicture with respect to the unit basis, whilst the coefficients in vector C are the coordinates with respect to the basis [T].

If the basis is orthogonal, then we have

$$[\,T\,]\,[\,T\,]^{t} = [\,I\,]$$

$$\text{or} \qquad [\,T\,]^{t} = [\,T\,]^{-1} \qquad \text{-------------------(2.20)}$$

where [I] is an identity matrix. In this case, the basis vectors of [T] are orthonormal to each other. That means

$$T_i^{t}\, T_j = 0 \qquad i \neq j$$

$$= 1 \qquad i = j \qquad \text{-----------------(2.21)}$$

Also, the energy of the transform coefficients and of the pels is the same.

Fig.2.1 gives an example showing how a vector can be represented with respect to the unit basis and to another basis. Equation 2.18 indicates that the ith coefficient c_i is the scalar product of the ith basis vector T_i and the signal vector X.

$$c_i = T_i^{t}\, X \qquad \text{----------------------------(2.22)}$$

Equations2.19 and 2.20 imply that the signal vector X equals the summation of the basis vectors weighted by the coefficients.

$$X = \sum_{i=1}^{n} c_i\, T_i \qquad \text{--------------------(2.23)}$$

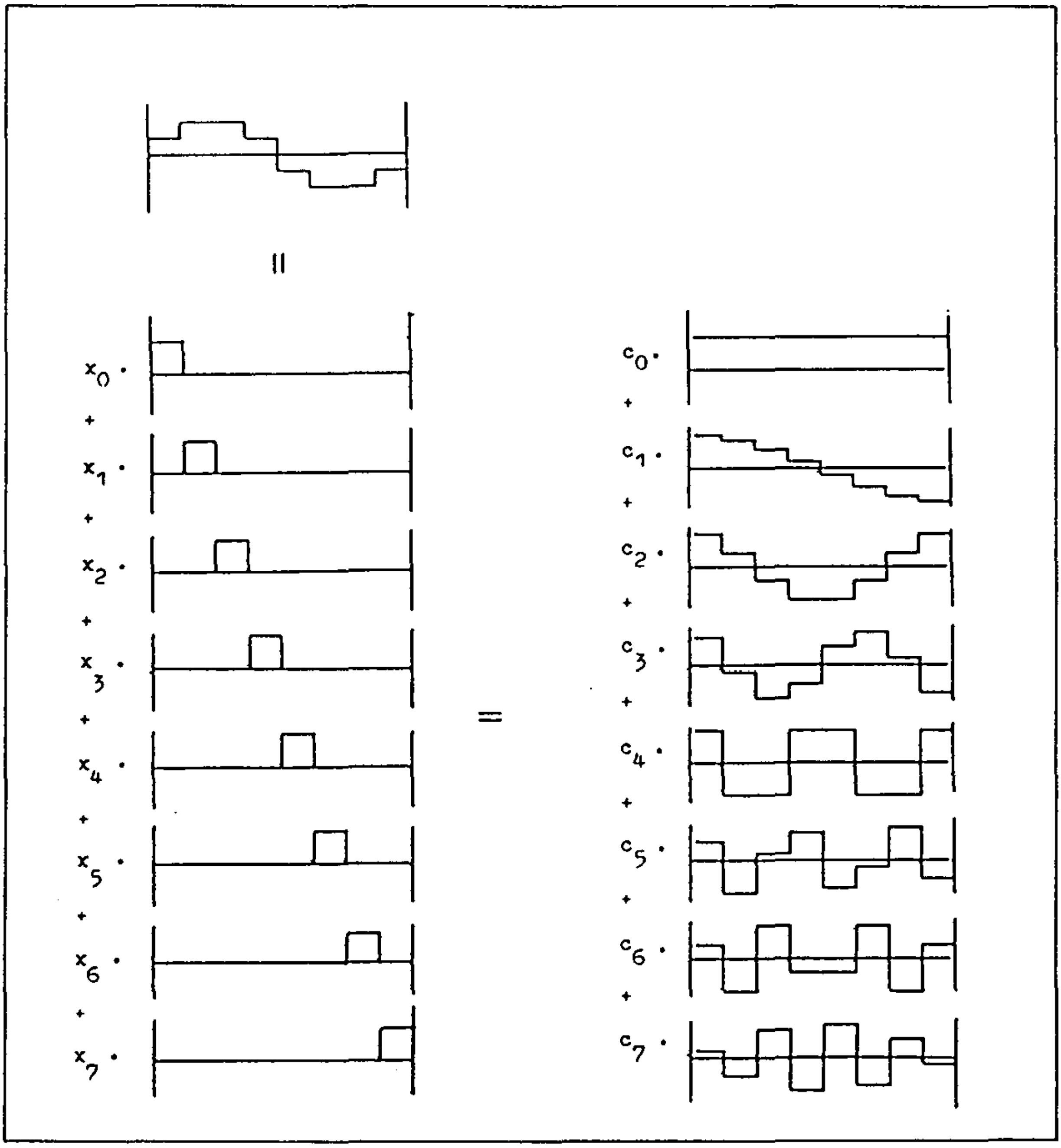


Fig.2.1

Representation of a 8-vector with respect to the unit basis with coordinates $(x_0, x_1, \ldots, x_7)$ and another basis with coordinates $(c_0, c_1, \ldots, c_7)$.

In transform coding, T_i is often interpreted as a spectral function, and the coefficient c_i is the corresponding spectral component indicating the amount of energy of the spectral function T_i in the subpicture. In this case, i runs from 0 to n-1, and relates to sequency (generalised frequency). Harmuth [100] defined "sequency" for any type of function as one-half the average number of zero-crossings per unit time. The definition of sequency coincides with that of frequency when applied to sinusoidal functions. Later, Yuen [112] proposed the term "zequency" to denote the number of zero-crossings of Walsh functions.

2.3.2 Two-dimensional linear orthogonal transforms

In this section, only separable two-dimensional orthogonal transforms are described. A more general approach to two-dimensional transforms is given by Pratt [84], chapter ten. Consider a block of n x n pels

$$[\,X\,] = [\,X_1, X_2, \ldots, X_n\,]$$

where the column vector X_i represents the ith column of the matrix $[\,X\,]$. A seperable two-dimensional transform can be performed on $[\,X\,]$ in two steps:

(1) $[\,K\,] = [\,T\,][\,X\,]$

$[\,X\,]$ is first transformed into $[\,K\,]$ by a pre-multiplication of $[\,X\,]$ by $[\,T\,]$. This in fact converts every X_i (a column vector of $[\,X\,]$) into κ_i, a column vector of $[\,K\,]$.

$$\kappa_i = [\,T\,]\,X_i \qquad \text{-----------------------------}(\,2.24\,)$$

2) $[\,C\,] = [\,K\,][\,T\,]^t$

$[\,K\,]$ is then transformed into $[\,C\,]$ by a post-multiplication of $[\,K\,]$ by $[\,T\,]^t$. This is to convert every K_j^t (a row vector of $[\,K\,]$) into C_j^t, a row vector of $[\,C\,]$.

$$C_j^t = K_j^t\,[\,T\,] \text{ ------------------------------------}(\ 2.25\ )$$

Equations 2.24 and 2.25 give

$$[\,C\,] = [\,T\,][\,X\,][\,T\,]^t \text{ ----------------------------}(\ 2.26\ )$$

Hence (using equation 2.20), the inverse two-dimensional transform is

$$[\,X\,] = [\,T\,]^t[\,C\,][\,T\,] \text{ ----------------------------}(\ 2.27\ )$$

Let T_i and T_j be the i th and j th row vectors of $[T]$. Equation 2.26 indicates that the (i,j)th coefficient is

$$c_{ij} = T_i^t\,[\,X\,]\,T_j \text{ ---------------------------------}(\ 2.28\ )$$

and c_{ij} can thus be viewed as the scalar product of $[\,X\,]$ and the basis picture $[\,T_i\,T_j^t\,]$. Equation 2.27 implies that the data matrix $[\,X\,]$ equals the summation of the basis pictures $T_i\,T_j^t$ weighted by the coefficients c_{ij}.

$$[\,X\,] = \sum_{i=1}^{n}\ \sum_{j=1}^{n}\ c_{ij}\,T_i\,T_j^t \text{ -------------------------}(\ 2.29\ )$$

Figs.2.2 and 2.3 show the basis pictures of the 4x4 Walsh transform and the unit basis. Comparisons between equations 2.28 and 2.29,

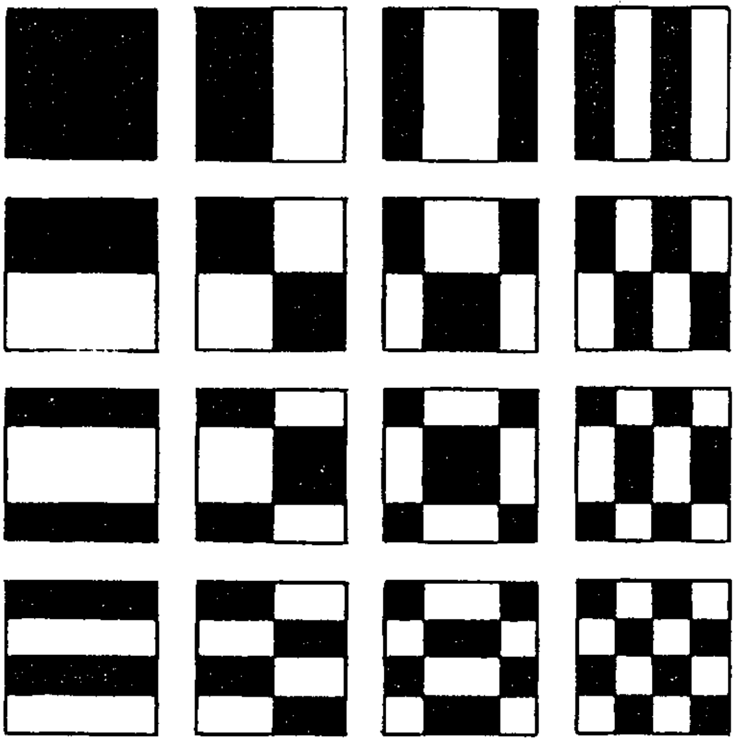


Fig. 2.2

The Walsh transform basis-pictures for n=4. Black=+1 ; white=-1.

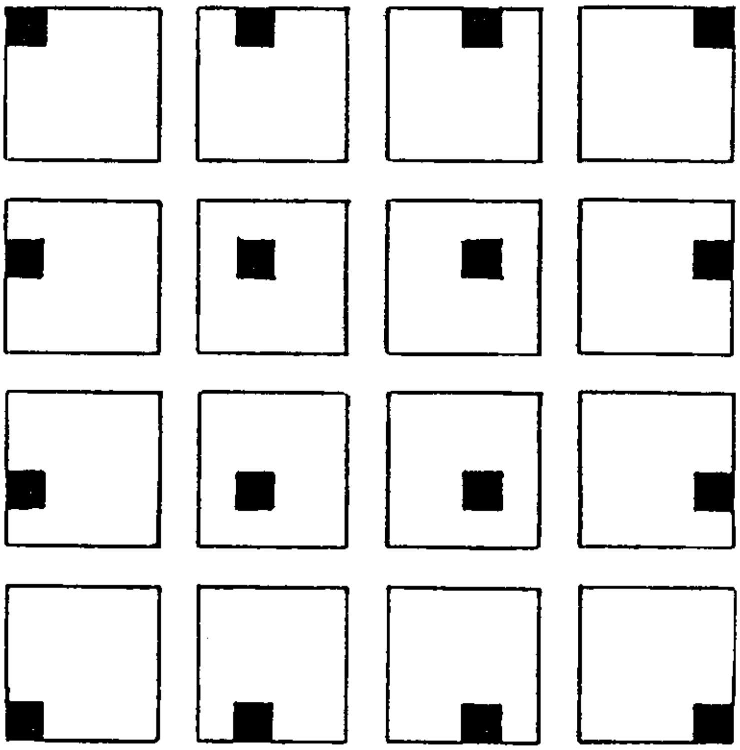


Fig. 2.3

The unit basis basis-pictures for n=4. Black=+1 ; white=-1.

2.22 and 2.23 show that the one- and two-dimensional transforms are in fact very similar.

2.4 TRANSFORMATIONS

2.4.1 The optimum transform

Consider a picture which is divided into N n-vectors, X_i with mean vector $\overline{X}$. With a transformation [T], each $X_i - \overline{X}$ is transformed into C_i. Define the covariance matrix of X_i as

$$[\ CX\ ] = \frac{1}{N} \sum_{i=1}^{N} (X_i - \overline{X})\,(X_i - \overline{X})^t \quad \text{-----------}(\ 2.30\ )$$

The covariance matrix [CC] of C_i can be expressed in terms of [CX] and [T].

$$\begin{aligned} [\ CC\ ] &= \frac{1}{N} \sum_{i=1}^{N} C_i\, C_i^t \\ &= \frac{1}{N} \sum_{i=1}^{N} [T]\,(X_i - \overline{X})\ (\ [T]\,(X_i - \overline{X})\ )^t \\ &= \frac{1}{N} \sum_{i=1}^{N} [T]\,(X_i - \overline{X})\ \ (X_i - \overline{X})^t\,[T]^t \end{aligned}$$

Therefore, we have

$$[\ CC\ ] = [\ T\ ]\,[\ CX\ ]\,[\ T\ ]^t \quad \text{--------------------}(\ 2.31\ )$$

The (i,j)th element of [CC], s_{ij}, is the covariance between c_i and c_j.

A transformation is optimum if it can transform n pels into completely uncorrelated coefficients. In other words, we have the following definition for the optimum transformation:

Definition 2.8:

The optimum transformation of a picture of covariance matrix [CX] is the one whose covariance matrix of the transform coefficients [CC] is a diagonal matrix.

The above definition implies that the basis vectors (row vectors) of the optimum transform are the eigenvectors of [CX]. Furthermore, the optimum transform under definition 2.8 results in the least normalized mean square error (equation 4.6) and has the best energy packing ability (equation 4.5) [109]. Since different images have different covariance matrices [CX], there is no single unique optimum transform. Much of the effort in studying the optimum transform has been directed at that of the first-order zero-mean, unit-variance Markov process whose covariance matrix is given by the Toeplitz matrix

$$[\ CX\ ] = \left[\begin{array}{lllll} 1 & \rho & \cdots\cdots\cdots\cdots & & \rho^{n-1} \\ \rho & 1 & \rho & \cdots\cdots\cdots\cdots & \\ \rho^{2} & \rho & 1 & \rho\ \cdots\cdots\cdots & \\ \cdots\cdots & \rho & 1 & \rho\ \cdots\cdots & \\ \cdots\cdots\cdots\cdots & & & & \\ \cdots\cdots\cdots\cdots & & & & \\ \rho^{n-1} & \cdots\cdots\cdots\cdots & & \rho & 1 \end{array} \right] \quad \text{----(2.32)}$$

where ρ is the adjacent element correlation coefficient. Although the eigenvectors of [CX] are known analytically [85], there is no known fast algorithm to transform a vector of data. For high order Markov processes,closed-form solutions are still not known in general, and the possibility of fast algorithms seems even more remote.

2.4.2 Suboptimum transformations

2.4.2.1 Sinusoidal transforms [86,87]

Consider the parametric family of matrices

$$J(k_1,k_2,k_3,k_4) = \begin{bmatrix} 1-k_1 a & -a & & & & k_3 a \\ -a & 1 & -a & & & \\ & -a & 1 & -a & & \\ & & \cdot\cdot & \cdot\cdot & \cdot\cdot & \\ & & & \cdot\cdot & \cdot\cdot & \cdot\cdot \\ & & & -a & 1 & -a \\ k_4 a & & & & -a & 1-k_2 a \end{bmatrix} \quad \text{--(2.33)}$$

If we define
$$a = \rho/(1+\rho^2)$$
$$\beta^2 = (1-\rho^2)/(1+\rho^2) \quad \text{-------------------(2.34)}$$

then we have

$$\beta^2 \; [J(\rho,\rho,0,0)]^{-1} = [\ CX\ ]$$

which is the covariance matrix in equation 2.32 of the stationary, first-order Markov process. Since the eigenvectors of a matrix and eigenvectors of its inverse are identical, the sparse matrix [$J(p,p,0,0)$] in equation 2.33 can be used to compute the optimum transform or, what is the same thing, the set of eigenvectors. Similarly, the sinusoidal transform family is defined on the sparse matrix [$J(k_1,k_2,k_3,k_4)$].

Definition 2.9:

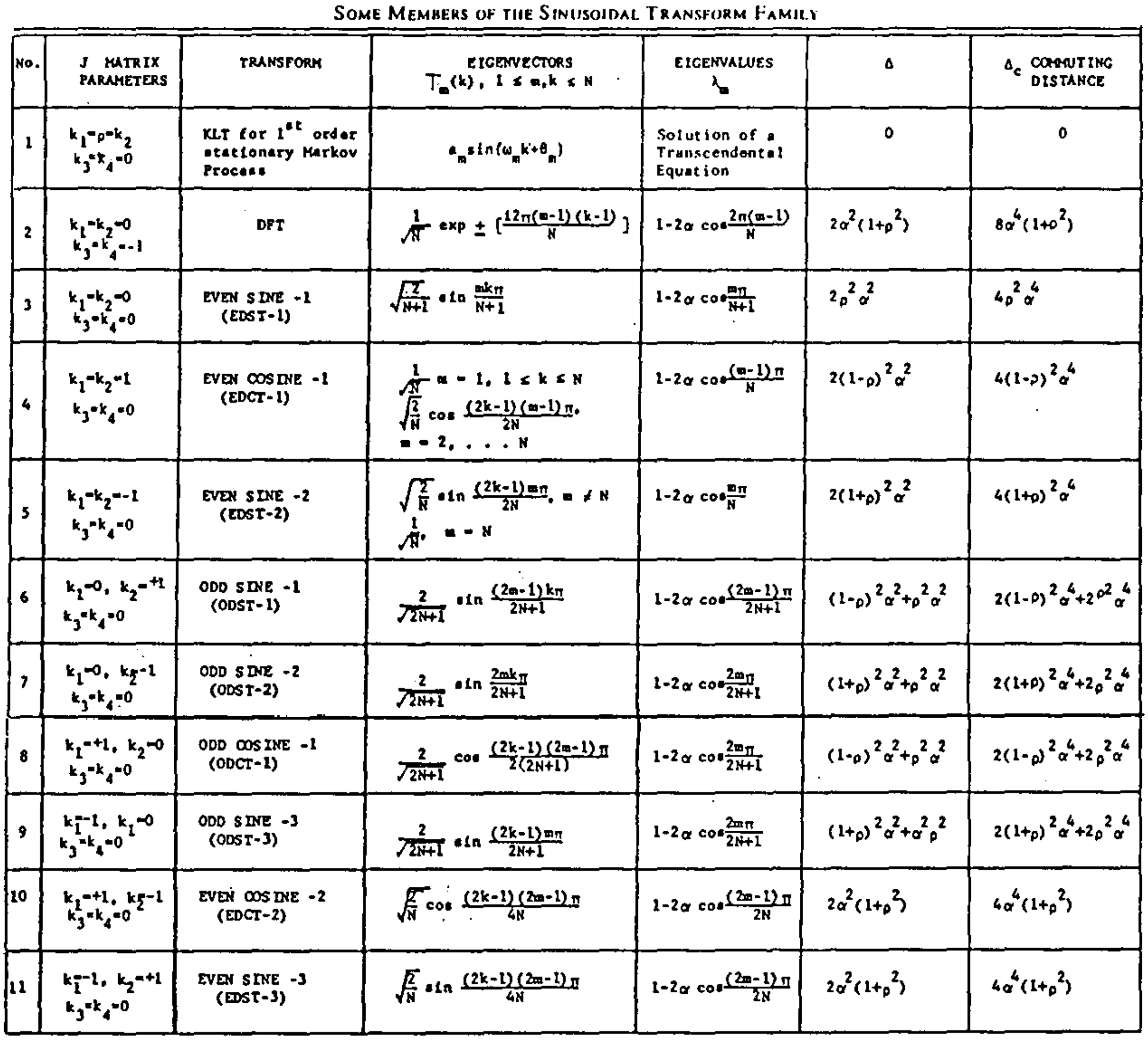
The sinusoidal family of unitary transforms is the class of orthonormal sets of eigenvectors T_m generated by the sparse matrces [$J(k_1,k_2,k_3,k_4)$] for those values of k_1, k_2, $k_3=k_4$ and a such that the matrix is positive definite.

In the other words, a sinusoidal transform [T] is one that satisfies equation 2.35 where [D] is a diagonal matrix.

$$[\,D\,] = [\,T\,]\ [\,J(k_1,k_2,k_3,k_4)\,]\ [\,T\,]^t \quad \text{----(2.35)}$$

Table 2.1 summarizes some of the sinusoidal transforms. In this table, T_m, the m th row vector of [T], represents the mth eigenvector.

In image processing, two of the most important sinusoidal transforms are the EDST-1 (transform 3) and EDCT-1 (transform 4) which are commonly known as the DST and DCT respectively. Figs 2.4 and 2.5 show the basis functions of the 16x16 DCT and DST. The DCT is asymptotically close to the optimum transform of the first-order Markov process whose covariance matrix is the Toeplitz matrix (equation 2.32), and at present is regarded as the best suboptimum transform in conven-



SOME MEMBERS OF THE SINUSOIDAL TRANSFORM FAMILY

No.	J MATRIX PARAMETERS	TRANSFORM	EIGENVECTORS $T_m(k)$, $1 \le m,k \le N$	EIGENVALUES λ_m	Δ	Δ_c COMMUTING DISTANCE
1	$k_1=\rho=k_2$ $k_3=k_4=0$	KLT for 1^{st} order stationary Markov Process	$a_m \sin(\omega_m k+\theta_m)$	Solution of a Transcendental Equation	0	0
2	$k_1=k_2=0$ $k_3=k_4=-1$	DFT	$\frac{1}{\sqrt{N}} \exp \pm [\frac{12\pi(m-1)(k-1)}{N}]$	$1-2\alpha \cos\frac{2\pi(m-1)}{N}$	$2\alpha^2(1+\rho^2)$	$8\alpha^4(1+\rho^2)$
3	$k_1=k_2=0$ $k_3=k_4=0$	EVEN SINE -1 (EDST-1)	$\sqrt{\frac{2}{N+1}} \sin \frac{mk\pi}{N+1}$	$1-2\alpha \cos\frac{m\pi}{N+1}$	$2\rho^2\alpha^2$	$4\rho^2\alpha^4$
4	$k_1=k_2=1$ $k_3=k_4=0$	EVEN COSINE -1 (EDCT-1)	$\frac{1}{\sqrt{N}}$ $m=1$, $1 \le k \le N$ $\sqrt{\frac{2}{N}} \cos \frac{(2k-1)(m-1)\pi}{2N}$, $m=2, \ldots N$	$1-2\alpha \cos\frac{(m-1)\pi}{N}$	$2(1-\rho)^2\alpha^2$	$4(1-\rho)^2\alpha^4$
5	$k_1=k_2=-1$ $k_3=k_4=0$	EVEN SINE -2 (EDST-2)	$\sqrt{\frac{2}{N}} \sin \frac{(2k-1)m\pi}{2N}$, $m \ne N$ $\frac{1}{\sqrt{N}}$, $m=N$	$1-2\alpha \cos\frac{m\pi}{N}$	$2(1+\rho)^2\alpha^2$	$4(1+\rho)^2\alpha^4$
6	$k_1=0$, $k_2=+1$ $k_3=k_4=0$	ODD SINE -1 (ODST-1)	$\frac{2}{\sqrt{2N+1}} \sin \frac{(2m-1)k\pi}{2N+1}$	$1-2\alpha \cos\frac{(2m-1)\pi}{2N+1}$	$(1-\rho)^2\alpha^2+\rho^2\alpha^2$	$2(1-\rho)^2\alpha^4+2\rho^2\alpha^4$
7	$k_1=0$, $k_2=-1$ $k_3=k_4=0$	ODD SINE -2 (ODST-2)	$\frac{2}{\sqrt{2N+1}} \sin \frac{2mk\pi}{2N+1}$	$1-2\alpha \cos\frac{2m\pi}{2N+1}$	$(1+\rho)^2\alpha^2+\rho^2\alpha^2$	$2(1+\rho)^2\alpha^4+2\rho^2\alpha^4$
8	$k_1=+1$, $k_2=0$ $k_3=k_4=0$	ODD COSINE -1 (ODCT-1)	$\frac{2}{\sqrt{2N+1}} \cos \frac{(2k-1)(2m-1)\pi}{2(2N+1)}$	$1-2\alpha \cos\frac{2m\pi}{2N+1}$	$(1-\rho)^2\alpha^2+\rho^2\alpha^2$	$2(1-\rho)^2\alpha^4+2\rho^2\alpha^4$
9	$k_1=-1$, $k_1=0$ $k_3=k_4=0$	ODD SINE -3 (ODST-3)	$\frac{2}{\sqrt{2N+1}} \sin \frac{(2k-1)m\pi}{2N+1}$	$1-2\alpha \cos\frac{2m\pi}{2N+1}$	$(1+\rho)^2\alpha^2+\alpha^2\rho^2$	$2(1+\rho)^2\alpha^4+2\rho^2\alpha^4$
10	$k_1=+1$, $k_2=-1$ $k_3=k_4=0$	EVEN COSINE -2 (EDCT-2)	$\sqrt{\frac{2}{N}} \cos \frac{(2k-1)(2m-1)\pi}{4N}$	$1-2\alpha \cos\frac{(2m-1)\pi}{2N}$	$2\alpha^2(1+\rho^2)$	$4\alpha^4(1+\rho^2)$
11	$k_1=-1$, $k_2=+1$ $k_3=k_4=0$	EVEN SINE -3 (EDST-3)	$\sqrt{\frac{2}{N}} \sin \frac{(2k-1)(2m-1)\pi}{4N}$	$1-2\alpha \cos\frac{(2m-1)\pi}{2N}$	$2\alpha^2(1+\rho^2)$	$4\alpha^4(1+\rho^2)$

Table 2.1 : Some members of the sinusoidal transform family. (after reference [87])

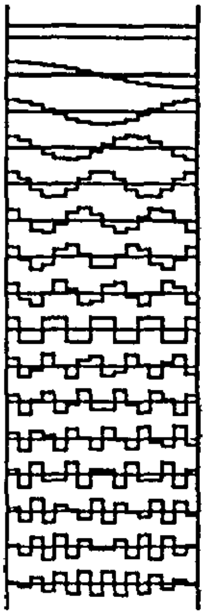


Fig. 2.4

The DCT basis functions for n=16.

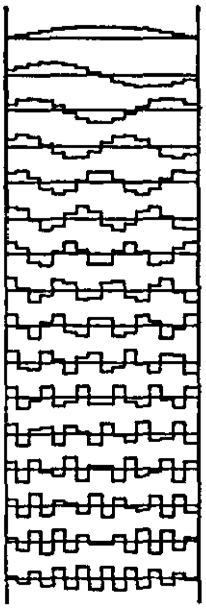


Fig. 2.5

The DST basis functions for n=16.

tional transform coding systems. On the other hand, if an n-dimensional vector X is a sample of the first-order Markov process, it has been shown that it has a decomposition [31]

$$X = X^{o} + X^{b} \text{ ------------------------- } (2.36)$$

where $$X^{b} = \alpha [Q]^{-1} \begin{bmatrix} x_{0} \\ 0 \\ \cdot\cdot \\ x_{n+1} \end{bmatrix} \text{ ----------------- } (2.37)$$

x_0 and x_{n+1} are sampled data immediately before and after vector X. The matrix [Q] is a symmetric, tridiagonal, Toeplitz matrix with unity along the main diagonal and $-\alpha$ along the other two diagonals, where $\alpha = \rho/(1+\rho^2)$, and ρ is the adjacent element correlation coefficient. X^b is called the boundary process whilst X^o is called the residual process. The DST is the optimum transform of the residual process, X^o. The class of coding schemes that makes use of this fact is called recursive block coding. Its coding procedures are given in section 1.3.3 and it has been shown to have better performance than conventional transform coding schemes.

2.4.2.2 Other orthogonal transforms

The most important non-sinusoidal orthogonal transform is perhaps the Walsh transform which only has element values of +1 and -1. Therefore, conversion of a signal vector into the Walsh transform domain requires only additions and subtractions. In addition, there are fast computational algorithms for the Walsh transform. The simplicity and ease of implementation of the Walsh transform has resulted in a wide range of applications [102-105] and investigations into its properties[88-90]. The generation and the properties of the Walsh transform are covered in chapter three. Fig.2.6 shows the basis functions of the 16x16 Walsh transform.

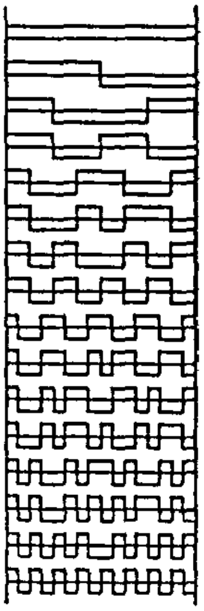


Fig.2.6

The Walsh transform basis functions for n =16.

Another orthogonal transform is the Haar transform which has an even smaller computational requirement than the Walsh transform. It is derived from the Haar matrix [91], which consists of plus and minus ones and zero elements. An example of an 8x8 Haar transform kernel is given below.

$$[\text{ Haar }] = \frac{1}{\sqrt{8}} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} \\ 2 & -2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & -2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & -2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & -2 \end{bmatrix} \text{ -----------} \tag{2.38}$$

Extensions to higher-order Haar matrices can be formed following the structure indicated by equation 2.38. The basis functions of a 16x16 Haar transform are shown in Fig.2.7. The Haar transform can be regarded as a sampling process in which basis vectors of the transform matrix sample the signal data with finer and finer resolution (increasing in powers of two).

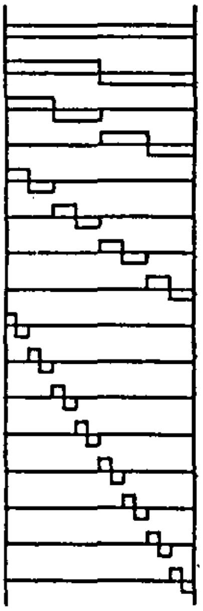


Fig.2.7

The Haar transform basis functions for n=16.

Both Walsh and Haar matrices were first developed by mathematicans and then borrowed for image processing.The first orthogonal transform designed specifically for pictorial data is the slant transform (the development of the slant transform can be found in section 1.3.1). A $2^m \times 2^m$ slant transform is designed to satisfy the following criteria: (1) one constant dc basis vector, (2) one slant basis vector monotonically decreasing in constant step size from a positive level to a negative level, (3) the sequency property (section 2.3.1), (4) a fast computational algorithm. Slant matrices of order 2 and 4 are

$$[\ S2\ ] = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \qquad \text{------------------------}(\ 2.39\ )$$

$$[\,S4\,] = \frac{1}{\sqrt{4}} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 3/\sqrt{5} & 1/\sqrt{5} & -1/\sqrt{5} & -3/\sqrt{5} \\ 1 & -1 & -1 & 1 \\ 1/\sqrt{5} & -3/\sqrt{5} & 3/\sqrt{5} & -1/\sqrt{5} \end{bmatrix} \text{------------}(2.40)$$

In general, the slant matrix of order n can be derived from the slant matrix of order n/2 via equation 2.41.

$$[\,S_n\,] = \frac{1}{\sqrt{2}} \left[\begin{array}{cc|c|cc|c} 1 & 0 & [0] & 1 & 0 & [0] \\ a_n & b_n & & -a_n & b_n & \\ \hline \multicolumn{2}{c|}{[0]} & [I]_{n/2-2} & \multicolumn{2}{c|}{[\,0\,]} & [I]_{n/2-2} \\ \hline 0 & 1 & [0] & 0 & -1 & [0] \\ -b_n & a_n & & b_n & a_n & \\ \hline \multicolumn{2}{c|}{[0]} & [I]_{n/2-2} & \multicolumn{2}{c|}{[0]} & -[I]_{n/2-2} \end{array}\right] \left[\begin{array}{c|c} [S]_{n/2} & [0] \\ \hline [0] & [S]_{n/2} \end{array}\right]$$

$$\text{------------------------------}(2.41)$$

where

$$a_{2n} = \left(\frac{3n^2}{4n^2-1} \right)^{1/2} \text{------------------------------}(2.42)$$

$$b_{2n} = \left(\frac{n^2-1}{4n^2-1} \right)^{1/2} \text{------------------------------}(2.43)$$

Fig.2.8 shows a 16x16 slant transform.

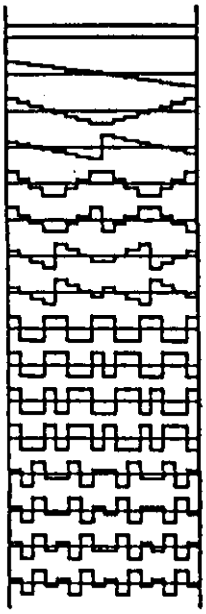


Fig.2.8

The slant transform basis functions for n=16.

2.5 OPTIMISATION OF PARAMETERS

In transform coding, data compression is essentially achieved by two processes. The first one is the transformation that packs most of the signal energy into a few coefficients. The second one is the quantization process in which the quantization error should be kept to a minimum. Therefore, to obtain efficient transform coding schemes which, whilst remaining relatively simple in implementation, achieve significant reduction in bit rate, both processes have to be optimum. Thus we have to choose the right transform and the right block size to optimize the first process; and choose the right quantizer and allocate a proper number of bits to each coefficient to optimize the second. The following four sections are devoted to these four considerations.

2.5.1 Selection of transform

The transformation that offers the best performance is the KLT which has the best energy packing ability, the highest ability to decorrelate signal data and results in the least mean square error. However, the KLT is data dependent and necessitates the computation of eigenvectors which creates many problems and prevents it from being used in practice.

In practice, the choice of transformations lies very much between the Walsh transform and the DCT depending on whether or not processing speed is paramount. In chapter four, two new transforms are proposed, which can be used as substitutes for the Walsh transform. The new transforms have virtually the same complexity and computational requirements as the Walsh transform but with performance which lies between that of the DCT and the Walsh transform. On the other hand, the discrete sine transform (DST) was found to have excellent performance when used with recursive block coding (section 1.3.3).

2.5.2 Selection of block size

Mean-square error performance should improve with increasing block size (n), since the number of correlations considered increases also. However, most pictures contain significant correlation between elements for only about 20 adjacent pels, although this number depends strongly on the amount of activity in the picture. Therefore, it seems very little can be gained by using block sizes larger than 32.

This can be verified by Fig.2.9 which contains a plot of the mean-square error for an image with a Markov process covariance as a function of block size for various transformations [22]. In this

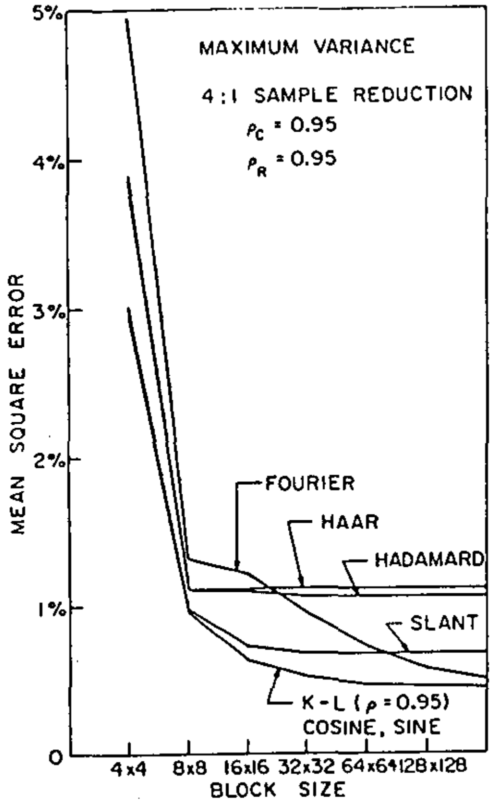


Fig.2.9

Mean square error performance of orthogonal transforms as a function of block size on an image with a Markov process covariance.

plot the 25 percent of coefficients with the largest variances were selected, and the remainder set to zero. From the figure, it is seen that the rate of decrease in mean square error for large block sizes becomes quite small for sizes larger than 16x16.

When subjective quality is the performance criterion, it is found that the result appears to be essentially the same for block size 4 or larger [26]. Furthermore, as given in section 1.3.3, large block sizes introduce two distinct disadvantages --- the requirement of large buffer memory and difficulty in achieving adaptation within a block. However, it should be noted that the choice of block size also depends on the kind of coding schemes used. For example, hybrid coding maintains its performance for small block sizes by taking into account interblock redundancy. In general, an optimum block size will be between 4 and 32.

2.5.3 Bit allocation

It is stated in rate distortion theory that the output of a source can be transmitted with average distortion D if the transmission rate is larger than the function R(D). For a source with Gaussian probability density function and mean square distortion measure D, the relation between D and R(D) is given by equation 1.1.

$$
\begin{aligned}
R(D) &= \log \sigma/\sqrt{D} & \sigma &> \sqrt{D} \\
&= 0 & \sigma &\leqslant \sqrt{D} \quad \text{--------------(1.1)}
\end{aligned}
$$

Equation 1.1 means that R(D)-bit quantization of a coefficient having variance σ^2 would result average distortion D. Therefore, given a distortion D,equation 1.1 can be used to determine the number of bits needed for the transmission of that coefficient. Usually, bits are allocated to the coefficients such that all the coefficients receive the same amount of distortion. However, in some schemes which aim at a better subjective quality instead of mean square error performance, the distortion allowed for each coefficient is modified according to a function H(w), the relative sensitivity of the human visual system to spatial light intensity distribution (section 1.3.3). Also, in most cases, 7 or 8 bits are allocated to dc coefficients regardless of equation 1.1. This is because 7 or 8-bit quantization of the dc coefficients is enough to make the quantization distortion imperceptible.

Exact equalization of quantization distortion for each ac coefficient

c_i requires $R_i(D)$, the number of bits allocated to c_i, to be a real number,but the number of bits allocated to a coefficient has to be an integer. Therefore, exact equalization of distortion is impossible in practice. However, the following steps can be used to achieve the closest possible equalization of distortion [96]:

(a) The quantization distortions of the ac coefficients for different numbers of bits are determined using equation 1.1. A bit counter is then set to zero.

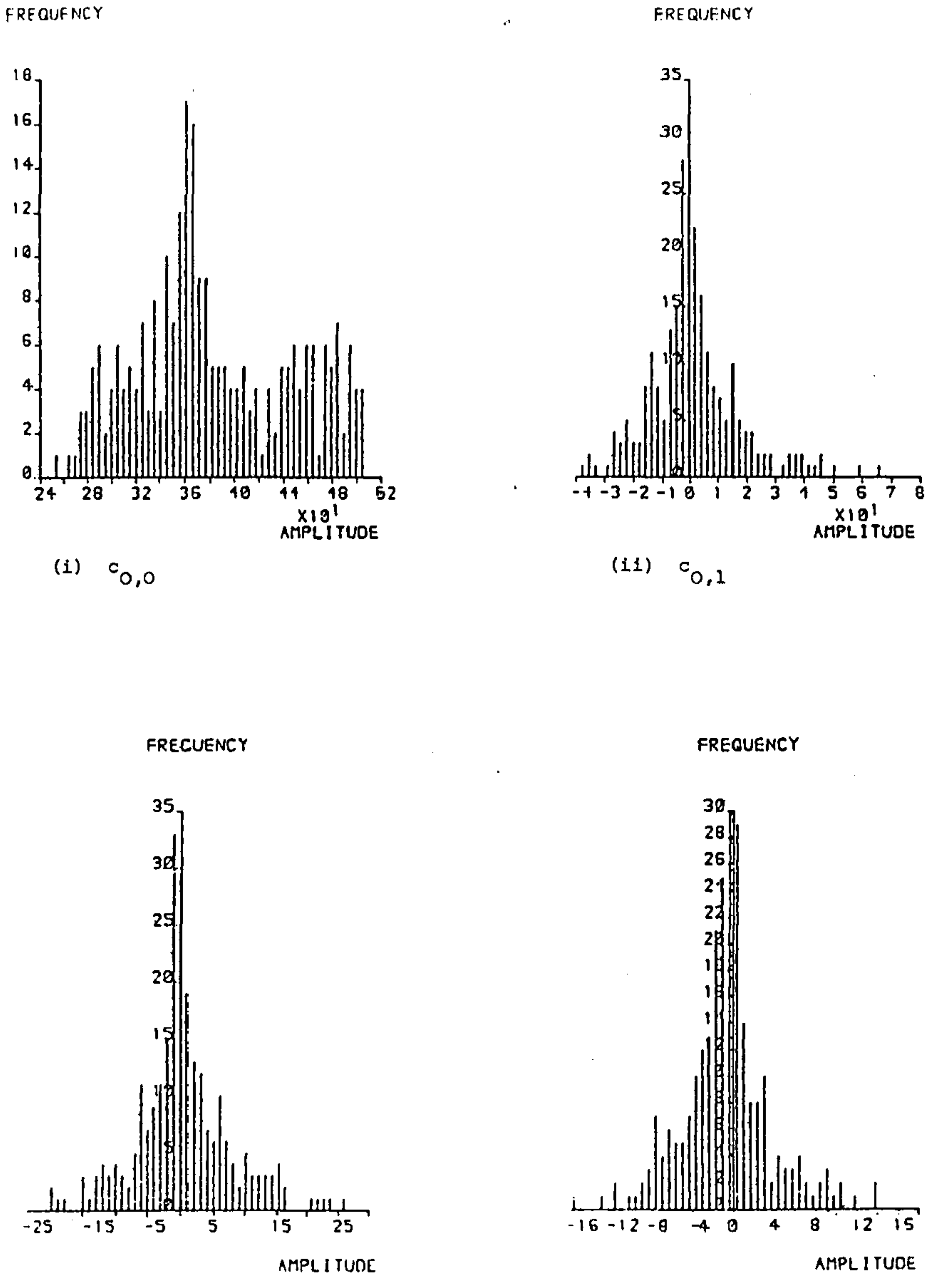
(b) Assign one bit to the ac coefficient with the largest distortion, thereby reducing the quantization distortion of that coefficient by 6db. Increment the bit counter by 1.

(c) If the bit counter equals the total number of bits available, stop, otherwise go to step (b).

2.5.4 Selection of quantizer

Given the probability density function (pdf) of a signal, the decision and reconstruction levels of the quantizer that minimizes the mean square quantization error can be found [92-94]. Usually, the pdf of transform coefficients is approximated by a function and then the quantizer that is optimum for that function is used. Ghanbari and Pearson [95] found that the distribution of the Walsh transform coefficients is approximated by a Gamma pdf. On the other hand, Chen and Smith modelled the pdf of the dc and ac DCT coefficients as Rayleigh and Gaussian densities respectively. However, sucessful results have been obtained by using a logarithmic model for the variances of ac transform coefficients [56,57]. For example, from a study of the histograms of the DCT coefficients of the 'Girl' picture (Fig.4.8a) as shown in Fig.2.10, it can be seen that the distribution of the ac coefficients is closer to a Laplacian than a Gaussian distribution. Therefore, it is believed that a Laplacian quantizer is more appropriate than a Gaussian quantizer for the ac coefficients [96].

On the other hand, the dc coefficients are simply the scaled sums of the pel levels within a block, so the distribution of the dc coefficients is closely related to that of the picture. Since pictures can have any distribution so may the dc coefficients, and therefore, a uniform quantizer is more suitable for their quantization.



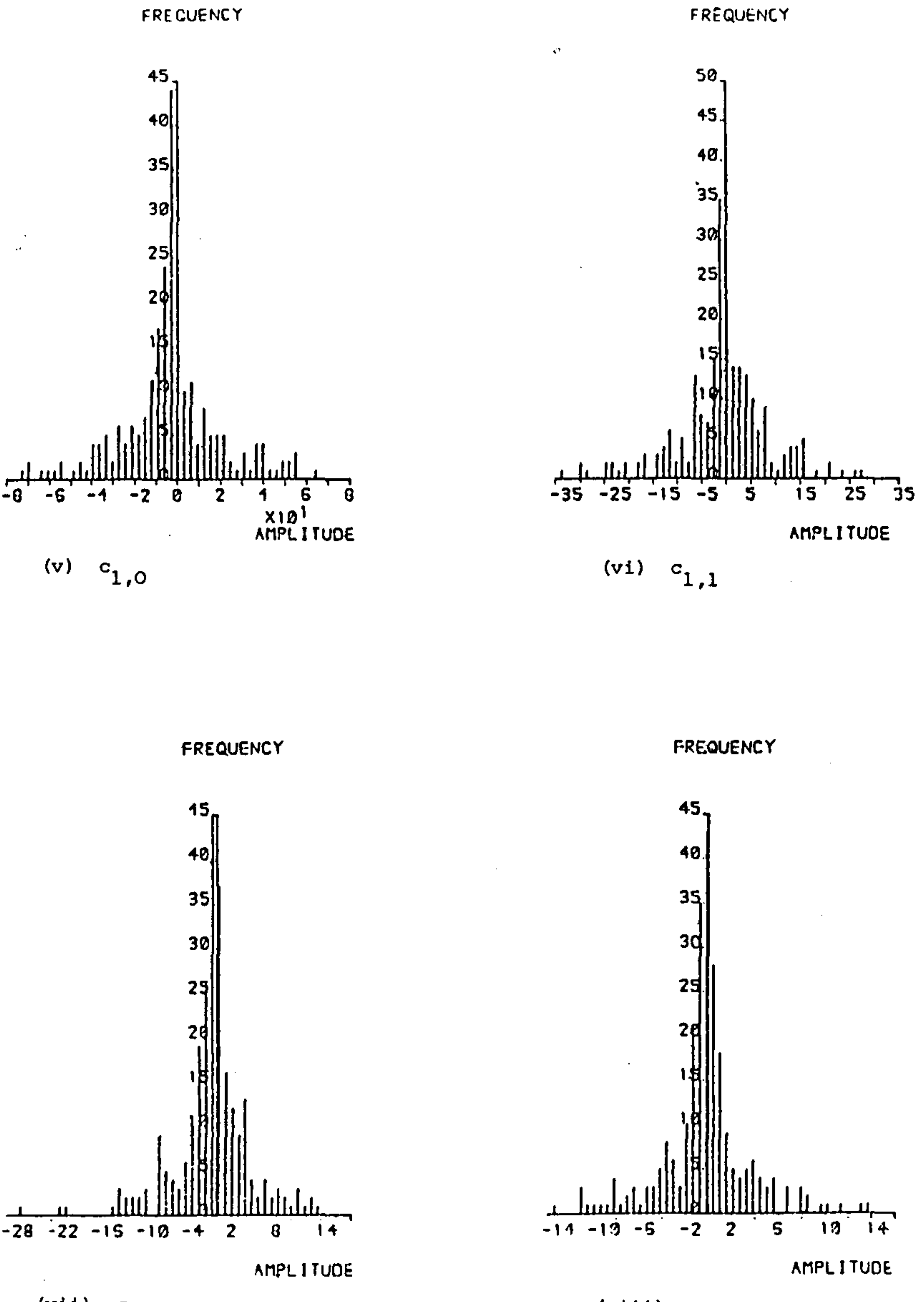
(i) $c_{0,0}$

(ii) $c_{0,1}$

(iii) $c_{0,2}$

(iv) $c_{0,3}$

Fig.2.10 Histograms of the DCT coefficients of the 'girl' picture.



(v) $c_{1,0}$

(vi) $c_{1,1}$

(vii) $c_{1,2}$

(viii) $c_{1,3}$

Fig.2.10 Histograms of the DCT coefficients of the 'girl' picture.

CHAPTER THREE

DYADIC SYMMETRY AND THE WALSH MATRICES

3.1 INTRODUCTION

In this chapter, a unified matrix treatment of Walsh transforms using the concept of dyadic symmetry is proposed. This unified treatment allows the straightforward derivation of a simple equation for the generation of Walsh matrices of different orderings, various re-ordering schemes and various fast computational algorithms, with the intention of providing a better understanding of the Walsh transform. Further, since the theory relates to a binary field with 'logical and' and modulo two addition as operations, it allows both the generation Walsh matrices of different orderings, and re-ordering schemes, to be carried out using simple logic circuits.

A historical note on the Walsh-Hadamard matrix will be given in the next section to describe the development of the Walsh-Hadamard transform as well as to clarify the nomenclature. In section 3.3, symmetry and dyadic symmetry within a vector are first defined, and then the properties of dyadic symmetry derived. The results obtained are then used in section 3.4 to obtain a non-recursive equation to generate binary Walsh matrices of different orderings. In section 3.5, the equation is used to derive some previously known results, including re-ordering schemes between different orderings, and the generation of Walsh matrices using Rademacher functions. In section 3.6, dyadic symmetry decomposition is defined and then used to generate various fast computational algorithms for Walsh transforms.

3.2 THE WALSH-HADAMARD MATRIX --- A HISTORICAL NOTE

It is well known that any waveform , $x(t)$,having a finite energy in an interval, say $[0,1)$[*1] , can be expressed as a weighted sum of a complete set of orthogonal functions. For example, if the complete set of orthogonal functions is $\{ \exp(j2\pi tk) \}$, then we have the famous Fourier series representation:

$$x(t) = \sum_{k=-\infty}^{\infty} c_k \cdot e^{-j2\pi tk}$$

$$c_k = \int_0^1 x(t) \cdot e^{j2\pi tk} \, dt \qquad t \varepsilon [0,1)$$

In 1922, H.Rademacher [77] devised an incomplete set of orthogonal functions which were then called Rademacher functions $\{ \mathrm{Rad}_i(t) \}$. Rademacher functions are defined within the half open interval $[0,1)$, and take the values +1 and -1. Fig.3.1 shows the first four Rademacher functions. The first Rademacher function $\mathrm{Rad}_0(t)$ is a unit pulse.

$$\mathrm{Rad}_0(t) = +1 \qquad t \varepsilon [0,1)$$

$R_i(t)$, $i \geqslant 1$,can then be generated using the recursive equations

[*1] $t \varepsilon [0,1)$ means $0 \leqslant t < 1$

$$\mathrm{Rad}_i(t/2) = \mathrm{Rad}_{i-1}(t)$$

$$\mathrm{Rad}_i(t/2+1/2) = -\mathrm{Rad}_{i-1}(t) \qquad t \in [0,1)$$

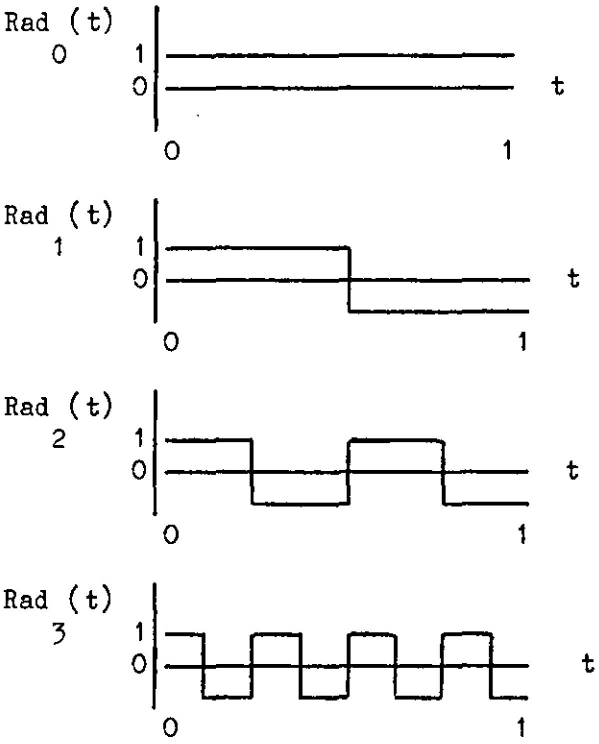


Fig.3.1 The first four Rademacher functions

The following year, J.L.Walsh added more new functions to the Rademacher functions and formed a complete orthonormal set of rectangular functions, now known as Walsh functions [119]. However, attention has been mainly concentrated on Fourier analysis, leading to the development of the fast Fourier transform (FFT) and a complete theory for discrete systems (excellent reviews of the progress made in these areas can be found in the papers by Jury [97], and Cooley et al [98]). In the late 1960's, the fast Fourier transform (FFT) was first applied to digital image coding [99]. At about the same time, Harmuth [100], using the Walsh functions as an example, generalized the concept of frequency (for sinusoidal functions only), to 'sequency' for any type of function. W.K.Pratt and others then used the Walsh transform, derived from the Walsh functions, in place of the FFT for image coding [101].

As for the Walsh functions, the Walsh matrix contains only the values +1 and -1. Therefore, conversion of a signal vector into the Walsh transform domain involves only simple additions and subtractions. Moreover, there exists a fast Walsh transform algorithm similar to that of the fast Fourier transform, and therefore the computational requirement of the Walsh transform is much less than that of the FFT.

In the early 1970's, the simplicity and ease of implementation of the Walsh transform resulted in a wide range of applications [102-5], including analysis, filtering and data compression of speech, electrocardio-, and electroencephalograms, and other signals, as well as for the multiplexing of communication channels and the processing of images for pattern recognition, data compression and image enhancement. All these processes are performed in the sequency domain in much the same way as they would be in the frequency domain. For pattern recognition or waveform analysis, one can search for recognizable configurations of Walsh coefficients. For data compression, instead of sending the signal, one can transmit Walsh coefficients with more bits allocated to those having larger variances. Section 1.3 has given a detailed review of this technique. To filter and enhance an image, each coefficient is multiplied by an appropriate function of its sequency (and, possibly, added to some linear combination of other coefficients) before inverse transformation back to the signal domain. To multiplex several signals that have constant amplitude over a period T, carrier Walsh functions whose amplitude represents each waveform are added together and sent through the channel. The orthogonal property of Walsh functions is then used to extract the signals at the distant end of the channel.

The effectiveness of most of these applications however, especially filtering and data compression, depends on one single important property. In the case of the Walsh transform, it is the ability to pack the signal energy into a few transform coefficients. For the Walsh functions, it is the ability to represent a signal waveform accurately using as few terms as possible.

Unfortunately, the Walsh functions and Walsh transform are inferior to Fourier series representation and discrete Fourier transform in that respect [106]. It is found that, to represent a smooth signal, far more terms are required in the Walsh series representation. Even for discontinuous signals, the Walsh series may also require a lot more terms. On the other hand, many other transforms have been found to have higher energy packing ability than the Walsh transform. Thus, in the late 1970's, the interest in applications of the Walsh functions and Walsh transform was diminishing.

Up to this time, however many fruitful results had been found. These results can be generally grouped into four areas:

1. New and better methods of generating Walsh matrices and Walsh functions.
2. New and more efficient fast computational algorithms for the Walsh transform.
3. Better understanding of the properties of Walsh functions (usually obtained by comparison with Fourier analysis).
4. Special applications and implementation methods.

During the development, different researchers adopted different nomenclatures and so created a lot of confusion. In this thesis, "Walsh functions" will refer to the set of orthogonal functions proposed by J.L.Walsh. Many other methods have also been found to generate the set of Walsh functions [107-8], some of which have individual functions ordered in different ways. Generally, the set of Walsh functions, and its discrete counterpart the Walsh transform, are classified into three groups according to their ordering.

1. Sequency-ordered Walsh functions and transform [109]:

These functions are also known as zequency-ordered Walsh functions [112] Walsh-ordered Walsh functions [110] or simply Walsh functions. Their discrete counterpart, the sequency-ordered Walsh transform, is also called the Walsh transform [114].

2. Dyadic-ordered Walsh functions and transform [109]:

These functions are also known as Paley-ordered Walsh functions, and their discrete counterpart is also called the Paley-ordered Walsh transform [110].

3. Natural-ordered Walsh functions and transform [109]:
These functions are also known as Hadamard-ordered Walsh functions [110] or the binary Fourier representation (BIFORE) [111,113]. Their discrete counterpart, the natural-ordered Walsh transform, is sometimes called the BIFORE transform [111] or simply the Hadamard transform [111,113].

On the other hand, the Hadamard matrix is defined as a square matrix of only plus and minus one whose rows (and columns) are orthogonal to one another. Hadamard functions, their counterpart in the continuous case, are also called Walsh-like functions [115]. Therefore, under this nomenclature,Walsh functions and transforms are particular cases of Hadamard functions and transforms.The lowest order Hadamard matrix is two by two and unique as shown below.

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

Methods of generating Hadamard matrices of other sizes can be found in the literature [115-6,118]. For any block size, it is always possible to derive from each Hadamard matrix a limited number of other Hadamard matrices. The Hadamard matrices which can be converted to each other by permutation of rows and by inversion of row signs are said to be identical. For example, the following two Hadamard matrices are identical.

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \equiv \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix}$$

In fact, they are the natural-ordered and sequency-ordered Walsh matrices. For a block size of four by four, there are only two non-identical Hadamard matrices [117]. They are

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} 1 & 1 & 1 & -1 \\ 1 & 1 & -1 & 1 \\ 1 & -1 & 1 & 1 \\ -1 & 1 & 1 & 1 \end{bmatrix}$$

The number of non-identical Hadamard matrices rises quickly with the block size. For a block size of eight by eight, the number of non--identical Hadamard matrices is already 432 [118]. From now on, attention is concentrated on the Walsh functions and transform. As the conversion between the Walsh functions and transform are straightforward, so results discovered for one are always applicable to the other.

Certain methods of generating Walsh functions were discovered by early pioneers [108,119,120]. Since then, many other methods have been found. Some methods aimed at providing a straightforward derivation, some were developed for various special purposes. One of the early attempts was to define the dyadic-ordered Walsh functions in terms of products of Rademacher functions [121-2]. This definition is convenient because Rademacher functions are simple and easy to remember, and their products are easy to form. In 1964, K.W.Henderson [107] found two simple methods of generating the binary sequency-ordered Walsh matrix. One method uses the gray code and Rademacher functions and the other uses the gray code and the binary code. In the same paper, a method of generating the natural-ordered Walsh matrix using only the binary code was given. In

1968, in contrast to the Fourier transform, which can be defined by linear differential equations of second order, H.F.Harmuth [100] found a difference equation by which the sequency-ordered Walsh functions could be defined.

In 1969, three more methods were proposed. P.A.Swick found a simpler method of generating sequency-ordered Walsh functions of any order by symmetry considerations [123]. W.K.Pratt and others found a way to generate the Walsh transforms which can be easily generalized to higher dimensions [101], and J.L.Shanks defined the dyadic-ordered Walsh functions using iterative equations for his development of a fast computational algorithm for the Walsh transform [124].

On the other hand, in searching for faster and more efficient methods for the Walsh transform, researchers have found a number of different fast computational algorithms, usually by suitably modifying the Cooley-Tukey fast Fourier transform algorithm. In 1968, based on the well known recursive structure of the natural-ordered Walsh matrix, Whechel and Guinn [125] derived a fast computational algorithm for the Walsh transform. Later, Shanks [124] derived an iterative equation for the dyadic-ordered Walsh matrix, and based on this equation, developed another fast computational algorithm. Whechel and Guinn's algorithm yields the transform coefficients in natural order while Shanks's algorithm results transform coefficients in dyadic order, which is the natural-ordered form after bit reversal.

Most applications, however, require transform coefficients arranged in sequency order. Therefore, the above two fast compuational algorithms require an extra process to re-order the transform coefficients to give a set of sequency-ordered transform coefficients. Conversion from dyadic-ordering to sequency-ordering is a simple permutation based on the gray code. Conversion from natural-ordering is usually done first by conversion to dyadic-ordering using bit reversal and then finally to sequency-ordering using gray code permutation [110].

In 1972, Manz [126],by suitably modifying Shanks's algorithm, derived a fast computational algorithm which results in the sequency-ordered transform coefficients when the input data is in bit-reversed order. At about the same time, Fino [127] and Fontaine [128] both produced fast computational algorithms which accept data in normal order and result in transform coefficients in sequency order. However, the algorithms require an extra N auxiliary storage locations (N is the number of input data points) which may eliminate any computational advantages. Larsen [129] later discovered a fast computational algorithm which was regarded as complementary to Manz's algorithm. Larsen's algorithm shares the advantages of Manz's algorithm but differs from it in that it accepts data in normal order and returns the transform coefficients in bit-reversed sequency order.

Throughout the development of Walsh transform theory, different nomenclatures and different methods for the generation of Walsh transforms have been adopted, leading to both rediscoveries [130-1], and to a certain degree of confusion [132-3]. Attempts have been made, therefore, to unify the nomenclature and Fino and Algazi produced a unified matrix treatment to provide a common framework for all areas of interest [134]. They defined Walsh transforms having different orderings using the Kronecker matrix product and various permutation matrices, such as perfect shuffling and block diagonal matrices. From these definitions, various fast computational algorithms for Walsh transforms,and the various re-ordering schemes,were derived by matrix manipulation. However, these definitions of Walsh transforms are more complex than most of the conventional ones. A good understanding of the properties of the various types of permutation matrices and the Kronecker product are also required to derive the various fast computational algorithms and various re-ordering schemes.

In this chapter, an alternative unified matrix treatment which is defined for the binary Walsh matrix instead of for the Walsh transform is proposed. Each element of the binary Walsh matrix is 0 or 1 and the conversion between the two type of matrices can be obtained from the transformation: { 1,-1 } in the Walsh matrix <--> { 0, 1 } in the binary Walsh matrix. This unified treatment allows the derivation of a simple equation for the generation of Walsh transforms of different orderings, various re-ordering schemes and various fast computational algorithms, within the same framework, by using the concept of dyadic symmetry.

In the next section, symmetry and dyadic symmetry within a vector are first defined, and then the properties of dyadic symmetry derived.

The results obtained are then used in section 3.4 to obtain a non-recursive equation for the generation of binary Walsh matrices of different orderings. In section 3.5, the equation is used to derive some previously known results, including the re-ordering schemes between different orderings, and the generation of Walsh matrices using Rademacher functions. In section 3.6, dyadic symmetry decomposition is defined and then used to generate various fast computational algorithms for Walsh transforms.

3.3 <u>THE BASIC THEORY OF DYADIC SYMMETRY</u>

In this section, basic definitions of even and odd symmetry are given. Attention is then concentrated on the n-1 dyadic symmetries among the numerous possible symmetries within a vector of n elements. The properties of dyadic symmetry are derived and an 8x8 Walsh transform is then generated as an example of the application of dyadic symmetry.

3.3.1 <u>The basic definition of symmetry</u>

<u>Definition</u> 3.1:

(1) A particular type of EVEN symmetry is said to exist in a vector of n elements if and only if the n elements can be divided into n/2 pairs of elements of the same value.

(2) A particular type of ODD symmetry is said to exist in a vector of n elements if and only if the n elements can be divided into n/2 pairs of elements of the same magnitude and opposite sign.

The above definitions suggest that n must be an even number, and the relation between n and the possible number of even symmetries existing in a vector of n elements is

$$P = (n-1)\ (n-3)\ (n-5)\ \ldots\ 1 \text{ ----------------------------------- } (\ 3.1\ )$$

For, let V be the vector $[\ a_1 , a_2 , a_3 , \ldots, a_n\ ]$. There are n-1 ways to form a pair after picking one element arbitrarily, n-3 number of ways to form another pair, and then n-5, n-7 and so on. Table 3.1 lists some values of P and n.

n	P
2	1
4	3
8	105
16	2027025
32	1.9×10^{17}

Table 3.1 : P is the number of even symmetries which exist in a vector of n elements.

3.3.2 The basic definition of dyadic symmetry

Definition 3.2:

A vector of 2^m elements $[\ a_0 , a_1 , \ldots, a_{2^m-1}\ ]$ is said to have the i th dyadic symmetry if and only if

$$a_j = s \times a_{j(+)i}$$

where

(i) (+) is 'exclusive or'

(ii) j lies in the range $[\ 0,\ 2^m -1\ ]$ and i in the range $[\ 1,\ 2^m -1\ ]$

(iii) s = 1 when the symmetry is even

s = -1 when the symmetry is odd.

Table 3.2 shows the dyadic symmetry when m is three and thus n is 2^3 or 8. The vectors, H_i , having even i th dyadic symmetry are given in Table 3.3.

	i	1	2	3	4	5	6	7
j		001	010	011	100	101	110	111
0	000	1	2	3	4	5	6	7
1	001	0	3	2	5	4	7	6
2	010	3	0	1	6	7	4	5
3	011	2	1	0	7	6	5	4
4	100	5	6	7	0	1	2	3
5	101	4	7	6	1	0	3	2
6	110	7	4	5	2	3	0	1
7	111	6	5	4	3	2	1	0

Table 3.2 : The value of i(+)j for i in the range [1,7] and j in the range [0,7].

i	vector H_i
1	$[h_1\ h_1\ h_2\ h_2\ h_3\ h_3\ h_4\ h_4]$
2	$[h_1\ h_2\ h_1\ h_2\ h_3\ h_4\ h_3\ h_4]$
3	$[h_1\ h_2\ h_2\ h_1\ h_3\ h_4\ h_4\ h_3]$
4	$[h_1\ h_2\ h_3\ h_4\ h_1\ h_2\ h_3\ h_4]$
5	$[h_1\ h_2\ h_3\ h_4\ h_2\ h_1\ h_4\ h_3]$
6	$[h_1\ h_2\ h_3\ h_4\ h_3\ h_4\ h_1\ h_2]$
7	$[h_1\ h_2\ h_3\ h_4\ h_4\ h_3\ h_2\ h_1]$

Table 3.3 : The seven vectors, H_i, having i th even dyadic symmetry.

It should be noted that each combination of two elements appears once and once only in each dyadic symmetry. For example, the combination of a_0 and a_1 appears only in the first, and in no other, dyadic symmetry.The total number of combinations of two elements in a vector of n elements is

$$n\ C\ 2 = n(n-1)/2 \qquad \text{-----}(3.2)$$

Each possible dyadic symmetry requires n/2 combinations. Therefore, the number of possible symmetries is

$$P = n\ C\ 2\ /\ (n/2) = n - 1 \qquad \text{-----}(3.3)$$

There are thus seven dyadic symmetries in a vector of eight elements.

3.3.3 Some properties of dyadic symmetry

Theorem 3.1:

If a 2^m-vector has dyadic symmetries $S_1, S_2, \ldots, S_r$, this vector has also dyadic symmetry S_k where $S_k = S_1 (+) S_2 (+) \ldots\ldots S_r$ and (+) is 'exclusive or'.

Proof:

Let vector A be $[a_0, a_1, a_2, \ldots, a_{2^m-1}]$ having dyadic symmetry $S_1, S_2, \ldots, S_r$ As given by the definition of dyadic symmetry, we have

$$a_j = s_1 a_{j(+)S_1}$$

$$a_j = s_2 a_{j(+)S_2}$$

$$a_j = s_3 a_{j(+)S_3}$$

$$\ldots\ldots\ldots\ldots\ldots\ldots$$

$$\ldots\ldots\ldots\ldots\ldots\ldots$$

$$a_j = s_r a_{j(+)S_r}$$

for all j within the range $[0, 2^m-1]$ where $s_i = 1$ or -1 for i within $[1, r]$.

Combining the first two equations together

$$a_{j(+)S_1} = s_1 \; s_2 \; a_{j(+)S_2}$$

Since both j and $j(+)S_1$ are within $[0,2^m-1]$, j can be replaced by $j(+)S_1$

$$a_{j(+)S_1(+)S_1} = s_1 \; s_2 \; a_{j(+)S_1(+)S_2}$$

$$a_j = s_1 \; s_2 \; a_{j(+)S_1(+)S_2}$$

Continuing with the same procedure for S_3, S_4,....., and S_r, we have

$$a_j = s_k \; a_{j(+)S_k} \quad \text{------------------------------------}(3.4)$$

where

(i) $s_k = s_1 \, s_2 \, s_3 \; \ldots \; s_r$

(ii) $S_k = S_1 \; (+) \; S_2 \; (+) \; \ldots \; (+) \; S_r$

The relationships within the 2^m-1 dyadic symmetries will now be examined in more detail. It will be shown that some m of them are independent and all the 2^m-1 dyadic symmetries can be expressed as linear combinations of m independent symmetries.

Let F be a binary field which has 0 and 1 as its elements, and 'logical and' $\{ * \}$ and 'exclusive or' $\{ (+) \}$ as its operations. For a vector over a number field of 2^m elements, there are 2^m-1

dyadic symmetries. These 2^m-1 dyadic symmetries are the m-vectors over the field F.

Definition 3.3:

The r dyadic symmetries, represented by the r binary m-vectors

$$
\begin{aligned}
S_1 &= [\, s_{11} , s_{12} , \ldots\ldots\ , s_{1m} \,]^t \\
S_2 &= [\, s_{21} , s_{22} , \ldots\ldots\ , s_{2m} \,]^t \\
&\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\
&\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\
S_r &= [\, s_{r1} , s_{r2} , \ldots\ldots\ , s_{rm} \,]^t
\end{aligned}
$$

are said to be dependent if there exist r elements $k_1 , k_2 , \ldots , k_r$, not all zero, such that

$$k_1 * S_1 \ (+)\ k_2 * S_2 \ (+)\ \ldots\ (+)\ k_r * S_r = 0 \quad \text{-----------}(\ 3.5\ )$$

Otherwise, the r symmetries are said to be linearly independent.

For example, the first, third and seventh dyadic symmetries which are represented by the three binary 3-vectors [0,0,1], [0,1,1] and [1,1,1] are independent. On the other hand, the first, second and third dyadic symmetries, represented by the three binary 3-vectors [0,0,1], [0,1,0] and [0,1,1], are dependent.

If in equation 3.5, k_i is not equal to zero, we may solve for

$$S_i = k_1 * S_1 \ (+) \ \ldots \ (+) \ k_{i-1} * S_{i-1} \ (+) \ k_{i+1} * S_{i+1} \ (+) \ \ldots \ldots\ldots\ldots (+) \ k_r * S_r \qquad \text{-----------}(3.6)$$

Therefore, the following properties exist:

(a) If r dyadic symmetries are dependent, any of them may always be expressed as a linear combination of the others.

(b) If r dyadic symmetries are independent then none of them may be expressed as a linear combination of the others.

(c) If r dyadic symmetries are independent while the set obtained by adding another dyadic symmetry S_{r+1} is dependent, then S_{r+1} can be expressed as a linear combination of S_1, S_2, ..., S_r

In an m-dimensional vector space over a field, the following two properties are well known.

(i) An m by m matrix [S] has an inverse if and only if the m row or column vectors of [S] are independent.

(ii) There are at most m independent vectors in a m-dimensional vector space over a field.

Thus, we have the following further properties of dyadic symmetry:

(d) The m dyadic symmetries represented by binary m-vectors S_1, S_2, .., S_m are independent if and only if the binary matrix [S] has an inverse, where [S] has the dyadic symmetries as its row or column vectors.

(e) There are 2^m-1 dyadic symmetries for a 2^m-vector and no more than m of the dyadic symmetries are independent.

(c) and (e) above lead directly to :

(f) All the 2^m-1 dyadic symmetries can be expressed as a linear combinations of m independent dyadic symmetries. That is

$$S_i = k_1 * S_1 \ (+) \ k_2 * S_2 \ (+) \ .... \ (+) \ k_m * S_m$$

where i is within $[1,2^m-1]$ and $S_1, S_2, ..., S_m$ are the m independent dyadic symmetries.

For example, the set of dyadic symmetries defined by the set M are independent.

$$M = \{ \ 2^k-1, \ k \in [1,m] \ \}$$

This set of dyadic symmetries will be called Mirror symmetry (or M-symmetry), and the 2^k-1 th dyadic symmetry will be called the k th M-symmetry. Table 3.4 shows the 2^m-vectors which have the even i th M-symmetry. Table 3.5 gives the corresponding dyadic symmetries for the M-symmetries of block size 8x8, 16x16 and 32x32.

i	H_i
1	a a b b .... c c d d
..	....
..	....
m-1	a_1 a_2 ..... a_2 a_1 b_1 b_2 .... b_2 b_1
m	a_1 .... $a_{n/2-1}$ $a_{n/2}$ $a_{n/2}$ $a_{n/2-1}$ .... a_1

Table 3.4 : The 2^m-vectors H_i having the even i th M-symmetry.

M-symmetry	3	2	1			block size 8
dyadic symmetry	7 111	3 011	1 001			
M-symmetry	4	3	2	1		block size 16
dyadic symmetry	15 1111	7 0111	3 0011	1 0001		
M-symmetry	5	4	3	2	1	block size 32
dyadic symmetry	31 11111	15 01111	7 00111	3 00011	1 00001	

Table 3.5 : The relation between M-symmetry and dyadic symmetry

3.3.4 Independent dyadic symmetry and the Walsh matrix ----- an example

It is well known that all the elements of a Walsh matrix have the same magnitude. Further, it is interesting to note that the distribution of signs within a $2^m \times 2^m$ Walsh matrix is such that each row and column have all the 2^m-1 dyadic symmetries. Now, using an 8x8 Walsh matrix as an example, it will be shown how m independent mirror symmetries can be used to generate any basis vector of a $2^m \times 2^m$ Walsh matrix.

If we use '0' to represent even dyadic symmetry, '1' to represent odd dyadic symmetry, the dyadic symmetries [0...01], [0..011], [0..111], ..., [1..111] of a Walsh basis vector can be used to form a code which indicates the number of zero crossings within that vector.

For example, let us consider H_5, the Walsh basis vector having five (i.e. 101) zero crossings. As the first element is always +1 and the first dyadic symmetry is 1 (i.e.odd), the first two elements are [1,-1]. Also, because the third dyadic symmetry is even, the first four elements are [1,-1,-1, 1]. Finally, the seventh symmetry is odd, therefore, the vector is :

$$H_5 = [\ 1,-1,-1,\ 1,-1,\ 1,\ 1,-1\ ]$$

Finally, the Walsh matrix is orthogonal as a consequence of:

<u>Theorem</u> 3.2:

Two N-dimensional vectors U and V are orthogonal if U and V have the same type of symmetry and U is even and V is odd.

Proof:

As U has a particular type of even symmetry, so elements of U can be grouped into n/2 ordered pairs, (u_{a_1}, u_{b_1}), (u_{a_2}, u_{b_2}),........,

$(u_{a_{n/2}}, u_{b_{n/2}})$ where a_i, b_i are integers within [1,n], i is an integer within the range [1,n/2], and n is the dimension of the vectors U and V. As V has the same type of symmetry (except that it is odd), so elements of V can be grouped into n/2 ordered pairs,

$$(v_{a_1}, v_{b_1}), (v_{a_2}, v_{b_2}), \ldots\ldots, (v_{a_{\frac{n}{2}}}, v_{b_{\frac{n}{2}}})$$

with $v_{a_i} = -v_{b_i}$ and $u_{a_i} = u_{b_i}$.

Therefore,

$$\begin{aligned} U^t \times V &= \sum_{k=1}^{n} (u_k \cdot v_k) \\ &= \sum_{i=1}^{n/2} (u_{a_i} \cdot v_{a_i}) + \sum_{j=1}^{n/2} (u_{b_j} \cdot v_{b_j}) \\ &= \sum_{i=1}^{n/2} (u_{a_i} \cdot v_{a_i} + u_{b_i} \cdot v_{b_i}) \\ &= 0 \end{aligned}$$

The dot product between U and V is zero, and therefore U and V are orthogonal.

Since all the Walsh basis vectors have at least one different dyadic symmetry, this theorem leads directly to the result that all the basis vectors of Walsh matrices are orthogonal.

3.4 GENERATION OF A WALSH MATRIX

3.4.1 The definition of a Walsh matrix

First of all, the Walsh matrix is defined as:

Definition 3.4:
The 2^m 2^m-dimensional vectors derived from the 2^m possible combinations of m independent dyadic symmetries are all different and and orthogonal, and they are the basis vectors of a Walsh matrix.

From property (b) in section 3.3.4, none of the m independent dyadic symmetries can be derived from combinations of the others. This implies that the m dyadic symmetries can form 2^m different combinations, and so by theorem 2.2 are orthogonal.

This definition reveals that any m independent dyadic symmetries from the total of the 2^{m-1} dyadic symmetries can be used to generate the 2^m 2^m-dimensional basis vectors of a 2^m by 2^m Walsh matrix. However, the actual ordering of the basis vectors still depends on the choices of 1) the m independent dyadic symmetries and 2) the m-bit code used.

Section 3.3.4 illustrates the generation of an eight by eight sequency-ordered Walsh matrix using the binary code and independent dyadic symmetry [1,3,7]. Table 3.6 summarizes the independent dyadic symmetries for the generation of the 8 x 8 sequency-ordered, dyadic-ordered and natural-ordered Walsh matrices using binary code and gray code.

code ordering	binary code	gray code
sequency	[1,3,7]	[1,2,4]
dyadic	[1,2,4]	[1,3,6]
natural	[4,2,1]	[4,6,3]

Table 3.6 : The codes and dyadic symmetries for the generation of 8x8 Walsh transforms of sequency, dyadic, and natural orderings

3.4.2 Derivation of the non-recursive equation for the binary Walsh matrix

For clarity, block size 8 x 8 is use as an example. The result can easily be generalized to block size $2^m \times 2^m$. Let [H] be the Walsh matrix of a particular ordering

$$[\,H\,] = \begin{bmatrix} h_{00} & h_{01} & \cdots\cdots\cdots & h_{07} \\ h_{10} & h_{11} & \cdots\cdots\cdots & h_{17} \\ \cdots & \cdots & \cdots\cdots\cdots & \cdots \\ h_{70} & h_{71} & \cdots\cdots\cdots & h_{77} \end{bmatrix} \qquad \text{---------------(3.7)}$$

where h_{ij} is the (i,j) th element of [H], $i=[\,i_1, i_2, i_3\,]$ and $j=[\,j_1, j_2, j_3\,]$.

Also, let

$$\begin{aligned} h_{ij} &= +1 \quad \text{if} \quad b_{ij} = 0 \\ &= -1 \quad \text{if} \quad b_{ij} = 1 \end{aligned} \qquad \text{------------------------(3.8)}$$

and the binary matrix [B] whose (i,j)th element is b_{ij} be called the binary Walsh transform. Then, the determination of [B] leads directly to that of [H] and vice versa.

Further, let S_1 , S_2 and S_3 be the binary representation of the three independent dyadic symmetries.

$$S_1 = [\, s_{11} , s_{12} , s_{13} \,]^t$$
$$S_2 = [\, s_{21} , s_{22} , s_{23} \,]^t$$
$$S_3 = [\, s_{31} , s_{32} , s_{33} \,]^t$$

The binary matrix [S] will be called the dyadic symmetry matrix where

$$[\, S \,] = \begin{bmatrix} S_1^t \\ S_2^t \\ S_3^t \end{bmatrix} = \begin{bmatrix} s_{11} & s_{12} & s_{13} \\ s_{21} & s_{22} & s_{23} \\ s_{31} & s_{32} & s_{33} \end{bmatrix}$$

In this section, it will be shown that

$$b_{ij} = [\, j_1 , j_2 , j_3 \,] * [\, S \,]^{-1} * [\, i_1 , i_2 , i_3 \,]^t$$
$$= j^t * [S]^{-1} * i \quad \text{--------------------------------} (3.9)$$

It has already been pointed out in section 3.3.4 property (d) that [S] always has an inverse if dyadic symmetries S_1 , S_2 and S_3 are independent. In order to clarify the development, the eight by eight sequency-ordered binary Walsh matrix will be used as an example. First of all, the following may be noted:

(1) If the binary representation of i is $[\, i_1, i_2, i_3 \,]$, then i_1, i_2 and i_3 determine the type (even or odd) of the three independent dyadic symmetries S_1, S_2 and S_3 within H_i.

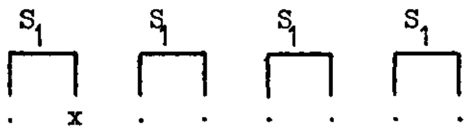
(2) The dyadic symmetry matrix of a sequency-ordered Walsh matrix is

$$[\, S \,] = \begin{bmatrix} S_1^t \\ S_2^t \\ S_3^t \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

(3) The (0,0,0)th element of every Walsh basis vector is usually taken to be +1. This convention will be adopted here also, and therefore the (0,0,0)th element of every binary Walsh basis vector is 0.

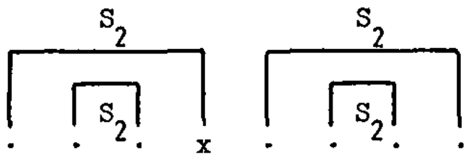
There are eight elements in H_i and h_{i0} is always +1. Now, we wish to determine the sign of h_{ij} ,which is the $(\, j_1, j_2, j_3 \,)$th element of the $(\, i_1, i_2, i_3 \,)$th basis vector. Consider the following three cases.

(i) When $[\, j_1, j_2, j_3 \,]$ is [0,0,1]



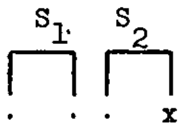
It is obvious that the sign of the (0,0,1)th element depends only on symmetry S_1 and so on i_1 in the vector $[\, i_1, i_2, i_3 \,]$. An i_1 of value 0 or 1 implies even or odd symmetry and so the (0,0,1)th and (0,0,0)th elements have the same or opposite sign.

(ii) When $[\, j_1, j_2, j_3 \,]$ is $[\, 0,1,1 \,]$



The sign of the (0,1,1)th element depends only on S_2 and so on i_2 in the vector $[\, i_1, i_2, i_3 \,]$.

(iii) When $[\, j_1, j_2, j_3 \,]$ is $[\, 0,1,0 \,]$



The sign of the (0,1,0)th element depends on symmetries S_1 and S_2 and so on $i_1 (+) i_2$

The result is summarized in Table 3.7.

j	j_1	j_2	j_3	b_{ij} or sign of h_{ij}
0	0	0	0	0
1	0	0	1	i_1
2	0	1	0	$i_1 (+)\, i_2$
3	0	1	1	i_2
4	1	0	0	$i_2 (+)\, i_3$
5	1	0	1	$i_1 (+)\, i_2 (+)\, i_3$
6	1	0	1	$i_1 (+)\, i_3$
7	1	1	1	i_3

Table 3.7 : Sign of h_{ij} as a function of i for different j.

Table 3.7 suggests that the sign of the j th element in a Walsh basis vector depends sometimes on one, sometimes on two and sometimes on all the independent dyadic symmetries. The actual relationship can be found by expressing $[j_1 , j_2 , j_3]$ with respect to the new basis S_1, S_2 and S_3. Let the new coordinates of j be

$$r = [r_1 , r_2 , r_3]^t$$

If r_k is 1, then the sign of h_{ij} depends on S_k, and vice versa. Hence, we have

$$j = [j_1 , j_2 , j_3]^t = r_1 * S_1 \ (+) \ r_2 * S_2 \ (+) \ r_3 * S_3 \quad \text{---------(3.10)}$$

Equation 3.10 can be converted into

$$j^t = r^t * \begin{bmatrix} S_1^t \\ S_2^t \\ S_3^t \end{bmatrix}$$

$$r^t = j^t * \begin{bmatrix} S_1^t \\ S_2^t \\ S_3^t \end{bmatrix}^{-1} \quad \text{-----------------------------------(3.11)}$$

Therefore r can be found easily from equation 3.11. The same three cases will be used as examples.

(i) When $[\, j_1, j_2, j_3 \,]$ is $[\, 0,0,1 \,]$

$$[\, r_1, r_2, r_3 \,] = [\, j_1, j_2, j_3 \,] * [\, S \,]^{-1}$$

$$= [\, 0,0,1 \,] * \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}^{-1}$$

As $$[S]^{-1} = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

We have

$$r^t = [\, 0,0,1 \,] * \begin{bmatrix} 0 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

$$= [\, 1,0,0 \,]$$

Thus the (0,0,1)th element only depends on S_1 and so on the first dyadic symmetry.

(ii) When $[\, j_1, j_2, j_3 \,]$ is $[\, 0,1,1 \,]$

$$r^t = [\, 0,1,1 \,] * \begin{bmatrix} 0 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

$$= [\, 0,1,0 \,]$$

Thus the [0,1,1]th element depends only on S_2 and so on the third dyadic symmetry.

(iii) When $[\, j_1, j_2, j_3 \,]$ is $[\, 0,1,0 \,]$

$$r^t = [\, 0,1,0 \,] * \begin{bmatrix} 0 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

$= [\ 1,1,0\ ]$

Thus the (0,1,0)th element depends on the S_1 and S_2 and so on the first and third dyadic symmetries.

Finally, we have equation 3.9

$$\begin{aligned} b_{ij} &= r^t * i \\ &= j^t * [\ S\ ]^{-1} * i \end{aligned}$$

3.4.3 Examples

In this section, the equation derived in the previous section will be used to generate the 8x8 binary Walsh matrices having different orderings. The three most common orderings are given first.

(1) Natural-ordered binary Walsh matrix

The dyadic symmetry matrix of a $2^m \times 2^m$ natural-ordered Walsh matrix is an m by m binary diagonal matrix with diagonal entries equal to 1. Therefore, for the 8 x 8 binary Walsh matrix, the dyadic symmetry matrix is

$$[\ N\ ] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{-----------------------------------}(\ 3.12\ )$$

$$[\ N\ ] = [\ N\ ]^{-1} \quad \text{-----------------------------------}(\ 3.13\ )$$

Hence, we have

$$b_{ij} = [\, j_1 , j_2 , j_3 \,] * \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} * [\, i_1 , i_2 , i_3 \,]^t$$

$$= j_1 * i_1 \ (+) \ j_2 * i_2 \ (+) \ j_3 * i_3 \qquad \text{----------------(3.14)}$$

no. of zero crossings	j	000	001	010	011	100	101	110	111
	r i	000	001	010	011	100	101	110	111
0	000	1	1	1	1	1	1	1	1
7	001	1	-1	1	-1	1	-1	1	-1
3	010	1	1	-1	-1	1	1	-1	-1
4	011	1	-1	-1	1	1	-1	-1	1
1	100	1	1	1	1	-1	-1	-1	-1
6	101	1	-1	1	-1	-1	1	-1	1
2	110	1	1	-1	-1	-1	-1	1	1
5	111	1	-1	-1	1	-1	1	1	-1

Table 3.8 : Natural-ordered binary Walsh matrix generated by dyadic symmetry [4,2,1].

(2) Dyadic-ordered binary Walsh matrix

Generally, the dyadic symmetry matrix of a $2^m \times 2^m$ dyadic-ordered Walsh matrix is an m by m binary matrix containing only opposite diagonal terms equal to 1. Therefore, for the 8 x 8 binary Walsh matrix, the dyadic symmetry matrix is

$$[\, D \,] = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \qquad \text{--------------------------------(3.15)}$$

$$[\, D \,]^{-1} = [\, D \,]$$

$$b_{ij} = [j_1 , j_2 , j_3] * \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} * [i_1 , i_2 , i_3]^t$$

$$= [\, j_3 , j_2 , j_1 \,] * [\, i_1 , i_2 , i_3 \,]^t$$

$$= i_1 * j_3 \ (+) \ i_2 * j_2 \ (+) \ i_3 * j_1 \qquad \text{-----------------(3.16)}$$

no. of zero crossings	j	000	001	010	011	100	101	110	111
	r i	000	100	010	110	001	101	011	111
0	000	1	1	1	1	1	1	1	1
1	001	1	1	1	1	-1	-1	-1	-1
3	010	1	1	-1	-1	1	1	-1	-1
2	011	1	1	-1	-1	-1	-1	1	1
7	100	1	-1	1	-1	1	-1	1	-1
6	101	1	-1	1	-1	-1	1	-1	1
4	110	1	-1	-1	1	1	-1	-1	1
5	111	1	-1	-1	1	-1	1	1	-1

Table 3.9 : Dyadic-ordered binary Walsh matrix generated by dyadic symmetry [1,2,4].

(3) Sequency-ordered binary Walsh matrix

In general, the dyadic symmetry matrix of a $2^m \times 2^m$ sequency-ordered Walsh matrix is an m by m binary matrix with its i th row having i consecutive '1's on the right hand side. Therefore, the dyadic symmetry matrix is

$$[\ Z\ ] = \begin{bmatrix} & & & & & 1 \\ & & & & 1 & 1 \\ & & & 1 & 1 & 1 \\ & & \cdot & \cdot & \cdot & \cdot \\ 1 & \cdot & \cdot & \cdot & 1 & 1 \end{bmatrix} \qquad \text{---------------------(3.17)}$$

and its inverse is

$$[\, Z \,]^{-1} = \begin{bmatrix} & & & & & 1 & 1 \\ & & & & 1 & 1 & \\ & & & 1 & 1 & & \\ & & \cdot & \cdot & & & \\ & 1 & 1 & & & & \\ 1 & 1 & & & & & \\ 1 & & & & & & \end{bmatrix}$$

Hence, we have

$$b_{ij} = [\, j_1, j_2, j_3 \,] * \begin{bmatrix} 0 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} * [\, i_1, i_2, i_3 \,]^t$$

$$= i_1 * j_2 \; (+) \; i_1 * j_3 \; (+) \; i_2 * j_1 \; (+) \; i_2 * j_2 \; (+) \; i_3 * j_1$$

----------------------------(3.18)

no. of zero crossings	j	000	001	010	011	100	101	110	111
	r i	000	100	110	100	011	111	101	001
0	000	1	1	1	1	1	1	1	1
1	001	1	1	1	1	-1	-1	-1	-1
2	010	1	1	-1	-1	-1	-1	1	1
3	011	1	1	-1	-1	1	1	-1	-1
4	100	1	-1	-1	1	1	-1	-1	1
5	101	1	-1	-1	1	-1	1	1	-1
6	110	1	-1	1	-1	-1	1	-1	1
7	111	1	-1	1	-1	1	-1	1	-1

Table 3.10 : Sequency-ordered binary Walsh matrix generated by dyadic symmetry [1,3,7].

Walsh matrices of the above three orderings are symmetric, and this results in several interesting properties. For example, the inverse of a Walsh matrix is equal to itself. It should be noted, however, that the above orderings are not the only ones that result in symmetrical Walsh matrices. For example, the dyadic symmetry matrix

$$[\,s\,] \;=\; \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix}$$

has a symmetric inverse

$$[\,s\,]^{-1} \;=\; \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \text{ ------------------------------ } (\,3.19\,)$$

and so the corresponding Walsh matrix is also symmetrical.

$$b_{ij} \;=\; [\,j_1,j_2,j_3\,] * \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} * \;[\,i_1,i_2,i_3\,]^t$$

$$=\; [\,j_1(+)j_2(+)j_3\,,\; j_1(+)j_2\,,\; j_1\,] * [\,i_1,i_2,i_3\,]^t$$

no. of zero crossings	j	000	001	010	011	100	101	110	111
	r i	000	100	110	010	111	011	001	101
0	000	1	1	1	1	1	1	1	1
1	001	1	1	1	1	-1	-1	-1	-1
2	010	1	1	-1	-1	-1	-1	1	1
3	011	1	1	-1	-1	1	1	-1	-1
5	100	1	-1	-1	1	-1	1	1	-1
4	101	1	-1	-1	1	1	-1	-1	1
7	110	1	-1	1	-1	1	-1	1	-1
6	111	1	-1	1	-1	-1	1	-1	1

Table 3.11 : Binary Walsh matrix generated by binary code and dyadic symmetry [1,3,6].

There are, of course, also Walsh matrices which are not symmetrical.

For example, when the dyadic symmetry matrix is

$$[\, S \,] = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

$$[\, S \,]^{-1} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \quad \text{----------------------------}(3.20)$$

$$b_{ij} = j^{t} * [\, S \,]^{-1} * i$$

$$= [\, j_2 (+) j_3 \,,\ j_2 \,,\ j_1 \,] * i \quad \text{--------------------}(3.21)$$

no. of zero crossings	j	000	001	010	011	100	101	110	111
	r	000	100	110	010	001	101	111	011
	i								
0	000	1	1	1	1	1	1	1	1
1	001	1	1	1	1	-1	-1	-1	-1
3	010	1	1	-1	-1	1	1	-1	-1
2	011	1	1	-1	-1	-1	-1	1	1
4	100	1	-1	-1	1	1	-1	-1	1
5	101	1	-1	-1	1	-1	1	1	-1
7	110	1	-1	1	-1	1	-1	1	-1
6	111	1	-1	1	-1	-1	1	-1	1

Table 3.12 : Binary Walsh matrix generated by dyadic symmetry [1,3,4].

3.5 FURTHER WALSH MATRIX INTERRELATIONSHIPS

Theorem 3.3:

$$i_Z = [Z][D]^{-1} * i_D = [Z][N]^{-1} * i_N$$

$$i_D = [D][Z]^{-1} * i_Z = [D][N]^{-1} * i_N$$

$$i_N = [N][Z]^{-1} * i_Z = [N][D]^{-1} * i_D \quad \text{--------------------}(3.21)$$

where $[Z]$, $[D]$ and $[N]$ are the dyadic symmetry matrices for 2^m by 2^m sequency-ordered, dyadic-ordered and natural-ordered Walsh matrices respectively, and i_Z, i_D and i_N are the corresponding row indices.

Proof :

Equation 3.9

$$b_{ij} = j^t * [S]^{-1} * i \text{ ------------------------(3.9)}$$

can be used to find the relationship between different orderings.

$$j^t *[Z]^{-1} *i_Z = j^t *[D]^{-1} *i_D = j^t *[N]^{-1} *i_N$$

$$\text{for all } j \text{ in } [0,2^m-1] \text{ --------------------------------(3.22)}$$

Equation 3.22 implies

$$[Z]^{-1} *i_Z = [D]^{-1} *i_D = [N]^{-1} *i_N \text{ ------------------------(3.23)}$$

and the conversion equations 3.21 follow. For example, when the block size is 8 x 8 the dyadic symmetry matrices are given in Table 3.13.

dyadic symmetry matrix	matrix	inverse
$[D]$	$\begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$
$[Z]$	$\begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$
$[N]$	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

Table 3.13 : Dyadic symmetry matrices of the three common orderings.

Hence, the conversion matrices will be

conversion matrix	matrix	inverse
$[D]\,[Z]^{-1}$	$\begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}$ binary to gray code	$\begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}$ gray to binary code
$[D]\,[N]^{-1}$	$\begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$ bit reversal	$\begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$ bit reversal
$[Z]\,[N]^{-1}$	$\begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$

Table 3.14 : Matrices for conversion between the three orderings

Let

$$i_Z = [\,z_1, z_2, \ldots, z_m\,]$$
$$i_D = [\,d_1, d_2, \ldots, d_m\,]$$
$$i_N = [\,n_1, n_2, \ldots, n_m\,] \qquad \text{-----------------------}(\ 3.24\ )$$

From Table 3.14, it is obvious that the conversion between i_N and i_D is carried out by bit reversal, and the conversion from i_D to i_Z is gray code-to-binary code conversion.

Furthermore, the conversion from i_N to i_Z can be done easily using the following recursive equations.

$$z_m = n_m$$
$$z_{m-1} = z_m + n_{m-1}$$
$$\cdot\cdot \quad \cdot\cdot \quad \cdot\cdot$$
$$z_1 = z_2 + n_1 \qquad \text{-----------------------------}(\ 3.25\ )$$

It is not necessary to go through the processes of first converting i_N to i_D by bit reversal, and then converting i_D to i_Z by gray code-to binary code conversion. Table 3.15 lists some of the conversions.

i	$i_D = [D][Z]^{-1} * i$	$i_Z = [Z][D]^{-1} * i$	$i_Z = [Z][N]^{-1} * i$
0 000	0 000	0 000	0 000
1 001	1 001	1 001	7 111
2 010	3 011	3 011	3 011
3 011	2 010	2 010	4 100
4 100	6 110	7 111	1 001
5 101	7 111	6 110	6 110
6 110	5 101	4 100	2 010
7 111	4 100	5 101	5 101

Table 3.15 : Conversion of i_Z to i_D , i_D to i_Z and i_N to i_Z

<u>Theorem</u> 3.4:

$$b_{u(+)v,j} = b_{u,j} \ (+) \ b_{v,j} \qquad \text{for all } j \quad \text{-----------}(\ 3.26 \ )$$

$$b_{i,u(+)v} = b_{i,u} \ (+) \ b_{i,v} \qquad \text{for all } j \quad \text{-----------}(\ 3.27 \ )$$

Proof :

$$b_{u,j} \ (+) \ b_{v,j}$$

$$= j^t [S]^{-1} u \ (+) \ j^t [S]^{-1} v$$

$$= r^t u \ (+) \ r^t v \quad \text{where} \quad r^t = j^t [S]^{-1}$$

$$= [r_1 \ \ldots \ r_m] \begin{bmatrix} u_1 \\ . \\ . \\ u_m \end{bmatrix} \ (+) \ [r_1 \ \ldots r_m] \begin{bmatrix} v_1 \\ . \\ . \\ v_m \end{bmatrix}$$

$$= r_1 * u_1 \ (+) \ r_2 * u_2 \ (+) \ \ldots \ (+) \ r_m * u_m \ (+) \ r_1 * v_1 \ (+) \ \ldots \ (+) \ r_m * v_m$$

$$= r_1 * \{u_1 (+) v_1\} \; (+) \; r_2 * \{u_2 (+) v_2\} \; (+) \; \dots . \; r_m * \{u_m (+) v_m\}$$

$$= [r_1 \dots . r_m] \begin{bmatrix} u_1 (+) v_1 \\ \dots\dots . \\ u_m (+) v_m \end{bmatrix}$$

$$= r^t \; \{ u \; (+) \; v \}$$

$$= j^t \; [S]^{-1} \; \{ u \; (+) \; v \}$$

$$= b_{u(+)v,j}$$

$$b_{i,u} \; (+) \; b_{i,v}$$

$$= u^t \; [S]^{-1} \; i \quad (+) \quad v^t \; [S]^{-1} \; i$$

$$= u^t \; r \quad (+) \quad v^t \; r \qquad \text{where} \quad r = [S]^{-1} \; i$$

$$= [\; u_1 \dots u_m \;] \begin{bmatrix} r_1 \\ .. \\ u_m \end{bmatrix} \quad (+) \quad [\; v_1 \dots v_m \;] \begin{bmatrix} r_1 \\ .. \\ v_m \end{bmatrix}$$

$$= u_1 * r_1 \; (+) \; u_2 * r_2 \; (+) \dots (+) \; u_m * r_m \; (+) \; v_1 * r_1 \; (+) \dots (+) \; v_m * r_m$$

$$= \{u_1 (+) v_1\} * r_1 \; (+) \; \{u_2 (+) v_2\} * r_2 \; (+) \; \dots . \; \{u_m (+) v_m\} * r_m$$

$$= \{ \; u \; (+) \; v \; \}^t * r$$

$$= \{ \; u \; (+) \; v \; \}^t \; [S]^{-1} \; i$$

$$= b_{i,u(+)v} \qquad \text{--} (\; 3.28 \;)$$

The above theorem suggests that each row vector (column vector), of [B], can be expressed as the linear combination of m row vectors (column vectors) $B_{i_1}, B_{i_2}, \ldots, B_{i_m}$ where $i_1, i_2, \ldots, i_m$ are independent binary m-vectors.

One significant case occurs when the m independent binary m-vectors are the dyadic symmetries $S_1, S_2, \ldots., S_m$. The m column vectors $B_{S_1}, B_{S_2}, \ldots., B_{S_m}$ are then binary Rademacher functions.

<u>Theorem</u> 3.5 :

The S_k th column vectors, B_{S_k}, of a binary Walsh matrix are the k th binary Rademacher functions where S_k is the k th row of the dyadic symmetry matrix [S] of the binary Walsh matrix.

Proof :

Letting j in equation 3.9 be S_k , we have

$$b_{i,S_k} = S_k^t \; [S]^{-1} \; i \qquad \text{------------------------------------(3.29)}$$

As $[S]\,[S]^{-1} = [I]$, we have

$$S_k^t \; [S] = [\; 0 \ldots 0\; 1\; 0 \ldots 0 \;] \qquad \text{--------------------(3.30)}$$
$$\qquad |\!\leftarrow k \rightarrow\!|$$

Equations 3.29 and 3.30 imply

$$b_{i,S_k} = i_k = \text{ the k th bit of i}$$

and so the theorem is proved.

It is interesting to note that this property may not be valid for the row indices. The Walsh matrix listed in Table 3.12 is an example. However, for symmetrical matrices, this property holds for both row and column indices.

3.6 FAST COMPUTATIONAL ALGORITHMS

3.6.1 Basic theory

Consider the conversion of X into C in equation 3.31 where [H] is a Walsh matrix all of whose basis vectors have the 2^m-1 dyadic symmetry (X and C are 2^m - vectors).

$$C = [H]\ X \qquad\text{--------------------------------------}(\ 3.31\ )$$

Implementation of the conversion process by direct matrix multiplication requires $2^m \times 2^m$ multiplications and $(2^m-1) \times 2^m$ additions (since the [H] matrix consists of only +1 and -1's , we only require $(2^m-1) \times 2^m$ addition or substraction operations to implement). However, it will be shown that by means of 2^{m-1} additions and 2^{m-1} subtractions, the $2^m \times 2^m$ transform can be decomposed into a $2^{m-1} \times 2^{m-1}$ 'even' transform and a $2^{m-1} \times 2^{m-1}$ 'odd' transform by a simple process which makes use of one of the 2^m-1 dyadic symmetries. An eight by eight Walsh matrix will be used to describe the process.

3.6.2 Examples of dyadic symmetry decompositions of an 8x8 sequency-ordered Walsh transform

3.6.2.1 The first dyadic symmetry decomposition

As shown in Table 3.10, H_i for i in [0,3] has even first dyadic symmetry and H_i for i in [4,7] has odd first dyadic symmetry. In general, for a $2^m \times 2^m$ Walsh matrix, the most significant bit, i_1, in the binary representation of $i = [\, i_1, i_2, \ldots, i_m \,]$, indicates whether H_i has even or odd first dyadic symmetry.

$$\begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \\ c_6 \\ c_7 \end{bmatrix} = \begin{array}{c} \begin{array}{cccc} \sqcap & \sqcap & \sqcap & \sqcap^{*1} \end{array} \\ \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \end{bmatrix} \end{array} \times \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} \quad \text{--(3.32)}$$

Therefore, by defining U and V in the following way

$$U = \begin{bmatrix} x_0+x_1 \\ x_2+x_3 \\ x_4+x_5 \\ x_7+x_6 \end{bmatrix} = \begin{bmatrix} u_0 \\ u_1 \\ u_2 \\ u_3 \end{bmatrix} \quad \text{--------------------------(3.33a)}$$

*1 These lines are for identification of the 1st dyadic symmetry.

$$V = \begin{bmatrix} x_0 - x_1 \\ x_3 - x_2 \\ x_4 - x_5 \\ x_7 - x_6 \end{bmatrix} = \begin{bmatrix} v_0 \\ v_1 \\ v_2 \\ v_3 \end{bmatrix} \text{ ------------------------(3.33b)}$$

We have

$$\begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} \text{ x } \begin{bmatrix} u_0 \\ u_1 \\ u_2 \\ u_3 \end{bmatrix}$$

$$= \begin{bmatrix} \text{even} \\ \text{transform} \end{bmatrix} \text{ x } U \text{ -------------------(3.34)}$$

$$\begin{bmatrix} c_4 \\ c_5 \\ c_6 \\ c_7 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} \text{ x } \begin{bmatrix} v_0 \\ v_1 \\ v_2 \\ v_3 \end{bmatrix}$$

$$= \begin{bmatrix} \text{odd} \\ \text{transform} \end{bmatrix} \text{ x } V \text{ ------------------(3.35)}$$

The transform in equation 3.34 (3.35) is said to be 'even' ('odd') because the basis vectors of this tranform are in fact halves of the basis vectors in equation 3.32, that have even (odd) 1st dyadic symmetry. Both the even and odd transforms are 4x4 sequency-ordered Walsh transforms. Computation of C by equation 3.31 requires

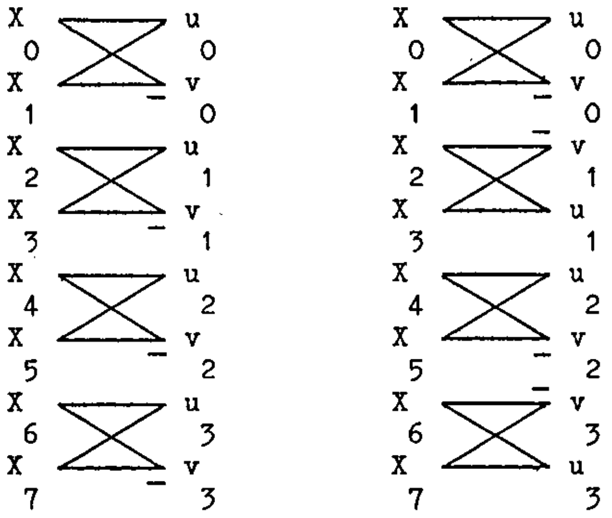


Fig.3.2 Two of the ways of implementing the first dyadic symmetry decomposition

2^m x 2^m or sixty four operations, while by equations 3.33 to 3.35 requires $2x(2^{m-1} x2^{m-1})$ or in this case thirty two operations. Two of the many ways of implementing the first dyadic symmetry decomposition are given in Fig.3.2.

3.6.2.2 The second dyadic symmetry decomposition

As shown in Table 3.10, H_i , i = 0,1,6,7, has even second dyadic symmetry and H_i , i = 2,3,4,5, has odd second dyadic symmetry. In general, a 2^m-dimensional H_i has even (odd) second dyadic symmetry if $i_1 (+) i_2$ is 0 (1) where $i = [\, i_1 , i_2 , \ldots , i_m \,]$.

$$\begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \\ c_6 \\ c_7 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \end{bmatrix} \times \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix}$$

Therefore, by defining U and V in the following way

$$U = \begin{bmatrix} x_0+x_2 \\ x_1+x_3 \\ x_6+x_4 \\ x_7+x_5 \end{bmatrix} = \begin{bmatrix} u_0 \\ u_1 \\ u_2 \\ u_3 \end{bmatrix} \text{---------------------------}(3.36)$$

$$V = \begin{bmatrix} x_0-x_2 \\ x_1-x_3 \\ x_6-x_4 \\ x_7-x_5 \end{bmatrix} = \begin{bmatrix} v_0 \\ v_1 \\ v_2 \\ v_3 \end{bmatrix} \text{---------------------------}(3.37)$$

We have

$$\begin{bmatrix} c_0 \\ c_1 \\ c_6 \\ c_7 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} \times \begin{bmatrix} u_0 \\ u_1 \\ u_2 \\ u_3 \end{bmatrix} \text{---------------}(3.38)$$

$$\begin{bmatrix} c_2 \\ c_3 \\ c_4 \\ c_5 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} \times \begin{bmatrix} v_0 \\ v_1 \\ v_2 \\ v_3 \end{bmatrix} \text{---------------} (3.39)$$

Both the even transform and odd transform are 4x4 sequency Walsh transforms. Two of the many ways of implementing the second dyadic symmetry decomposition are given below:

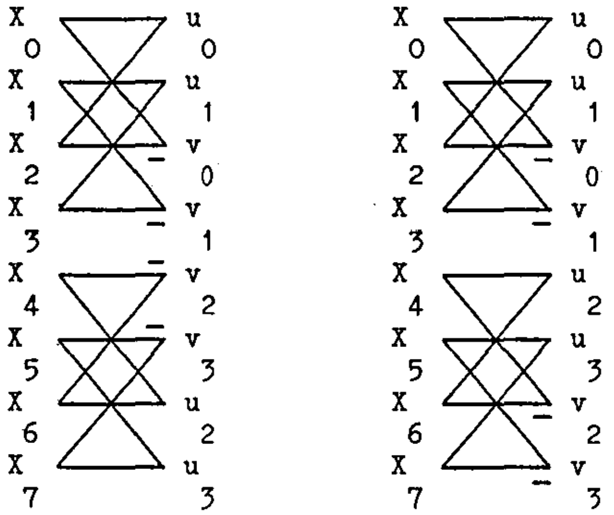


Fig.3.3 Two of the ways of implementing the second dyadic symmetry decomposition.

3.6.2.3 The third dyadic symmetry decomposition

As shown in Table 3.10, H_i, i = 0,1,4,5, has even third dyadic symmetry and H_i, i = 2,3,6,7, has odd third dyadic symmetry. In general, a 2^m-dimensional H_i has even or odd third dyadic symmetry if i_2 is 0 or 1 where $i = [\, i_1, i_2, \ldots, i_m \,]$.

$$\begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \\ c_6 \\ c_7 \end{bmatrix} = \begin{array}{c} \begin{array}{cc} \overbrace{\hphantom{1 \quad 1 \quad 1 \quad 1}} & \overbrace{\hphantom{1 \quad 1 \quad 1 \quad 1}} \end{array} \\ \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \end{bmatrix} \end{array} \times \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix}$$

Therefore, by defining U and V in the following way

$$U = \begin{bmatrix} x_0 + x_3 \\ x_1 + x_2 \\ x_6 + x_5 \\ x_7 + x_4 \end{bmatrix} = \begin{bmatrix} u_0 \\ u_1 \\ u_2 \\ u_3 \end{bmatrix} \text{----------------------------(3.40)}$$

$$V = \begin{bmatrix} x_0 - x_3 \\ x_1 - x_2 \\ x_6 - x_5 \\ x_7 - x_4 \end{bmatrix} = \begin{bmatrix} v_0 \\ v_1 \\ v_2 \\ v_3 \end{bmatrix} \text{----------------------------(3.41)}$$

We have

$$\begin{bmatrix} c_0 \\ c_1 \\ c_4 \\ c_5 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} \times \begin{bmatrix} u_0 \\ u_1 \\ u_2 \\ u_3 \end{bmatrix} \text{-----------------(3.42)}$$

$$\begin{bmatrix} c_2 \\ c_3 \\ c_6 \\ c_7 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} \times \begin{bmatrix} v_0 \\ v_1 \\ v_2 \\ v_3 \end{bmatrix} \text{-----------------} (3.43)$$

Both the even and odd transforms are 4x4 sequency-ordered Walsh transforms. One of the many ways of implementing the third dyadic symmetry decomposition is given below:

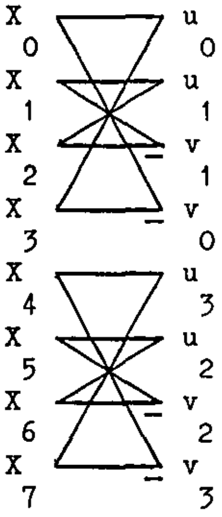


Fig.3.4 One of the ways of implementing the third dyadic symmetry decomposition.

3.6.2.4 The fourth dyadic symmetry decomposition

As shown in Table 3.10, H_i, i = 0,3,4,7, has even fourth dyadic symmetry and H_i, i = 1,2,5,6, has odd fourth dyadic symmetry. In general, a 2^m-dimensional H_i has even (odd) fourth dyadic symmetry if $i_2 (+) i_3$ is 0 (1) where $i = [\, i_1, i_2, \ldots, i_m \,]$.

$$\begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \\ c_6 \\ c_7 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \end{bmatrix} \text{x} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix}$$

Therefore, by defining U and V in the following way

$$U = \begin{bmatrix} x_0+x_4 \\ x_1+x_5 \\ x_2+x_6 \\ x_3+x_7 \end{bmatrix} = \begin{bmatrix} u_0 \\ u_1 \\ u_2 \\ u_3 \end{bmatrix} \text{-----------------------------} (3.44)$$

$$V = \begin{bmatrix} x_0-x_4 \\ x_1-x_5 \\ x_2-x_6 \\ x_3-x_7 \end{bmatrix} = \begin{bmatrix} v_0 \\ v_1 \\ v_2 \\ v_3 \end{bmatrix} \text{-----------------------------} (3.45)$$

We have

$$\begin{bmatrix} c_0 \\ c_3 \\ c_4 \\ c_7 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} \text{x} \begin{bmatrix} u_0 \\ u_1 \\ u_2 \\ u_3 \end{bmatrix} \text{----------------} (3.46)$$

$$\begin{bmatrix} c_1 \\ c_2 \\ c_5 \\ c_6 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} x \begin{bmatrix} v_0 \\ v_1 \\ v_2 \\ v_3 \end{bmatrix} \text{-----------------} (3.47)$$

Both the even transform and odd transforms and 4x4 sequency-ordered Walsh transforms. One of the ways to implement the fourth dyadic symmetry decomposition is given below:

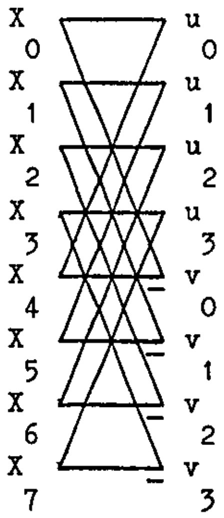
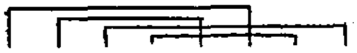


Fig.3.5 One of the ways to implement the fourth dyadic symmetry decomposition.

3.6.2.5 The fifth dyadic symmetry decomposition

As shown in Table 3.10, H_i, i = 0,3,5,6, has even fifth dyadic symmetry and H_i, i = 1,2,4,7, has odd fifth dyadic symmetry. In general, a 2^m-dimensional H_i has even (odd) fifth dyadic symmetry if $i_1 (+) i_2 (+) i_3$ is 0 (1) where $i = [\, i_1, i_2, \dots, i_m \,]$.



$$
\begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \\ c_6 \\ c_7 \end{bmatrix}
=
\begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\
1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\
1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\
1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\
1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \\
1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\
1 & -1 & 1 & -1 & 1 & -1 & 1 & -1
\end{bmatrix}
\times
\begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix}
$$

Therefore, by defining U and V in the following way

$$
U = \begin{bmatrix} x_0 + x_5 \\ x_1 + x_4 \\ x_2 + x_7 \\ x_3 + x_6 \end{bmatrix} = \begin{bmatrix} u_0 \\ u_1 \\ u_2 \\ u_3 \end{bmatrix} \quad \text{------------------------------}(3.48)
$$

$$
V = \begin{bmatrix} x_0 - x_5 \\ x_1 - x_4 \\ x_2 - x_7 \\ x_3 - x_6 \end{bmatrix} = \begin{bmatrix} v_0 \\ v_1 \\ v_2 \\ v_3 \end{bmatrix} \quad \text{------------------------------}(3.49)
$$

We have

$$\begin{bmatrix} c_0 \\ c_3 \\ c_5 \\ c_6 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} \times \begin{bmatrix} u_0 \\ u_1 \\ u_2 \\ u_3 \end{bmatrix} \text{----------------(3.50)}$$

$$\begin{bmatrix} c_1 \\ c_2 \\ c_4 \\ c_7 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} \times \begin{bmatrix} v_0 \\ v_1 \\ v_2 \\ v_3 \end{bmatrix} \text{----------------(3.51)}$$

Both the even transform and odd transforms are 4x4 sequency-ordered Walsh transforms. One of the ways to implement the fifth dyadic symmetry decomposition is given below:

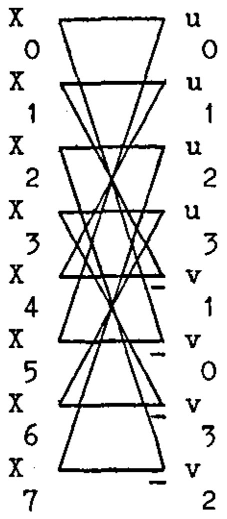


Fig.3.6 One of the ways to implement the fifth dyadic symmetry decomposition

3.6.2.6 The sixth dyadic symmetry decomposition

As shown in Table 3.10, H_i, i = 0,2,5,7, has even sixth dyadic symmetry and H_i, i = 1,3,4,6, has odd sixth dyadic symmetry. In general, a 2^m-dimensional H_i has even (odd) sixth dyadic symmetry if $i_1 (+) i_3$ is 0 (1) where $i = [\, i_1, i_2, i_3 \,]$.

$$\begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \\ c_6 \\ c_7 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \end{bmatrix} \times \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix}$$

Therefore, by defining U and V in the following way

$$U = \begin{bmatrix} x_0+x_6 \\ x_1+x_7 \\ x_2+x_4 \\ x_3+x_5 \end{bmatrix} = \begin{bmatrix} u_0 \\ u_1 \\ u_2 \\ u_3 \end{bmatrix} \text{------------------------------}(3.52)$$

$$V = \begin{bmatrix} x_0-x_6 \\ x_1-x_7 \\ x_2-x_4 \\ x_3-x_5 \end{bmatrix} = \begin{bmatrix} v_0 \\ v_1 \\ v_2 \\ v_3 \end{bmatrix} \text{------------------------------}(3.53)$$

We have

$$\begin{bmatrix} c_0 \\ c_2 \\ c_5 \\ c_7 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} x \begin{bmatrix} u_0 \\ u_1 \\ u_2 \\ u_3 \end{bmatrix} \text{----------------}(3.54)$$

$$\begin{bmatrix} c_1 \\ c_3 \\ c_4 \\ c_6 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} x \begin{bmatrix} v_0 \\ v_1 \\ v_2 \\ v_3 \end{bmatrix} \text{----------------}(3.55)$$

Both the even and odd transforms are 4x4 sequency-ordered Walsh transforms. One of the ways to implement this decomposition is given below:

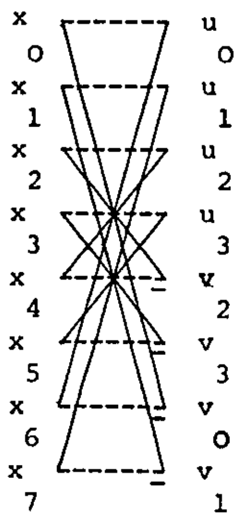


Fig.3.7 One of the ways to implement the sixth dyadic symmetry decomposition.

3.6.2.7 The seventh dyadic symmetry decomposition

As shown in Table 3.10, H_i, i = 0,2,4,6, has even seventh dyadic symmetry and H_i, i = 1,3,5,7, has odd seventh dyadic symmetry. In general, a 2^m-dimensional H_i has even (odd) seventh dyadic symmetry if i_3 is 0 (1) where $i = [\, i_1, i_2, \ldots, i_m \,]$.

$$\begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \\ c_6 \\ c_7 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \end{bmatrix} \times \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix}$$

Therefore, by defining U and V in the following way

$$U = \begin{bmatrix} x_0+x_7 \\ x_1+x_6 \\ x_2+x_5 \\ x_3+x_4 \end{bmatrix} = \begin{bmatrix} u_0 \\ u_1 \\ u_2 \\ u_3 \end{bmatrix} \text{------------------------------}(3.56)$$

$$V = \begin{bmatrix} x_0-x_7 \\ x_1-x_6 \\ x_2-x_5 \\ x_3-x_4 \end{bmatrix} = \begin{bmatrix} v_0 \\ v_1 \\ v_2 \\ v_3 \end{bmatrix} \text{------------------------------}(3.57)$$

We have

$$\begin{bmatrix} c_0 \\ c_2 \\ c_4 \\ c_6 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} x \begin{bmatrix} u_0 \\ u_1 \\ u_2 \\ u_3 \end{bmatrix} \text{----------------(3.58)}$$

$$\begin{bmatrix} c_1 \\ c_3 \\ c_5 \\ c_7 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} x \begin{bmatrix} v_0 \\ v_1 \\ v_2 \\ v_3 \end{bmatrix} \text{----------------(3.59)}$$

Both of the even and odd transforms are 4x4 sequency-ordered Walsh transforms. One of the ways to implement the seventh dyadic symmetry decomposition is given below:

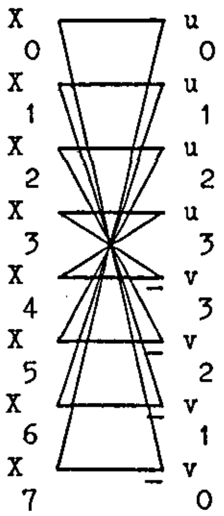


Fig.3.8 One of the ways to implement the seventh dyadic symmetry decomposition.

3.6.3 Fast computational algorithms from dyadic symmetry decomposition

Dyadic symmetry decomposition can convert a 2^m-order Walsh transform into two 2^{m-1}-order Walsh transforms using 2^m addition and subtraction operations. With another two decompositions (each requires 2^{m-1} addition and subtraction operations), the two 2^{m-1}-order Walsh transforms can be converted into four 2^{m-2}-order Walsh transforms. Therefore, m applications of the decomposition can complete the 2^m-order Walsh transform by reducing the order by half each time to 1 with $m \times 2^m$ addition and subtraction operations.

It has been shown that there are 2^m-1 kinds of dyadic symmetry decompositions which can convert a 2^m-order Walsh transform to two 2^{m-1}-order Walsh transforms. Furthermore, there is more than one way to implement a dyadic symmetry decomposition. This implies there are indeed numerous ways to implement the m decompositions. However, some of the ways are more straightforward than others.

Fig.3.9 shows how an eight by eight Walsh transform of the vector X is accomplished by the repeated use of the 2^{m-1}th dyadic symmetry decomposition. From the signal flow diagram, it can be seen that the fast computational algorithm using the 2^{m-1}th dyadic symmetry decomposition is very close to that of the Cooley-Tukey algorithm for the FFT. Indeed, the final result also requires to be re-ordered in the same way as that of the Cooley-Tukey algorithm to convert it to sequency ordering.

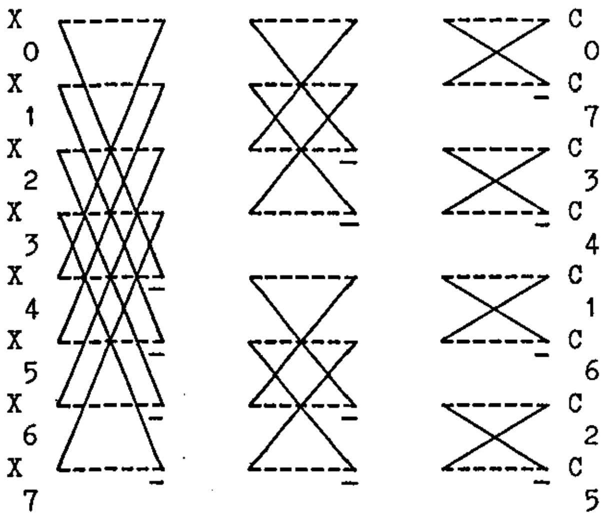


Fig.3.9 Fast algorithm usingthe 2^{m-1} th dyadic symmetry decomposition

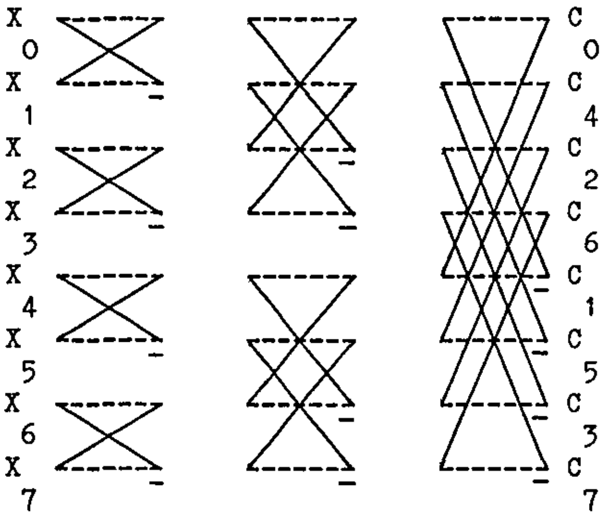


Fig.3.10 Fast algorithm using the first dyadic symmetry decomposition.

Fig.3.10 shows how the 8-order Walsh transform of vector X is accomplished by the repetitive use of the first dyadic symmetry decomposition. An attractive feature of this fast algorithm is that j in C_j is simply the bit reversal of i in X_i.

Fig.3.11 shows how an 8-order Walsh transform of vector X is accomplished by the repetitive use of the 2^m-1 th dyadic symmetry decomposition of X. Fig.3.12 shows an alternative way of implementing this fast algorithm. The relation between i in X_i and j in C_j is simply that of bit reversal.

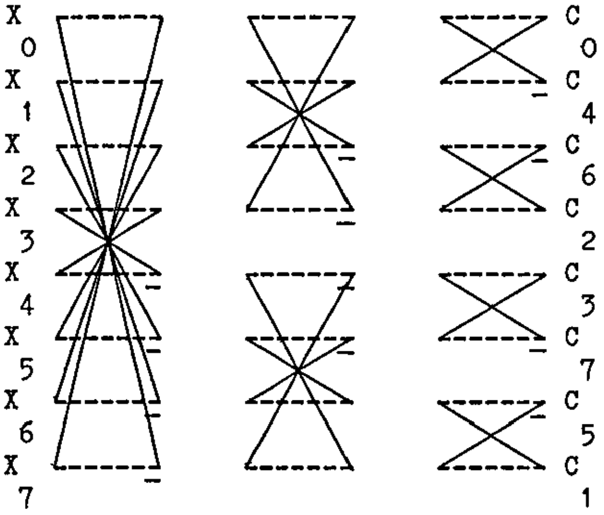


Fig.3.11 Fast algorithm using the 2^m-1 th dyadic symmetry decomposition

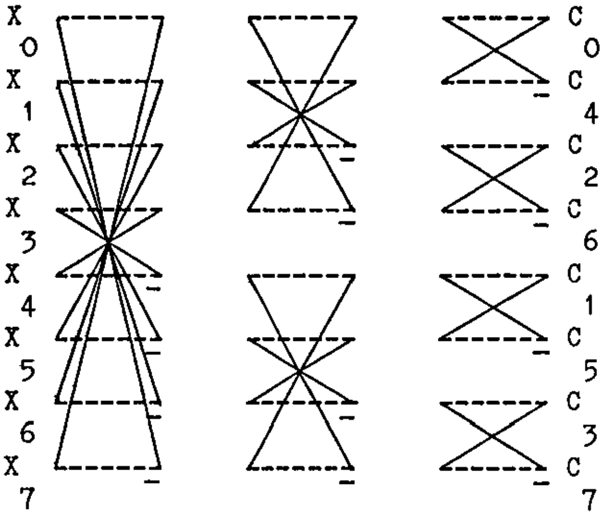


Fig.3.12 Fast algorithm using the 2^m-1 th dyadic symmetry decomposition

3.7 CONCLUSIONS

A unified matrix treatment for the Walsh transform using the concept of dyadic symmetry has been presented. This unified treatment allows a straightforward derivation of a simple equation for the generation of Walsh matrices of different orderings, various re-ordering schemes and various fast computational algorithms. It is believed that this will provide a better understanding of the Walsh transform and hence, allow further fast computational algorithms and new properties to be found. The whole theory relates to a binary field with 'logical and' and modulo two addition as operations and thus allows both the generation of Walsh matrices of different orderings, and re-ordering schemes, to be implemented using simple logic circuits.

3.8 NOTE ON PUBLICATION

A paper entitled 'Dyadic symmetry and its application to Walsh transform theory' was submitted to IEEE transactions on Electromagnetic Compatibility in 1983. This paper was jointly authored with R.J.Clarke.

CHAPTER FOUR

NEW TRANSFORMS

4.1 INTRODUCTION

Transform coding of image data is a topic which has been extensively investigated over the past 10-15 years, and it has been shown to be an efficient technique for low bit rate image representation. The effectiveness of transform coding is mainly due to the transformation which packs energy into the low sequency coefficients. The first transform chosen to accomplish this task was the discrete Fourier transform (DFT) which was reported to produce good results [99].

In contrast to the other main contender for image compression, predictive coding, transform coding suffers from the requirement of a high degree of processing sophistication. Therefore, there was a search for a simple transform which would ease this problem. This led to the application of the Walsh transform [101] which requires only additions and subtractions. Later, more transformations were proposed with either lower computational requirement or better performance. Some, like the Haar transform, aim at simplicity whilst others, like the slant transform and the discrete cosine transform (DCT), aim at better energy packing ability.

Although there should be a trade-off between performance and simplicity, continuing developments in semiconductor technology have convinced most researchers that favour should be given to performance. This is why the discrete cosine transform, which has been shown to be asymtotically close to the optimal KLT, has attracted much attention even though it is more complicated than the Walsh transform. In practice, however, for transform coding of moving pictures in real time, simplicity of the transformation is still desirable. This is why a real time digital image coding system reported recently still adopts the Walsh transform [47]. In view of performance and ease of implementation, the choice of the transformations lies very much between the DCT and the Walsh transform depending on whether signal processing speed is paramount.

In this chapter, two new transforms are proposed, which can be used as substitutes for the Walsh transform. The new transforms have virtually the same complexity and computational requirement as the Walsh transform. They employ additions, subtractions and binary shifts only but have improved efficiencies (equation 4.5), defined in terms of the ability of the transform to decorrelate signal elements by converting them to transform coefficients. The efficiencies of the two new transforms both lie between that of the Walsh transform and that of the DCT for moderate block sizes.

The two approaches used to generate the new transforms will be described in the next two sections, 4.2 and 4.3. The first approach is to seek basis vectors which, whilst still satisfying the criteron of orthogonality, correlate well with commonly occuring image vectors, and yet allow easy implementation. The second is to employ computer search techniques based solely upon the transform efficiency criteria mentioned earlier. The first approach results a transform called the high correlation transform or HCT. The second approach results in two transforms, one of which is again the HCT, and another termed the low correlation transformation or LCT. Section 4.4 shows how these two transforms, originally generated using a matrix size 8x8, may be determined in the general case of dimension $2^m \times 2^m$, where m is an integer.

In section 4.5, the new transforms will be compared with other well known transforms, including the DCT, slant, Haar and Walsh transform, for different block sizes. Two types of tests were carried out, one on the one-dimensional first-order Markov process using efficiency and energy packing ability as criteria, the other using two-dimensional transforms on real images using normalized mean square error (NMSE) defined in equation 4.6 as the criterion.

Finally, the implementation of the LCT and HCT will be described in section 4.6. Description will first be given of the implementation of the real orthonormal transforms LCT and HCT using the unnormalized LCT (ULCT) and unnormalized HCT (UHCT), then fast computational algorithms to implement the two unnormalized transforms are described.

4.2 CREATION OF THE HIGH CORRELATION TRANSFORM (HCT)

4.2.1 Basic principle

Transform coefficient c_i is the scalar product of the basis vector

T_i and the signal vector. Hence, a close resemblance of the basis vector T_i and the signal vector results in a large magnitude of c_i. A good transform, such as the DCT, packs most energy into the low sequency coefficients. This implies that low sequency basis vectors of a good image transform always resemble slowly varying image signal vectors. For example, as shown in Fig.4.1, the low sequency basis vectors of the 8 by 8 DCT change smoothly, whilst the Walsh basis vectors have sudden jumps between the positive and negative elements. This is why the DCT can pack more energy into the low sequency coefficients.

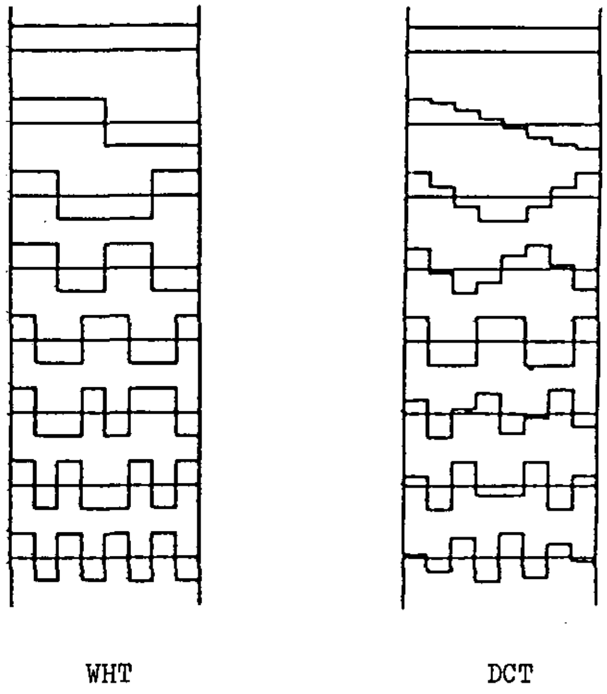


Fig.4.1 The basis vectors of the 8x8 Walsh transform (WHT) and the discrete cosine transform (DCT).

The new transforms are obtained by a technique which can replace pairs of Walsh basis vectors by others to form a new set of linearly independent basis vectors. For clarity, the technique is explained by the following example which replaces the (000)th and (100)th eight-dimensional Walsh basis vectors.

By means of theorem 3.2, it can be seen that the (000)th and (100)th Walsh basis vectors are orthogonal to all other Walsh basis vectors because of dyadic symmetries (011) and (111). Therefore, a new set of linearly independent basis vectors can be obtained, if the (000)th and (100)th Walsh basis vectors are replaced by a pair of linearly independent vectors which have the (011)rd and (111)th dyadic symmetries but without the (001)st dyadic symmetry. For example the following two pairs of linearly independent vectors satisfy this requirement.

$$[\ 1\ \ a\ \ a\ \ 1\ \ 1\ \ a\ \ a\ \ 1\ ] \quad \text{and} \quad [\ a\ -1\ -1\ \ a\ \ a\ -1\ -1\ \ a\ ]$$

$$[\ a\ \ 1\ \ 1\ \ a\ \ a\ \ 1\ \ 1\ \ a\ ] \quad \text{and} \quad [\ 1\ -a\ -a\ \ 1\ \ 1\ -a\ -a\ \ 1\ ]$$

Table 4.1 lists all the vectors which are without one of the three independent dyadic symmetries. Also, Walsh vectors 001 and 101, 010 and 110, 011 and 111 all have the same (011)rd and (111)th but opposite (001)st dyadic symmetries. Hence, all these pairs can be replaced by vector pairs which have the (011)rd and (111)th, but lack the (001)st dyadic symmetries. Similarly, other pairs of Walsh basis vectors can be replaced by vector pairs without one of the other two dyadic symmetries, (011) and (111), in the ways shown in Table 4.2.

i	vectors without the dyadic symmetry		
	(001)	(011)	(111)
0	a 1 1 a a 1 1 a or 1 a a 1 1 a a 1	a a 1 1 1 1 a a or 1 1 a a a a 1 1	a a a a 1 1 1 1 or 1 1 1 1 a a a a
1	a 1 1 a-a-1-1-a or 1 a a 1-1-a-a-1	a a 1 1-1-1-a-a or 1 1 a a-a-a-1-1	a a a a-1-1-1-1 or 1 1 1 1-a-a-a-a
2	a 1-1-a-a-1 1 a or 1 a-a-1-1-a a 1	a a-1-1-1-1 a a or 1 1-a-a-a-a 1 1	a a-a-a-1-1 1 1 or 1 1-1-1-a-a a a
3	a 1-1-a a 1-1-a or 1 a-a-1 1 a-a-1	a a-1-1 1 1-a-a or 1 1-a-a a a-1-1	a a-a-a 1 1-1-1 or 1 1-1-1 a a-a-a
4	a-1-1 a a-1-1 a or 1-a-a 1 1-a-a 1	a-a-1 1 1-1-a a or 1-1-a a a-a-1 1	a-a-a a 1-1-1 1 or 1-1-1 1 a-a-a a
5	a-1-1 a-a 1 1-a or 1-a-a 1-1 a a-1	a-a-1 1-1 1 a-a or 1-1-a a-a a 1-1	a-a-a a-1 1 1-1 or 1-1-1 1-a a a-a
6	a-1 1-a-a 1-1 a or 1-a a-1-1 a-a 1	a-a 1-1-1 1-a a or 1-1 a-a-a a-1 1	a-a a-a-1 1-1 1 or 1-1 1-1-a a-a a
7	a-1 1-a a-1 1-a or 1-a a-1 1-a a-1	a-a 1-1 1-1 a-a or 1-1 a-a a-a 1-1	a-a a-a 1-1 1-1 or 1-1 1-1 a-a a-a

Table 4.1 : The vectors which lack one of the 3 independent dyadic symmetries. 'i' is the number of zero crossings and 'a' is a positive constant less than unity.

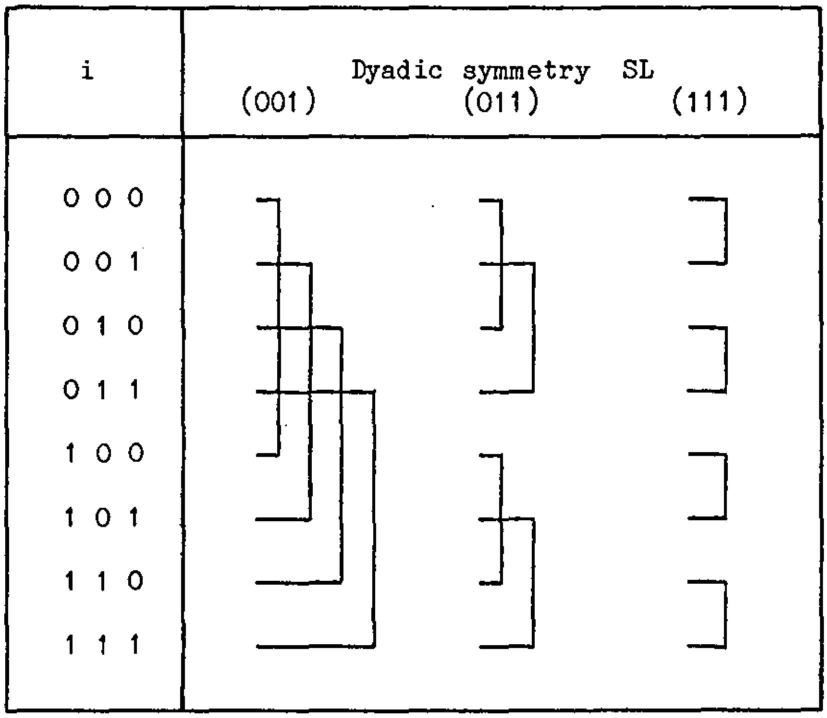


Table 4.2 : The pairs of eight dimensional Walsh basis vectors which have all dyadic symmetries the same except for dyadic symmetry SL (symmetry lacking).

4.2.2 Generation of the 8 by 8 High Correlation Transform

In this section, each of the 8 Walsh basis vectors H_i is examined in turn and modified to form vector T_i according to the principles described in the previous section to generate a new transform similar to the DCT.

(i) $H_0 = [\ 1\ \ 1\ \ 1\ \ 1\ \ 1\ \ 1\ \ 1\ \ 1\ ]$

This vector is exactly the same as that of the DCT, hence, T_0 is taken as H_0.

(ii) $$H_1 = [\ 1 \quad 1 \quad 1 \quad 1 \ -1 \ -1 \ -1 \ -1 \ ]$$

There is a sudden change between the fourth and the fifth element in H_1. It can be smoothed out by replacing H_1 by T_1 which is the vector without the (011)rd dyadic symmetry and is given as follows.

$$T_1 = [\ 1 \quad 1 \quad a \quad a \ -a \ -a \ -1 \ -1 \ ]$$

'a' is a positive constant less than unity. T_1 has even (001)st and odd (111)th dyadic symmetry but lacks the (011)rd dyadic symmetry. In order to maintain linear independence, H_3 has to be replaced by T_3 (Table 4.2).

$$T_3 = [\ a \quad a \ -1 \ -1 \quad 1 \quad 1 \ -a \ -a \ ]$$

Both T_1 and T_3 have even (001)st and odd (111)th symmetry but lack the (011)rd dyadic symmetry, and are orthogonal to each other.

(iii) $$H_2 = [\ 1 \quad 1 \ -1 \ -1 \ -1 \ -1 \quad 1 \quad 1 \ ]$$

The changes between the second and the third, and between the sixth and the seventh elements could be reduced by replacing H_2 by T_2 which has odd (011)rd, even (111)th dyadic symmetry, but lacks the (001)st dyadic symmetry.

$$T_2 = [\ 1 \quad a \ -a \ -1 \ -1 \ -a \quad a \quad 1 \ ]$$

To maintain linear independence, H_6 has to be replaced by

$$T_6 = [\ a \ -1 \quad 1 \ -a \ -a \quad 1 \ -1 \quad a \ ]$$

(iv) $H_4 = [\ 1\ -1\ -1\ \ 1\ \ 1\ -1\ -1\ \ 1\ ]$

Since H_4 is exactly the same as in the DCT, no modification is required, and T_4 is set equal to H_4 .

(v) $H_5 = [\ 1\ -1\ -1\ \ 1\ -1\ \ 1\ \ 1\ -1\ ]$

$H_7 = [\ 1\ -1\ \ 1\ -1\ \ 1\ -1\ \ 1\ -1\ ]$

In order to make these two vectors resemble those of the DCT, they are replaced by T_5 and T_7 which are vectors of the odd (001)st and (111)th dyadic symmetries but without the (011)rd dyadic symmetry.

$T_5 = [\ 1\ -1\ -a\ \ a\ -a\ \ a\ \ 1\ -1\ ]$

$T_7 = [\ a\ -a\ \ 1\ -1\ \ 1\ -1\ \ a\ -a\ ]$

The results are summarized in Table 4.3.

T_i	dyadic symmetry		
	001	011	111
0	0	0	0
1	0	x	1
2	x	1	0
3	0	x	1
4	1	0	0
5	1	x	1
6	x	1	0
7	1	x	1

Table 4.3 : Dyadic symmetry within the basis vectors of the new transform HCT . x stands for absence of such symmetry.

Finally, each T_i, being divided by a constant k_i, is normalized to form the basis vectors of the HCT where

$$k_i^2 = \sum_{j=0}^{n-1} t_{ij}^2 \qquad \text{------------------------------(4.1)}$$

and t_{ij} is the j th element of T_i.

4.2.3 Optimum value for the constant 'a'

The previous discussion suggests that the constant 'a' should be positive and less than unity, and its exact value should maximize the ability of the HCT to transform a typical image signal vector X into a vector Y of uncorrelated elements. This ability may be measured by the transform efficiency η defined on a one-dimensional first order Markov process of adjacent element correlation ρ. The larger the efficiency, the greater is the ability of the transform to convert X into a set of uncorrelated elements. A formal definition of transform efficiency is given in section 4.3.1. Furthermore, for easy implementation, 'a' should equal the inverse of an integer. Fig.4.2 shows the dependence of the transform efficiency on the constant 'a' for different values of adjacent element correlation. Most of the curves have maxima when 'a' is equal to 1/2, which can conveniently be implemented by a simple right binary shift.

4.3 COMPUTER SEARCH FOR HIGH EFFICIENCY TRANSFORMS

In this section, a more objective approach will be used to determine the best transformation. Transform efficiency is used as a criteron

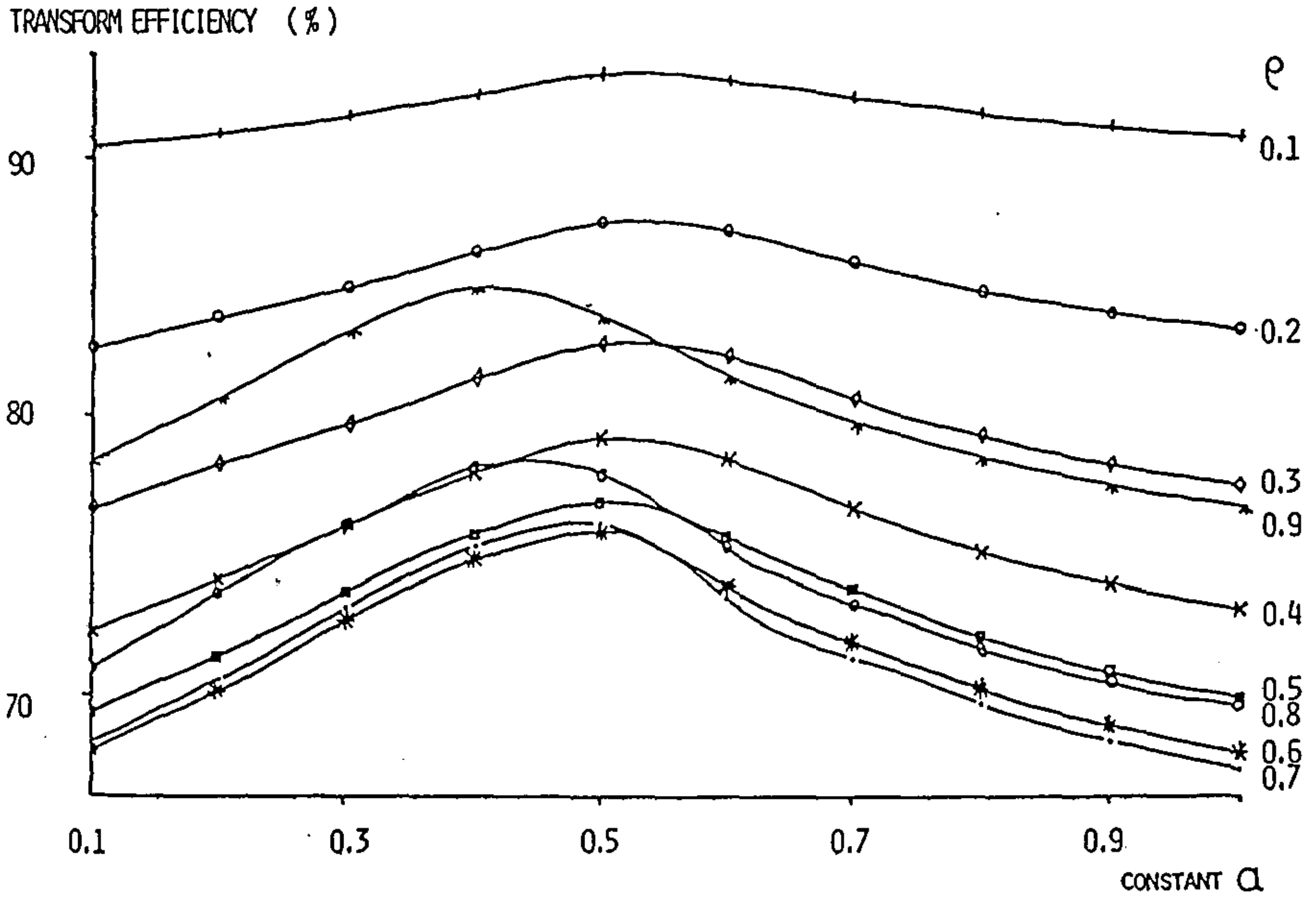


Fig.4.2 The dependence of transform efficiency on the constant 'a' for different values of adjacent element correlation ρ .

and a computer search is carried out to find the transforms of the highest efficiency. A formal definition of transform efficiency is given in section 4.3.1. The magnitude of the transform efficiency indicates the capability of the transform to convert a signal vector into an uncorrelated transform coefficient vector.

4.3.1 Transform efficiency

Let the n-dimensional vector X be a sample from a one-dimensional, zero mean, unit-variance first-order Markov process with adjacent element correlation ρ , and covariance matrix, [CX], where

$$[CX] = E\,[X\,X^t] = \begin{bmatrix} 1 & \rho & \rho^2 & & & & \rho^{n-1} \\ \rho & 1 & \rho & \ddots & & & \\ \rho^2 & \rho & 1 & \rho & & & \\ & \ddots & \rho & 1 & \rho & & \\ & & \ddots & \ddots & \ddots & \ddots & \rho^2 \\ & & & \ddots & \rho & 1 & \rho \\ \rho^{n-1} & & & & \rho^2 & \rho & 1 \end{bmatrix} \qquad (4.2)$$

and E [ ] denotes expected value.

The efficiency of a transform [T] is defined on the transform domain covariance matrix [CC] of vector C where

$$\begin{aligned} C &= [T]\,X \qquad \text{--------------------}(4.3) \\ [CC] &= E\,[\,C\,C^t\,] \\ &= [T]\,[CX]\,[T]^t \end{aligned}$$

$$= \begin{bmatrix} s_{11} & \cdots\cdots & s_{1n} \\ \cdots & \cdots\cdots & \cdots \\ \cdots & \cdots\cdots & \cdots \\ s_{n1} & \cdots\cdots & s_{nn} \end{bmatrix} \qquad \text{--------(4.4)}$$

$$\text{Efficiency } \eta = \frac{\sum_{i=1}^{n} | s_{ii} |}{\sum_{p=1}^{n} \sum_{q=1}^{n} | s_{pq} |} \times 100\% \qquad \text{---------(4.5)}$$

The larger is η, the greater is the ability of [T] to transform X into a vector Y of uncorrelated elements.

4.3.2 Experimental procedures

New transforms can be formed by replacing some pairs of the Walsh basis vectors by other pairs lacking one of the independent dyadic symmetries. A computer program was generated to form all the possible pair combinations and then to compute the corresponding transform efficiency for

(i) different values of the constant 'a', and

(ii) different values of ρ.

The value of 'a' to be tested are 0.0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8 and 0.9. The adjacent element correlations to be considered are 0.1, 0.3, 0.5, 0.7 and 0.9.

4.3.3 Results and discussion

It is found that two ways of pairing are of particular interest. As shown in Table 4.4, one of them, denoted HC, has the same combinations as the HCT, which is the transform designed to resemble the DCT in section 4.2. The other, denoted LC, is generated by destroying the first dyadic symmetry of all the Walsh basis vectors except the pair containing the dc vector. The results are summarized in Table 4.5. It can be seen that

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & a & a & -a & -a & -1 & -1 \\ 1 & a & -a & -1 & -1 & -a & a & 1 \\ a & a & -1 & -1 & 1 & 1 & -a & -a \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & -a & a & -a & a & 1 & -1 \\ a & -1 & 1 & -a & -a & 1 & -1 & a \\ a & -a & 1 & -1 & 1 & -1 & a & -a \end{bmatrix}$$

a) The combination HC.

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ a & 1 & 1 & a & -a & -1 & -1 & -a \\ 1 & a & -a & -1 & -1 & -a & a & 1 \\ 1 & a & -a & -1 & 1 & a & -a & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -a & -a & 1 & -1 & a & a & -1 \\ a & -1 & 1 & -a & -a & 1 & -1 & a \\ a & -1 & 1 & -a & a & -1 & 1 & -a \end{bmatrix}$$

b) The combination LC.

Table 4.4 : The two combinations which have optimum performance among all the combinations generated during the computer search.

(1) For adjacent element correlations 0.1, 0.3, 0.7, the transform efficiency is highest when the constant, a, is 0.5. For adjacent element correlations 0.5 and 0.9, the efficiency is highest when 'a' equals 0.6 and 0.4 respectively.

(2) For adjacent element correlations 0.1, 0.3 and 0.5, the combination LC with the constant equal to 0.5 has very high efficiency.

(3) For adjacent element correlations 0.5, 0.7 and 0.9, the combination HC with the constant equal to 0.5 has very high efficiency.

The implementation of a binary multiplication by one-half is simply a right shift. Therefore, for both simplicity and performance, we may conclude that among all the possible transforms that could be

ρ	FIRST TRANSFORM	SECOND TRANSFORM	EFFICIENCY OF THE DCT & WHT
0.1	LC $a=0.5$ 94.23	———	94.54 90.95
0.3	LC $a=0.5$ 84.72	———	87.11 77.73
0.5	LC $a=0.6$ 77.53	HC $a=0.5$ LC $a=0.5$ 77.07	83.14 69.75
0.7	HC $a=0.5$ 76.32	———	82.86 67.23
0.9	HC $a=0.4$ 88.20	HC $a=0.5$ 84.09	89.83 77.13

Table 4.5

The transforms that have the highest efficiency for different values of adjacent element correlation ρ.

created using the technique described in section 4.2.1, the LC combination with the constant equal to 0.5 is the best transform for low correlation data, whilst the HC combination with the constant equal to 0.5 is the best transform for high correlation data. Combinations LC and HC with constant equal to 0.5 are termed, respectively, the Low Correlation Transform (LCT) and the High Correlation Transform (HCT).

4.4 The HCT AND LCT FOR BLOCK SIZE $2^m \times 2^m$

4.4.1 The HCT for block size $2^m \times 2^m$

The 2^m Walsh basis vectors H_i can be modified according to the following rules to form a 2^m-order HCT whose basis vectors resemble those of the DCT.

(i) Solve the equation Mod $[i-2^{t-1},2^t] = 0$ for all the i (Table 4.6 lists the solution for $i \in [0,15]$).

(ii) Replace H_i by the vector without the $2^{m-t}-1$ th dyadic symmetry and whose first element is 1 if mod$[i,2\text{x}2^t] = 2^{t-1}$, and $\frac{1}{2}$ if mod$[i,2\text{x}2^t] = 3\text{x}2^{t-1}$.

(iii) The remaining two basis vectors H_0 and H_{2^m-1} remain unchanged where t is in $[1,m-1]$.

Finally, the basis vectors are obtained after being normalized as given by equation 4.1. As an example, the value of t, the dyadic symmetry to be destroyed and the first element of vector T_i of a 16x16 HCT are given in Table 4.6. The basis vectors for the 4x4, 8x8 and 16x16 HCT are given in Fig.4.3.

i	first element	t	dyadic symmetry to be destroyed
0	1	-	-
1	1	1	7
2	1	2	3
3	1/2	1	7
4	1	3	1
5	1	1	7
6	1/2	2	3
7	1/2	1	7
8	1	-	-
9	1	1	7
10	1	2	3
11	1/2	1	7
12	1/2	3	1
13	1	1	7
14	1/2	2	3
15	1/2	1	7

Table 4.6 The values of the first element, t, and dyadic symmetry to be destroyed corresponding to each value of i for a 16-order HCT.

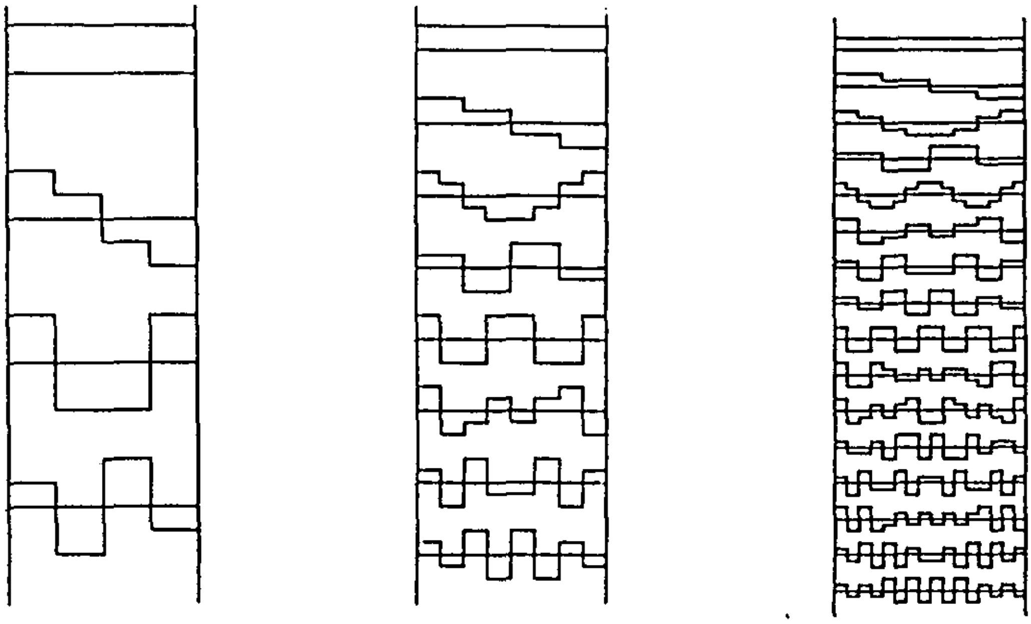


Fig.4.3 Basis vectors of the 4-order, 8-order and 16-order HCT.

4.4.2 The LCT of block size 2^m by 2^m

Let the i th basis vector of the LCT be T_i where i is in $[0,2^m-1]$ and T_0 and $T_{2^{m-1}}$ are the Walsh basis vectors having zero and 2^{m-1} zero crossings. For the other values of i, T_i is the vector which satisfies the following conditions.

(i) T_i has all but the first dyadic symmetry.

(ii) The first element of T_i is 1 if $i \varepsilon [\, 2^{m-2}, 3x2^{m-2} - 1 \,]$, otherwise it is 1/2.

Finally, the basis vectors of the LCT are obtained after being normalized by the process given by equation 4.1. The basis vectors for the 4-order ,8-order and 16-order LCT are shown in Fig.4.4.

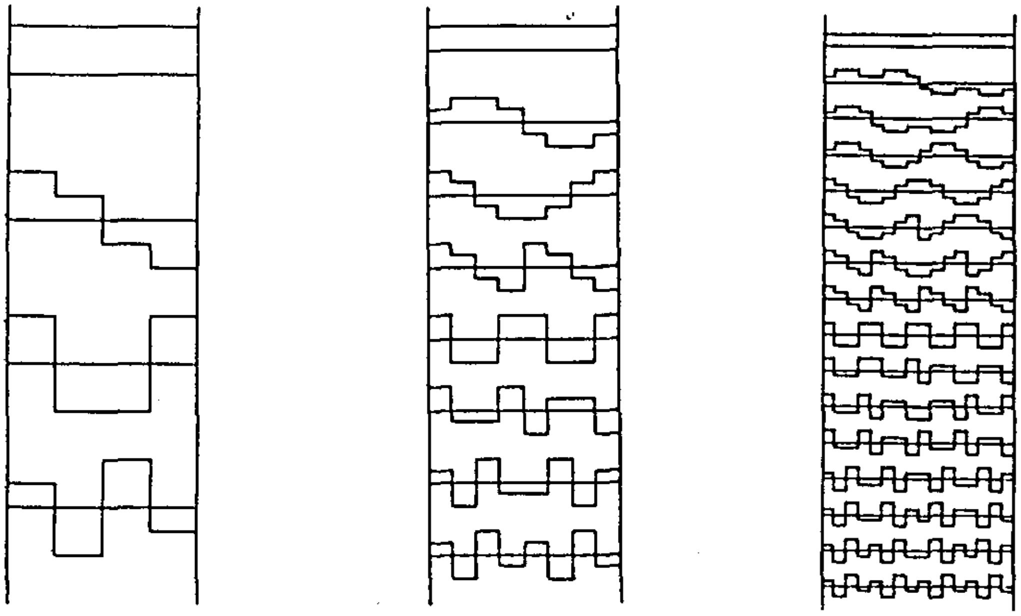


Fig.4.4 Basis vectors of the 4-order, 8-order and 16-order LCT.

4.5 PERFORMANCE OF THE HCT AND LCT

4.5.1 Tests on the one-dimensional first order Markov process

Using the Markov process of section 4.3.1,

$$\text{let} \qquad PER_r = \frac{\sum_{i=0}^{r} s_{ii}^2}{\sum_{j=0}^{n-1} s_{jj}^2} \times 100\% \qquad \text{----------(4.6)}$$

PER_r is the percentage of energy that is packed into the first r+1 transform coefficients. Figs. 4.5 to 4.7 show the energy packing ability of the DCT, the Walsh transform,the HCT and LCT with adjacent element correlation equal to 0.9, for block sizes 4x4, 8x8 and 16x16.

The results show that, a) for the same number of coefficients, the DCT

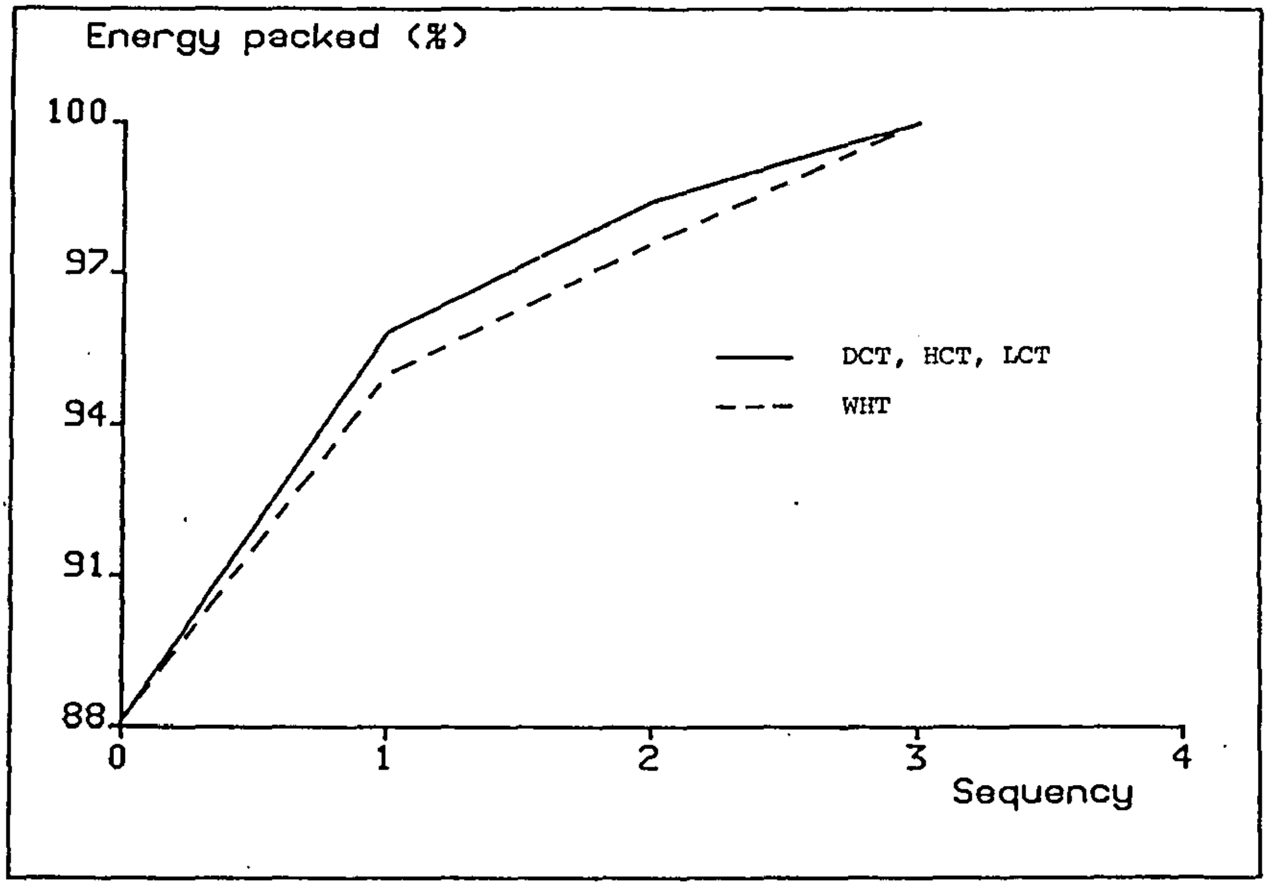


Fig. 4.5 Energy packing ability of the 4-order DCT, HCT, LCT and WHT vs number of retained coefficients.

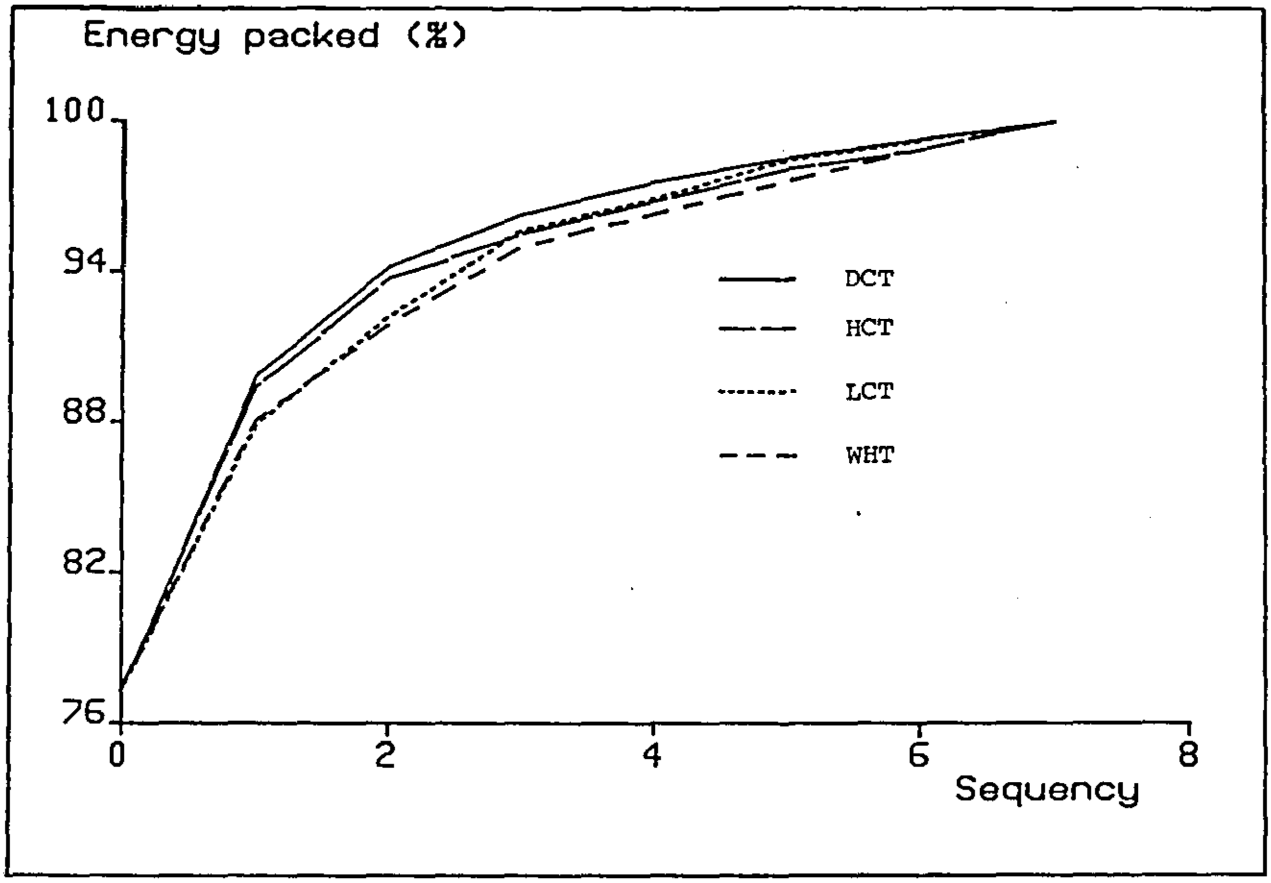


Fig. 4.6 Energy packing ability of the 8-order DCT, HCT, LCT and WHT vs number of retained coefficients.

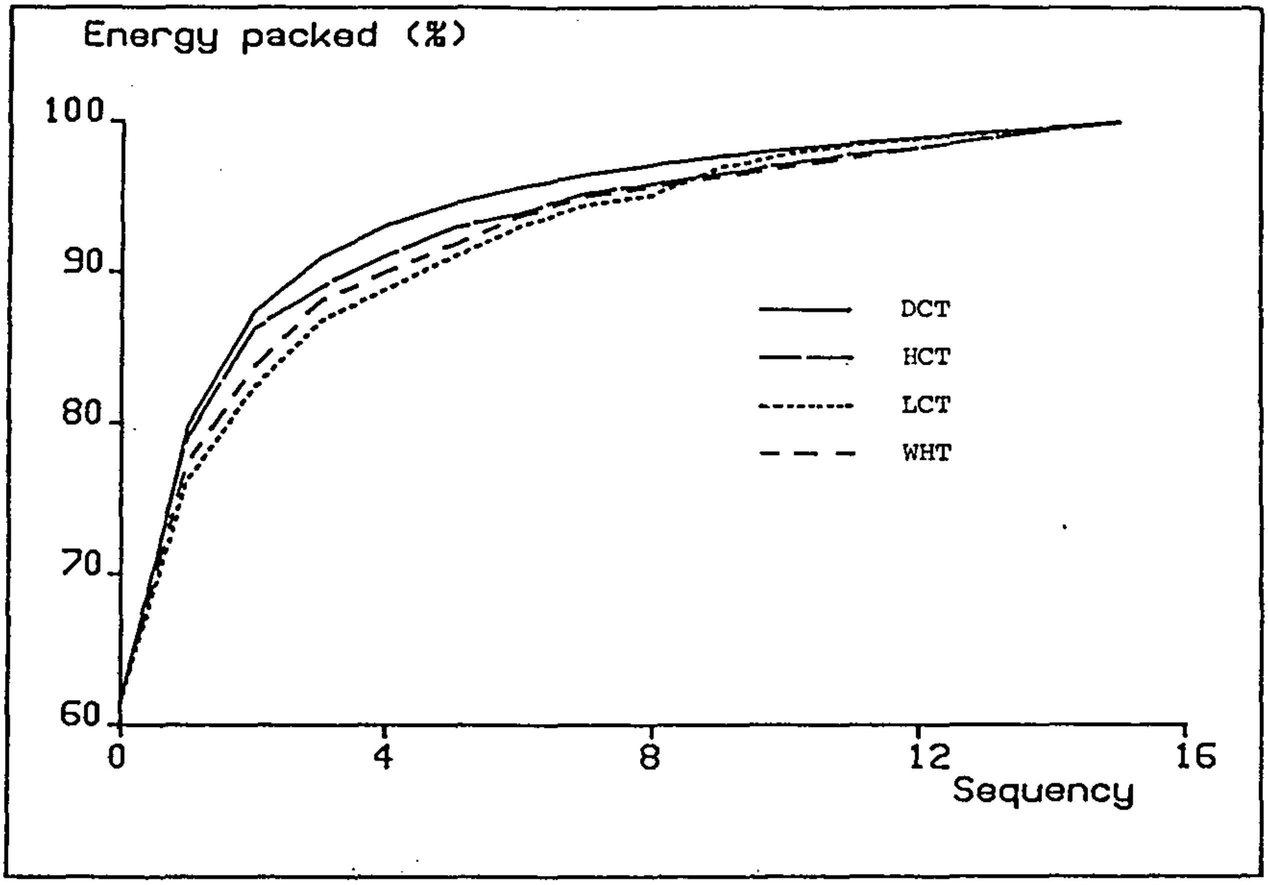


Fig. 4.7 Energy packing ability of the 16-order DCT, HCT, LCT and WHT vs number of retained coefficients.

can always pack more energy than the other transforms and the HCT can always pack more energy than the Walsh transform, b) the LCT can pack more energy than Walsh transform when the block size is 8 x 8; and for r larger than 9 when block size is 16 x 16, c) the HCT is better than the LCT for smaller r and vice versa, d) when the block size is 4 x 4, only the Walsh transform has inferior energy packing ability, all the other three transforms perform equally well.

Further, the HCT and LCT were compared with the DCT, Walsh transform, slant transform and Haar transform using transform efficency (equation 4.5) as the criterion. Markov processes with adjacent element correlations 0.9, 0.5 and 0.2, representing low, medium and high activity pictures respectively, were examined. The results for block sizes 4x4, 8x8, 16x16 , 32x32 and 64x64 are listed in Table 4.7.

When the adjacent element correlation is 0.9, the DCT has the highest transform efficiency, followed by the slant transform and the HCT except when the block size is 4x4 where the HCT has higher transform efficiency than the slant transform. The results are very much the same when the adjacent element correlations are 0.5 and 0.2. However, at block size 8x8, the transform efficiencies of the HCT and LCT, although less than that of the DCT, are higher than that of the slant transform. At block size 4x4, both the HCT and LCT have the highest efficiency.

In general, both the HCT and LCT have a closer performance to that of the DCT for smaller block size. This is probably because only four

	Transform	4	8	16	32	64
0.2	Haar	85.63	66.23	75.96	70.61	67.75
	WHT	91.24	83.61	78.06	71.63	66.84
	HCT	94.55	87.62	81.12	75.31	70.16
	LCT	94.55	89.21	81.94	75.77	70.57
	Slant	92.31	85.38	79.72	74.99	70.88
	DCT	93.40	90.33	88.87	88.16	87.81
0.5	Haar	77.13	59.26	49.52	44.74	42.33
	WHT	84.21	69.75	59.39	51.71	45.79
	HCT	91.42	77.07	65.30	56.61	49.80
	LCT	91.42	77.07	62.82	52.77	45.82
	Slant	87.91	75.71	66.48	59.72	54.44
	DCT	89.61	83.14	79.76	78.12	77.31
0.9	Haar	89.03	70.33	51.06	36.91	29.00
	WHT	92.12	77.13	60.84	48.20	39.62
	HCT	95.24	84.09	68.39	54.36	44.07
	LCT	95.24	79.18	56.50	40.36	30.85
	Slant	94.95	85.84	74.09	62.76	54.04
	DCT	95.75	89.83	82.75	76.41	72.34

Table 4.7 : The transform efficiency of the Haar transform, Walsh transform(WHT), HCT, LCT, slant transform and DCT for block sizes 4x4, 8x8, 16x16, 32x32 and 64x64 and adjacent element coefficients (ρ) 0.2, 0.5 and 0.9.

levels are allowed in the n-order HCT and LCT for all n whilst other transforms like the n-order DCT has n levels. For large n, four levels are insufficient to constitute smooth changing low sequency basis vectors, hence resulting in smaller variances for the low sequency coefficients, and smaller transform efficiency compared with the DCT. Also, as expected, the HCT has higher transform efficiency than the LCT when adjacent element correlation is high, and vice versa.

4.5.2 Test using real pictures

4.5.2.1 Experimental procedure

Figs. 4.8a to 4.8c show the three 256 x 256 eight-bit/pel pictures used to test the transformations.

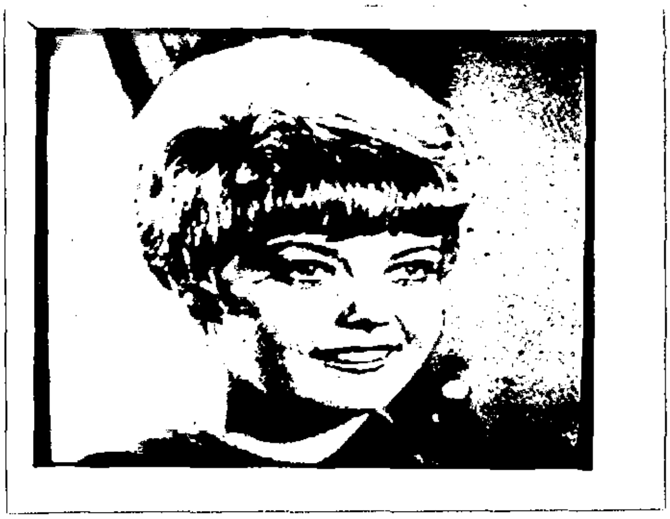


Fig. 4.8a

The 'girl'

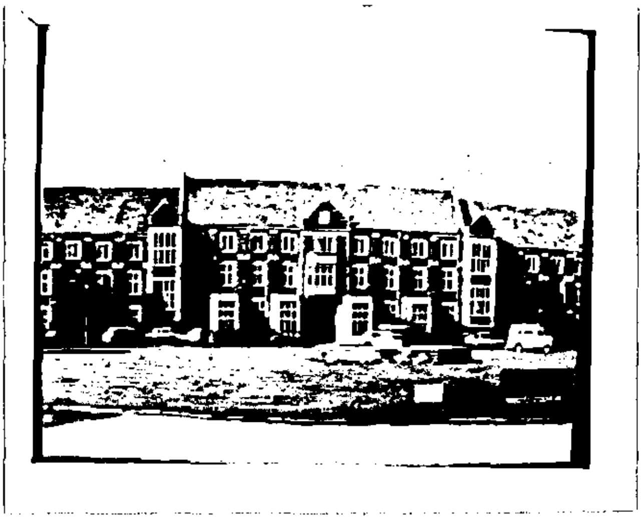


Fig. 4.8b

The 'house'

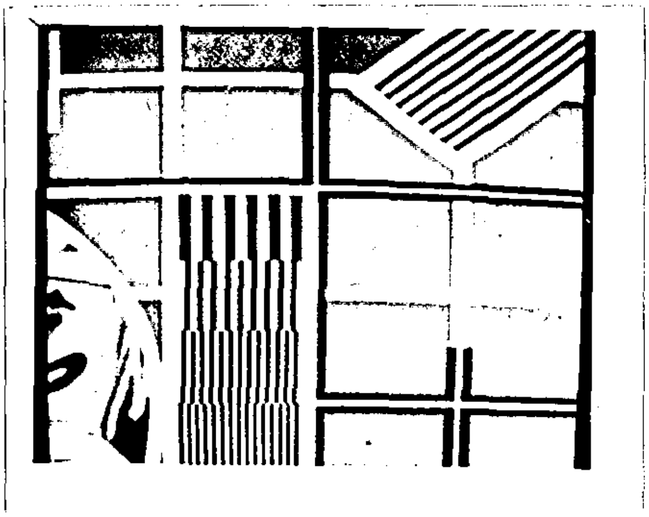


Fig. 4.8c

The 'BBC testcard'

Fig. 4.8 The 3 original pictures.

(i) The head and shoulder picture of a girl (Fig. 4.8a):
This is a typical low activity picture and has been used by many researchers for several years now. It was supplied by British Telecom.

(ii) The picture of a house (Fig. 4.8b):
The top and bottom parts of the picture have very low activity, whilst the central region contains quite a lot of detail. It represent a picture of medium activity. This picture was taken from a print through the Video Acquisition and Display system developed by W.C.Wong [135].

(iii) A part of the BBC testcard (Fig. 4.8c):
The picture contains artificial patterns and has a high degree of activity. It was supplied by British Telecom.

The pictures were divided into square blocks [X] of size equal to that of the transform [T], and then transformed into blocks of transform coefficients [C].

$$[C] = [T]\,[X]\,[T]^t \qquad \text{------------------------------}(4.7)$$

In each block, 75% of the coefficients whose variances are the smallest are truncated, and then the remaining coefficients [D] are inverse transformed to form [Y].

$$[Y] = [T]^t\,[D]\,[T] \qquad \text{--------------------------------}(4.8)$$

The ability of the transform to pack energy into a few transform coefficients is indicated by the normalized mean square error (NMSE).

$$\text{NMSE} = \frac{\sum_{p=0}^{n-1}\sum_{q=0}^{n-1}\sum_{i=0}^{n-1}\sum_{j=0}^{n-1}(y_{pqij} - x_{pqij})^2}{\sum_{p=0}^{n-1}\sum_{q=0}^{n-1}\sum_{i=0}^{n-1}\sum_{j=0}^{n-1} x_{pqij}^2} \times 100\% \qquad \text{--(4.9)}$$

where x_{pqij} and y_{pqij} are the (p,q)th element of the (i,j)th blocks of [X] and [Y] respectively. Small NMSE indicates high ability of the transform to pack energy into a few transform coefficients. The above procedure was carried out using the six transformations at block sizes 4x4, 8x8, 16x16, 32x32 for the three pictures.

4.5.2.2 Results and discussion

The results are summarized in Table 4.8. For the girl and house pictures for block sizes 8x8 and above, the DCT has the lowest NMSE followed by the slant transform; when the block size is 4x4, the HCT and LCT have the lowest NMSE. For the BBC testcard, the DCT always has the lowest NMSE, then comes the LCT at block sizes 4x4 and 8x8, and the slant transform at block sizes 16x16 and above. In conclusion, the following points can be summarized from the Table 4.8.

(1) The NMSE of the HCT is always lower than that of the Walsh transform and lies about midway between that of the DCT and Walsh transforms. The LCT does not always perform better than the Walsh transform. It has a higher NMSE than that of the Walsh transform at block size 32x32 for the two natural images.

(2) At block size 4x4, both the LCT and HCT beat all other contenders on the two natural pictures. However, they cannot maintain their

performance as the block size increases.

(3) The HCT is a better transformation for natural pictures whilst the LCT has excellent performance using the artificial picture when the block size is smaller than 16x16.

	4	8	16	32
Haar	2.20	2.10	2.06	2.02
WHT	2.18	2.09	2.04	1.98
HCT	1.82	1.83	1.89	1.89
LCT	1.82	1.83	1.97	2.06
Slant	1.91	1.78	1.77	1.74
DCT	1.85	1.59	1.49	1.40

a) The 'Girl' picture

	4	8	16	32
Haar	9.32	7.67	7.46	7.19
WHT	9.04	7.65	7.36	7.26
HCT	8.43	7.31	7.02	6.99
LCT	8.43	7.35	7.42	7.51
Slant	8.50	7.13	6.92	6.70
DCT	8.44	6.96	6.39	6.07

b) The 'House' picture

	4	8	16	32
Haar	6.74	5.21	4.97	4.41
WHT	5.76	4.83	4.35	3.81
HCT	4.64	3.72	3.55	3.35
LCT	4.64	2.48	3.50	3.38
Slant	4.74	3.80	3.25	2.76
DCT	4.64	2.88	2.11	1.76

c) The 'BBC testcard'

Table 4.8 : NMSE of the processed pictures as a function of block size and type of transform used.

4.5.3 Conclusions

The tests on the one-dimensional Markov process and on real pictures both show that the HCT is better than the Walsh transform for all block sizes. For large block sizes, the LCT is not as good as the Walsh transform, in terms of both NMSE and efficiency. The LCT seems more suitable for an artificial picture such as the BBC testcard. Finally, both the HCT and LCT have better performance at small block sizes than at large block sizes. For small block sizes such as 4x4, 8x8 and 16x16, which are most suitable for image transform coding, the two tranforms have good performance.

4.6 IMPLEMENTATION OF THE HCT AND LCT BY FAST COMPUTATIONAL ALGORITHMS

4.6.1 Introduction

In this section the implementation of the two new transforms, the HCT and LCT is presented. Let X and C be the vectors containing the input data and the transform coefficients respectively.

$$C = [\,T\,] \times X$$

where

$$[T] = \begin{bmatrix} k_1 \times T_1^t \\ k_2 \times T_2^t \\ \cdots \\ k_n \times T_n^t \end{bmatrix} \quad \text{-----------------------------} (4.11)$$

$$k_i = 1/(| T_i |) \quad \text{------------------------------------}(4.12)$$

$[T_1 , T_2 , \ldots , T_n]^t$ represent the unnormalized HCT and LCT. Tables 4.9 and 4.10 list these transforms of order four, eight and sixteen. Fig. 4.9 shows a transform coding system in which the transformation of X into C requires N^2 real number multiplications. However, with a proper arrangement, the transformation can be achieved with $N \log_2 N$ subtractions or additions together with a number of right shifts. The configuration of this system is shown in Fig.4.10. Thus,

$$C = \begin{bmatrix} k_1 \times T_1^t \\ k_2 \times T_2^t \\ \cdot\cdot \quad \cdot\cdot \\ k_n \times T_n^t \end{bmatrix} \times X$$

$$= \begin{bmatrix} k_1 & & & \\ & k_2 & & \\ & & \ddots & \\ & & & k_n \end{bmatrix} \times \begin{bmatrix} T_1^t \\ T_2^t \\ \cdot\cdot \\ T_n^t \end{bmatrix} \times X \quad (4.13)$$

Let

$$W = \begin{bmatrix} T_1^t \\ T_2^t \\ \cdot\cdot \\ T_n^t \end{bmatrix} \times X \quad \text{--------------}(4.14)$$

1	1	1	1
1	a	-a	-1
1	-1	-1	1
a	-1	1	-a

a) The 4-order UHCT

1	1	1	1	1	1	1	1
1	1	a	a	-a	-a	-1	-1
1	a	-a	-1	-1	-a	a	1
a	a	-1	-1	1	1	-a	-a
1	-1	-1	1	1	-1	-1	1
1	-1	-a	a	-a	a	1	-1
a	-1	1	-a	-a	1	-1	a
a	-a	1	-1	1	-1	a	-a

b) The 8-order UHCT

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	a	a	a	a	-a	-a	-a	-a	-1	-1	-1	-1
1	1	a	a	-a	-a	-1	-1	-1	-1	-a	-a	a	a	1	1
a	a	a	a	-1	-1	-1	-1	1	1	1	1	-a	-a	-a	-a
1	a	-a	-1	-1	-a	a	1	1	a	-a	-1	-1	-a	a	1
1	1	-1	-1	-a	-a	a	a	-a	-a	a	a	1	1	-1	-1
a	a	-1	-1	1	1	-a	-a	-a	-a	1	1	-1	-1	a	a
a	a	-a	-a	1	1	-1	-1	1	1	-1	-1	a	a	-a	-a
1	-1	-1	1	1	-1	-1	1	1	-1	-1	1	1	-1	-1	1
1	-1	-1	1	a	-a	-a	a	-a	a	a	-a	-1	1	1	-1
1	-1	-a	a	-a	a	1	-1	-1	1	a	-a	a	-a	-1	1
a	-a	-a	a	-1	1	1	-1	1	-1	-1	1	-a	a	a	-a
a	-1	1	-a	-a	1	-1	a	a	-1	1	-a	-a	1	-1	a
1	-1	1	-1	-a	a	-a	a	-a	a	-a	a	1	-1	1	-1
a	-a	1	-1	1	-1	a	-a	a	-a	-1	1	-1	1	-a	a
a	-a	a	-a	1	-1	1	-1	1	-1	1	-1	a	-a	a	-a

c) The 16-order UHCT

Table 4.9

This table lists the unnormalized HCT kernels of order 4, 8 and 16.

note: 'a' is used to represent 1/2

1	1	1	1
1	a	-a	-1
1	-1	-1	1
a	-1	1	-a

a) The 4-order ULCT

1	1	1	1	1	1	1	1
a	1	1	a	-a	-1	-1	-a
1	a	-a	-1	-1	-a	a	1
1	a	-a	-1	1	a	-a	-1
1	-1	-1	1	1	-1	-1	1
1	-a	-a	1	-1	a	a	-1
a	-1	1	-a	-a	1	-1	a
a	-1	1	-a	a	-1	1	-a

b) The 8-order ULCT

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
a	1	1	a	a	1	1	a	-a	-1	-1	-a	-a	-1	-1	-a
a	1	1	a	-a	-1	-1	-a	-a	-1	-1	-a	a	1	1	a
a	1	1	a	-a	-1	-1	-a	a	1	1	a	-a	-1	-1	-a
1	a	-a	-1	-1	-a	a	1	1	a	-a	-1	-1	-a	a	1
1	a	-a	-1	-1	-a	a	1	-1	-a	a	1	1	a	-a	-1
1	a	-a	-1	1	a	-a	-1	-1	-a	a	1	-1	-a	a	1
1	a	-a	-1	1	a	-a	-1	1	a	-a	-1	1	a	-a	-1
1	-1	-1	1	1	-1	-1	1	1	-1	-1	1	1	-1	-1	1
1	-a	-a	1	1	-a	-a	1	-1	a	a	-1	-1	a	a	-1
1	-a	-a	1	-1	a	a	-1	-1	a	a	-1	1	-a	-a	1
1	-a	-a	1	-1	a	a	-1	1	-a	-a	1	-1	a	a	-1
a	-1	1	-a	-a	1	-1	a	a	-1	1	-a	-a	1	-1	a
a	-1	1	-a	-a	1	-1	a	-a	1	-1	a	a	-1	1	-a
a	-1	1	-a	a	-1	1	-a	-a	1	-1	a	-a	1	-1	a
a	-1	1	-a	a	-1	1	-a	a	-1	1	-a	a	-1	1	-a

c) The 16-order ULCT

Table 4.10

This table lists the unnormalized LCT kernels of order 4, 8 and 16.

note: 'a' is used to represent 1/2.

Then

$$C = \begin{bmatrix} k_1 & & & \\ & k_2 & & \\ & & \ddots & \\ & & & k_n \end{bmatrix} x\ W \qquad \text{--------------}(\ 4.15\ )$$

Therefore, the real number transformation of X into C is divided into two stages. The first step is to transform X into W (equation 4.14) which involves only additions, subtractions and right shift operations. The second step is the real number multiplication process to convert W into C (equation 4.15) which can be incorporated into the quantization operation. At the receiver, the vector of quantized coefficients (CQ) can be re-transformed into the signal domain to form vector XQ by equation 4.16.

$$\begin{aligned} XQ &= [T]^t\ x\ CQ \qquad \text{----------------------------------}(\ 4.16\ ) \\ &= [\ k_1 x T_1 \quad k_2 x T_2 \quad \ldots. \quad k_n x T_n\ ]\ x\ CQ \\ &= [T_1\ T_2\ T_3\ \ldots\ T_n\ ]\ x \begin{bmatrix} k_1 & & & \\ & k_2 & & \\ & & \ddots & \\ & & & k_n \end{bmatrix} x\ CQ \end{aligned}$$

Let

$$WQ = \begin{bmatrix} k_1 & & & \\ & k_2 & & \\ & & \ddots & \\ & & & k_n \end{bmatrix} x\ CQ \qquad \text{----------------------}(\ 4.17\ )$$

Hence,

$$XQ = [\, T_1 \;\; T_2 \;\; \ldots \;\; T_n \,] \times WQ \qquad \text{------------------------(4.18)}$$

Therefore, the real number inverse transformation of CQ into XQ is also divided into two stages. The first step is the conversion of the bit stream L (representing CQ) into WQ and then the unnormalized inverse transfomation of WQ into XQ.

Fast computational algorithms for the forward and inverse UHCT and ULCT algorithms requiring only $N \log_2 N$ operations will be given in the next section.

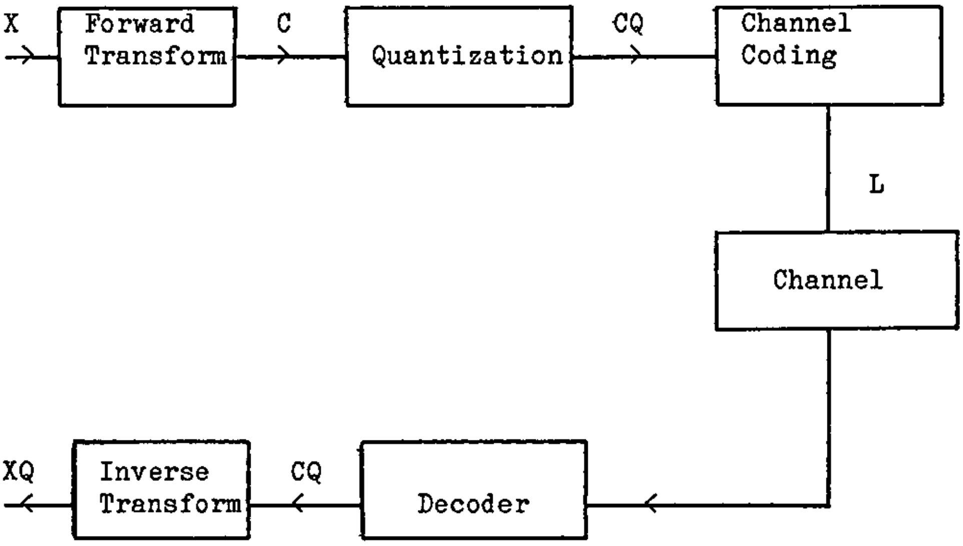


Fig.4.9 A transform coding system in which the transforms in both directions require real number multiplications.

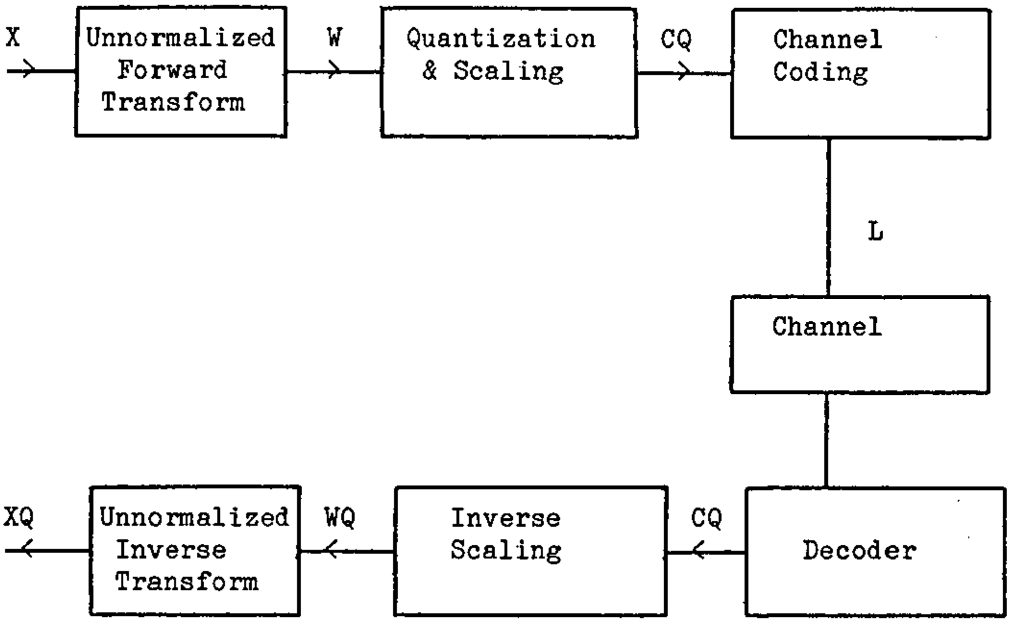


Fig.4.10 A transform coding system in which the transforms in both directions require only additions, subtractions and right shifts.

4.6.2 Fast computational algorithms for the HCT and LCT generated from the dyadic symmetry decompositions

The derivation of the fast computational algorithms for the Walsh transform by repeated application of dyadic symmetry decomposition has been given in section 3.6. In this section the same technique is used to obtain a fast computational algorithm for the forward and inverse UHCT and ULCT. A Walsh transform has all the dyadic symmetries, but the ULCT and UHCT have only some of them. However, in some cases, it is still possible to apply the k th dyadic decomposition to a matrix, some of whose basis vectors do not have the k th dyadic symmetry. Generally, the k th dyadic symmetry decomposition is possible for a transform [T] if

$$\frac{t_{i\,j}}{t_{i\,j(+)k}} = c_{j0}$$

$$\frac{t_{r\,s}}{t_{r\,s(+)k}} = -c_{s1} \qquad \text{------------------------------------}(4.19)$$

where

(1) c_{j0} and $-c_{s1}$ are positive constants

(2) i is such that T_i has even k th dyadic symmetry and

r is such that T_r has odd k th dyadic symmetry.

(3) j and s are dummy variables. j, s ,j(+)k and s(+)k are in the range $[0,2^m-1]$.

4.6.2.1 A fast computational algorithm for the forward unnormalized LCT (ULCT)

From section 4.4, it can be seen that a 2^m by 2^m LCT lacks the first dyadic symmetry. In fact, the LCT has the 2^{m-1} dyadic symmetries. For example, when m is three, the LCT has three dyadic symmetries which are

S	[s_1	s_2	s_3]
3	0	1	1
4	1	0	0
7	1	1	1

In general, the LCT has dyadic symmetry S equal to [s_1 s_2 .... s_m]

where s_{m-1} is equal to s_m. Therefore, except for second order all 2^m-order LCT have the 2^m-1 th dyadic symmetry. Hence, a fast computational algorithm can be obtained by repeated applications of the 2^m-1 th dyadic symmetry decomposition. Details of the process are now explained by using the 16x16 ULCT as an example.

By means of the fifteenth dyadic symmetry decomposition, a 16x16 LCT can be broken down into an 8x8 even-transform and an 8x8 odd-transform. The even-transform is equal an 8x8 ULCT and the odd-transform is

$$[Y3] = \begin{bmatrix} a & 1 & 1 & a & a & 1 & 1 & a \\ a & 1 & 1 & a & -a & -1 & -1 & -a \\ 1 & a & -a & -1 & -1 & -a & a & 1 \\ 1 & a & -a & -1 & 1 & a & -a & -1 \\ 1 & -a & -a & 1 & 1 & -a & -a & 1 \\ 1 & -a & -a & 1 & -1 & a & a & -1 \\ a & -1 & 1 & -a & -a & 1 & -1 & a \\ a & -1 & 1 & -a & a & -1 & 1 & -a \end{bmatrix} \qquad \text{-----------(4.20)}$$

where $a = 0.5$

Again, using the seventh dyadic symmetry decomposition, the 8x8 ULCT is broken down into an even-transform which equals an 4x4 ULCT, and an odd-transform which equals [Y2].

$$[Y2] = \begin{bmatrix} a & 1 & 1 & a \\ 1 & a & -a & -1 \\ 1 & -a & -a & 1 \\ a & -1 & 1 & -a \end{bmatrix} \qquad \text{---------------------(4.21)}$$

where $a = 0.5$

For 8x8 [Y3], both the 4x4 even-transform and odd-transform are [Y2]. The 4x4 ULCT and [Y2] transform are then decomposed into an even-transform and an odd-transform by the third dyadic symmetry decomposition. For the 4x4 ULCT transform, the 2x2 even-transform is the 2x2 Walsh transform, and the 2x2 odd-transform is

$$[Y1] = \begin{bmatrix} 1 & a \\ a & -1 \end{bmatrix} \qquad \text{------------------------------------} (\ 4.22\ )$$

For the 4x4 [Y2] transform, the two by two even-transform is [X1] and the 2x2 odd-transform is [Y1].

$$[X1] = \begin{bmatrix} a & 1 \\ 1 & -a \end{bmatrix} \qquad \text{------------------------------------} (\ 4.23\ )$$

The whole process is summarized in Fig. 4.11. In the binary tree, each node, representing a transform, has two offsprings. The one on the left is its even transform and the one on the right is its odd transform.

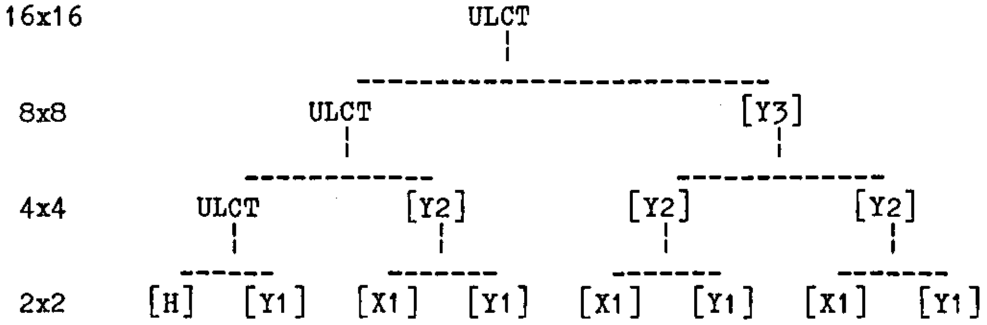


Fig.4.11 The binary tree depicting the dyadic symmetry decompositions of a 16-order ULCT.

One of the ways to implement this fast computational algorithm for a 16x16 ULCT is given in the signal flow diagram shown in Fig. 4.12. The signal flow diagram of the fast computational algorithm for the ULCT is the same as that for the Walsh transform until the last iter-

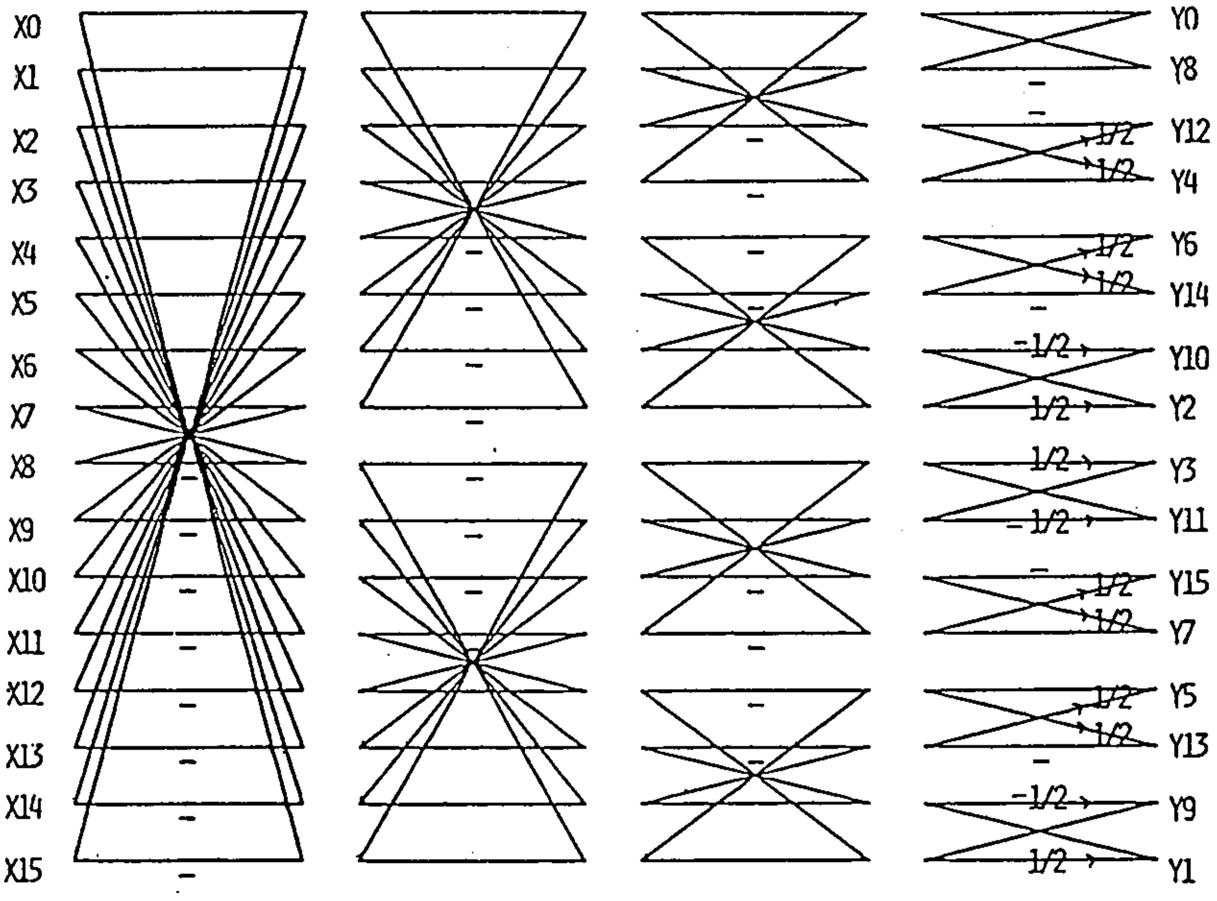


Fig. 4.12 A signal flow diagram for the fast forward 16x16 ULCT.

ation where the sublock size is two. In the last iteration, 14 right shifts are required before certain additions and subtractions.

4.6.2.2 A fast computational algorithm for the forward unnormalized HCT (UHCT)

As given in section 4.4.1, a 2^m by 2^m HCT has only the 2^m-1 th dyadic symmetry. However, a fast computational algorithm can still be obtained by the repetitive use of the dyadic symmetry decompositions. Details of the process are now given by using a 16x16 UHCT as example.

A 16x16 HCT has the fifteenth dyadic symmetry, thus, it can be decomposed into an even-transform and odd-transform. The even-transform is the 8x8 UHCT and the odd-transform is

$$[X3] = \begin{bmatrix} 1 & 1 & 1 & 1 & a & a & a & a \\ a & a & a & a & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & -a & -a & a & a \\ a & a & -a & -a & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & a & -a & -a & a \\ a & -a & -a & a & -1 & 1 & 1 & -1 \\ 1 & -1 & 1 & -1 & -a & a & -a & a \\ a & -a & a & -a & 1 & -1 & 1 & -1 \end{bmatrix} \quad \text{----(4.24)}$$

Similarly, the 8x8 UHCT transform can be decomposed into a 4x4 UHCT (even-transform) and [X2] transform(odd-transform) by the seventh dyadic symmetry decomposition.

$$[X2] = \begin{bmatrix} 1 & 1 & a & a \\ a & a & -1 & -1 \\ 1 & -1 & -a & a \\ a & -a & 1 & -1 \end{bmatrix} \quad \text{-----------------(4.25)}$$

Let U and V be

$$U = \begin{bmatrix} x_0 + a \cdot x_7 \\ x_1 + a \cdot x_6 \\ x_2 + a \cdot x_5 \\ x_3 + a \cdot x_4 \end{bmatrix}$$

$$V = \begin{bmatrix} a \cdot x_0 - x_7 \\ a \cdot x_1 - x_6 \\ a \cdot x_2 - x_5 \\ a \cdot x_3 - x_4 \end{bmatrix} \qquad \text{-------------------(4.26)}$$

we have both the even-transform and odd-transform of the [X3] transform equal to the 4x4 Walsh transform. The third dyadic symmetry decomposition can be used to break down both the Walsh and the [X3] transform. The only difference is that the [X3] transform requires modified vectors U and V and so requires eight extra right shifts. Similarly, the 4x4 UHCT and Walsh transform can be decomposed using the third dyadic symmetry decomposition. The whole process is summarized in Fig. 4.13. The signal flow diagram of one of the ways to implement the fast computational algorithm is given in Fig. 4.14.

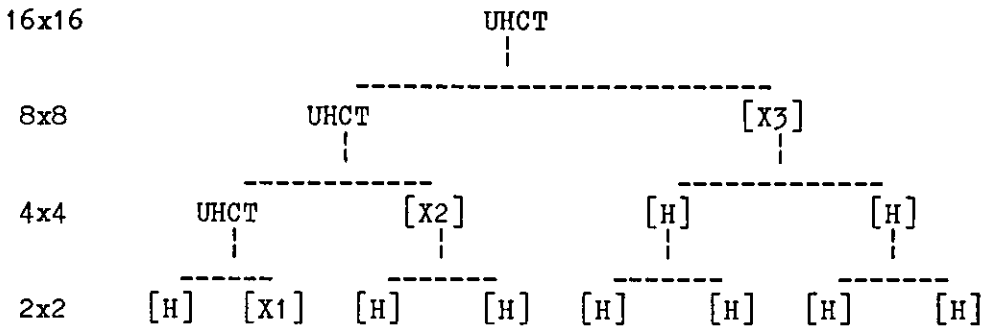


Fig.4.13 The binary tree depicting the dyadic symmetry decompositions of a 16-order UHCT.

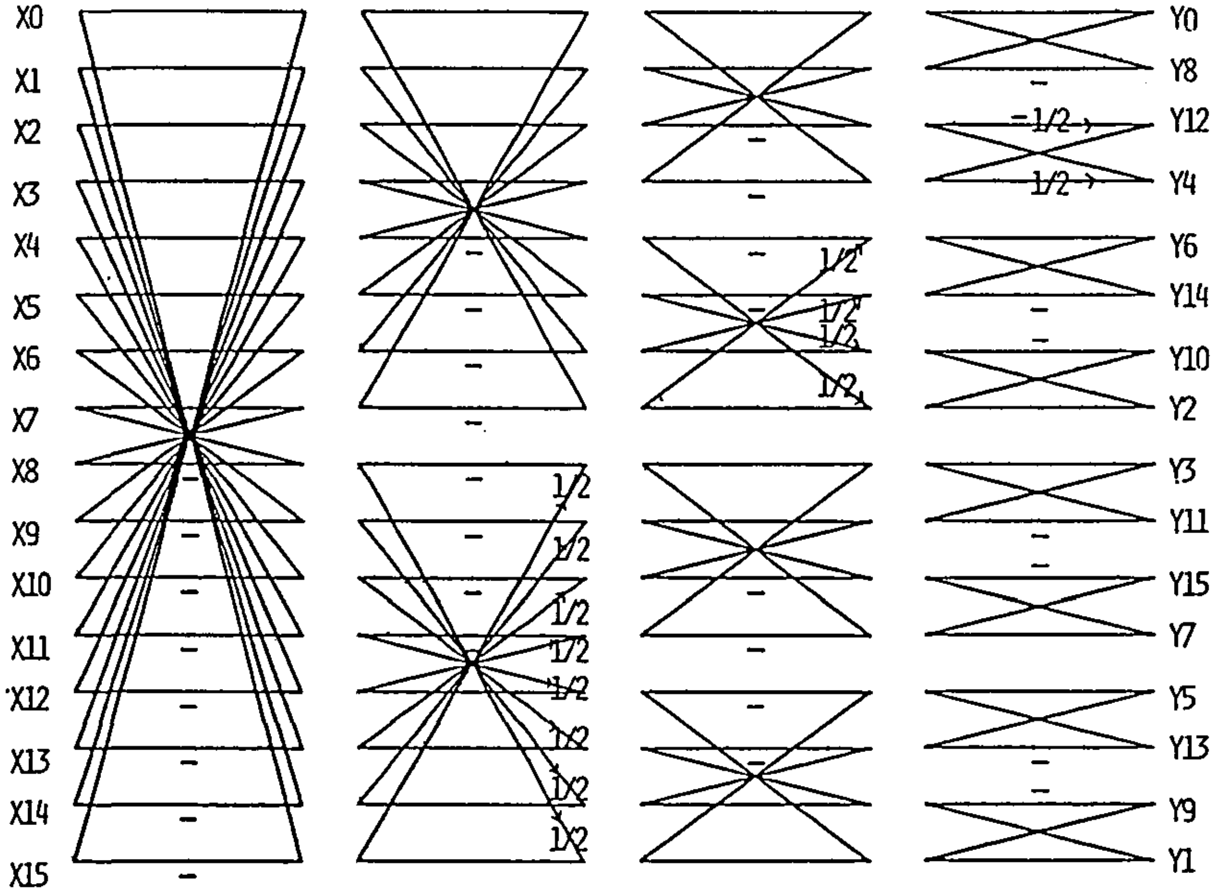


Fig. 4.14 A signal flow diagram for the fast forward 16x16 UHCT.

4.6.2.3 A fast computational algorithm for the transpose of the unnormalized LCT

In section 4.4.1 it has been shown that the inverse LCT can be implemented using the transpose of the ULCT. This section will show that a fast computational algorithm for the transpose of a 16x16 ULCT can be obtained by the repetitive application of the 2^{m-1}th dyadic symmetry decomposition.

The eighth dyadic symmetry decomposition decomposes the transpose of the 16x16 ULCT into

$$U_3 = \begin{bmatrix} x_0 + x_8 \\ a\cdot x_1 + x_9 \\ a\cdot x_2 + x_{10} \\ a\cdot x_3 + x_{11} \\ x_4 + a\cdot x_{12} \\ x_5 + a\cdot x_{13} \\ x_6 + a\cdot x_{14} \\ x_7 + a\cdot x_{15} \end{bmatrix}$$

$$V_3 = \begin{bmatrix} x_0 - x_8 \\ x_1 - a\cdot x_9 \\ x_2 - a\cdot x_{10} \\ x_3 - a\cdot x_{11} \\ a\cdot x_4 - x_{12} \\ a\cdot x_5 - x_{13} \\ a\cdot x_6 - x_{14} \\ a\cdot x_7 - x_{15} \end{bmatrix} \quad \text{-------------------(4.27)}$$

and both the even- and odd-transforms are equal to an 8x8 Walsh transform. The fast computational algorithm obtained by the repetitive use use of the 2^{m-1}th dyadic symmetry decomposition described in section 3.6.3 thus can be used to compute the signal data. The whole process is summarized in Fig. 4.15. The signal flow diagram for the fast computational algorithm is given in Fig. 4.16.

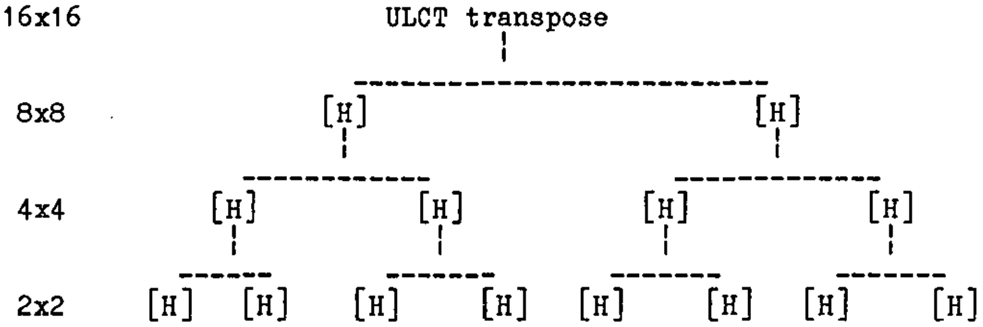


Fig.4.15 The binary tree depicting the dyadic symmetry decompositions of the transpose of a 16-order ULCT.

4.6.2.4 A fast computational algorithm for the transpose of the unnormalized HCT

Section 4.4.1 has shown how an inverse HCT can be implemented by the transpose of UHCT. In this section, a fast computational algorithm using the 2^{m-1}th dyadic symmetry decomposition is given to compute the transpose of the UHCT. By defining

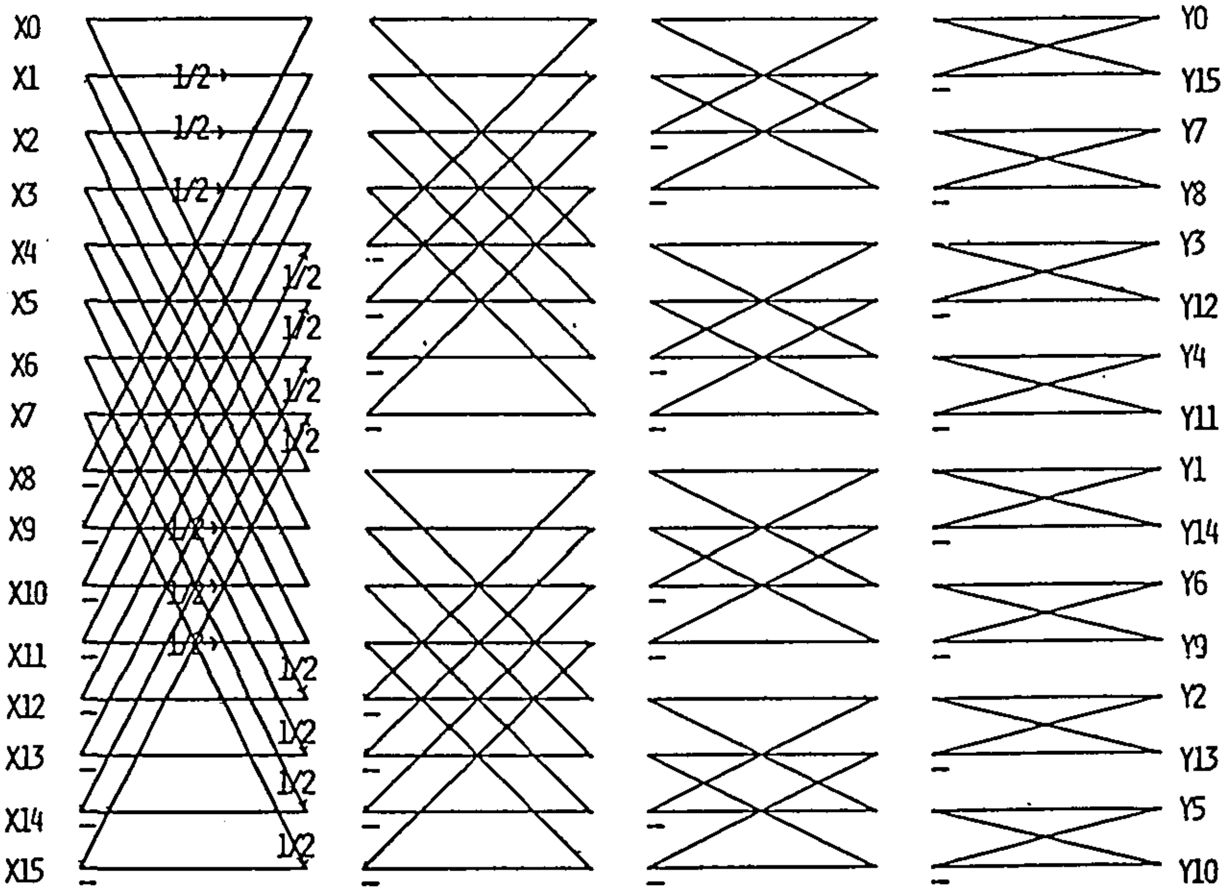


Fig. 4.16 A signal flow diagram for the fast reverse 16x16 ULCT.

$$U_3 = \begin{bmatrix} x_0 + x_8 \\ x_1 + x_9 \\ x_2 + x_{10} \\ x_3 + x_{11} \\ x_4 + a\,x_{12} \\ x_5 + x_{13} \\ x_6 + x_{14} \\ x_7 + x_{15} \end{bmatrix}$$

$$V_3 = \begin{bmatrix} x_0 - x_8 \\ x_1 - x_9 \\ x_2 - x_{10} \\ x_3 - x_{11} \\ a\,x_4 - x_{12} \\ x_5 - x_{13} \\ x_6 - x_{14} \\ x_7 - x_{15} \end{bmatrix} \qquad \text{------------(4.28)}$$

we have both the even- and odd-transforms equal to the transpose of an 8x8 UHCT which can then be decomposed by defining

$$U_2 = \begin{bmatrix} x_0 + x_4 \\ x_1 + x_5 \\ x_2 + a\,x_6 \\ x_3 + x_7 \end{bmatrix}$$

$$V_2 = \begin{bmatrix} x_0 - x_4 \\ x_1 - x_5 \\ a\cdot x_2 - x_6 \\ x_3 - x_7 \end{bmatrix} \qquad \text{------------(4.29)}$$

The 4x4 even- and odd-transforms are the transpose of a 4x4 UHCT which can then be decomposed into two Walsh transforms by defining

$$U_1 = \begin{bmatrix} x_0 + x_2 \\ x_1 + a\cdot x_3 \end{bmatrix}$$

$$V_1 = \begin{bmatrix} x_0 - x_2 \\ a\cdot x_1 - x_3 \end{bmatrix} \qquad \text{------------------------(4.30)}$$

The whole process is summarized in Fig. 4.17. The signal flow diagram of one of the ways to implement this process is shown in Fig. 4.18.

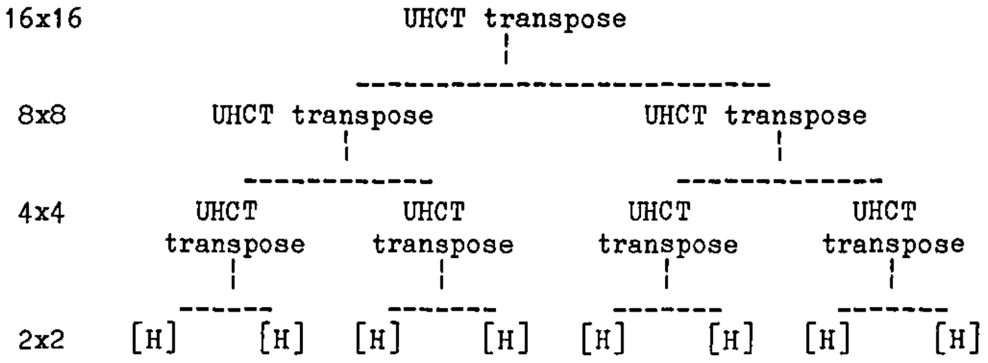


Fig.4.17 The binary tree depicting the dyadic symmetry decompositions of the transpose of a 16-order UHCT.

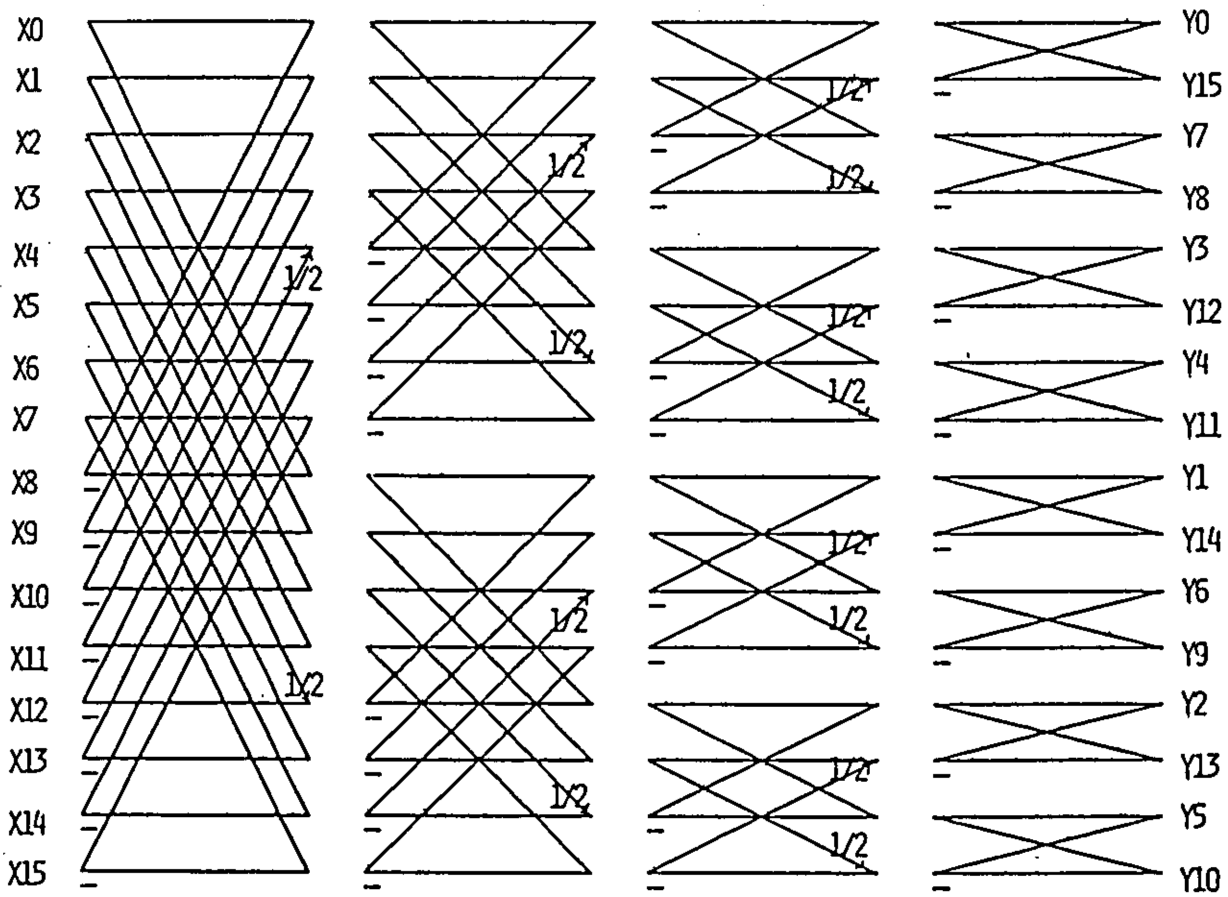


Fig. 4.18 A signal flow diagram for the fast reverse 16x16 UHCT.

4.7 CONCLUSIONS

This chapter has demonstrated the use of the theory of dyadic symmetry to generate two new transforms which can be used as substitutes for the Walsh transform. The new transforms have virtually the same complexity and computational requirements as the Walsh transform. They employ additions, subtractions and binary shifts only but have an efficiency, defined in terms of their ability to decorrelate signal data, which lies between that of the Walsh transform and that of the DCT.

4.8 NOTE ON PUBLICATIONS

The result described in section 4.2 has appeared in a paper entitled 'A technique for generating new image transforms', presented at the 1983 IEE colloquium on 'Transform Techniques in Image Processing' at Savoy Place, London, England. Another paper based on the results described in section 4.2 to 4.5 and entitled 'Simple high efficiency transforms for image coding' was presented at the 1983 Picture Coding Symposium, Davis, California, USA. A paper based on all the results reported in this chapter and entitled 'Generation of orthogonal transforms using the theory of dyadic symmetry' was submitted to the IEEE transactions on Electromagnetic compatibility in 1983. All these papers were jointly authored with R.J.Clarke.

Also, an European patent (patent no.82303825.2) entitled 'Method of transmitting an image and apparatus for carrying out the method', in co-authorship with Dr R.C.Nicol, Mr.B.A.Fenn, Mr.R.J.Clarke and Dr.K.N.Ngan, has been made to claim originality of invention on the HCT and LCT.

CHAPTER FIVE

DC COEFFICIENT RESTORATION SCHEMES

5.1 INTRODUCTION

Picture elements are often highly correlated. In conventional transform coding, the high correlation between pels has been largely, if not entirely, exploited for those pels within the same block. However, the high correlation between pels in different blocks is completely neglected.

Therefore, schemes such as recursive block coding, hybrid coding and the pinned sine transform (section 1.3.3) were devised to utilize this correlation to achieve further reduction in bit rate. On the other hand, Mitchell and Tabatabai[136] used this redundancy to provide channel error correction for Chen and Smith's adaptive transform coding system, thus eliminating the need for channel error protection bits. Their basic approach is to check the four boundaries around each reconstructed image block as shown in Fig.5.1. If sharp grey level changes exist along these boundaries that agree with a dominant error in a single transform coefficient, the coefficient location and magnitude are estimated and a basis picture corresponding to the estimated error is subtracted from the block.

This chapter describes a technique that utilises the same interblock

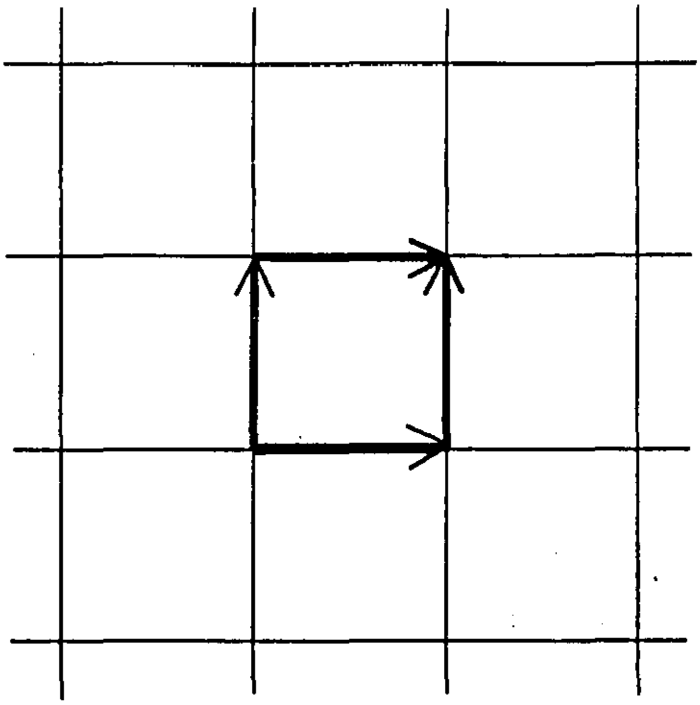


Fig.5.1

The four edges considered in the error correction proposed by Mitchell and Tabatabai.

redundancy to allow, in some cases, the dc coefficients to be estimated at the receiver, thus allowing reductions in bit rate as well as eliminating a major source of difficulty with respect to channel errors. Three schemes, called ELEMENT ESTIMATION, ROW ESTIMATION, and PLANE ESTIMATION, are proposed, and the results of simulations of these methods using different block sizes and different pictures are given.

Section 5.2 evaluates the degree of data compression that can be achieved by not sending the dc coefficients. Also, it is shown that, using the sequency-ordered Walsh transformation, dc coefficient truncation is equivalent to low sequency coefficient truncation. In sections 5.3 to 5.5, the three schemes are described, and results of simulations of these methods are given in section 5.6.

5.2 SIGNIFICANCE OF DC COEFFICIENT RESTORATION SCHEMES

Eight-bit quantization of the dc coefficients is enough to make quantization noise imperceptible. Thus, if the dc coefficients are not sent but estimated at the receiver using a dc coefficient restoration scheme, these 8 bits can be saved. In a two-dimensional transform coding system, this is equivalent to 0.008, 0.031, 0.125 and 0.5 additional bit/pel for block sizes 32x32, 16x16, 8x8 and 4x4 respectively. Consequently, a dc coefficient restoration scheme can save more bits at a smaller, rather than a larger, block size. More insight into this is provided by the following theorem:

Theorem 5.1:

Let a set of 2^m data elements $[\,x_0, \ldots, x_{2^m-1}\,]$ be divided into 2^r sets. Each set thus contains 2^{m-r} data points. The 2^r dc coefficients of the 2^r sets of Walsh transform coefficients can be derived from the first 2^r transform coefficients of the 2^m-order sequency-ordered Walsh transform, and vice versa.

Proof:

The (i,j)th element of the 2^m-order sequency-ordered binary Walsh matrix is as given by equation 3.9

$$b_{ij} = [\,j_1, j_2, \ldots, j_m\,] \begin{bmatrix} & & & & & 1 & 1 \\ & & & & 1 & 1 & \\ & & & 1 & 1 & & \\ & & \cdot & \cdot & & & \\ & \cdot & \cdot & & & & \\ 1 & 1 & & & & & \\ 1 & & & & & & \end{bmatrix} \begin{bmatrix} i_1 \\ \cdot\cdot \\ \cdot\cdot \\ \cdot\cdot \\ i_m \end{bmatrix}$$

If i is within the range $[0, 2^r-1]$, then we have

$$b_{ij} = [\, j_1 , j_2 , \ldots , j_m \,] \left[\begin{array}{cccc|cccc} &&&&&&1&1\\ &&&&&1&1&\\ &&&&1&1&&\\ &&&1&1&&&\\ &&\cdot&1&&&&\\ &\cdot&\cdot&&&&&\\ 1&&&&&&&\end{array}\right] \left[\begin{array}{c} 0\\ \cdot\cdot\\ 0\\ i_{m-r+1}\\ \cdot\cdot\\ i_m \end{array}\right]$$

$\leftarrow m-r \rightarrow | \longleftarrow r \longrightarrow$

$$= [\, j_1 , j_2 , \ldots , j_m \,] \left[\begin{array}{cccc|cccc} &&&&&&1&1\\ &&&&&1&1&\\ &&&&1&1&&\\ &&&&1&&&\\ \hline &&&&&&&\\ &&&&&&&\end{array}\right] \left[\begin{array}{c} 0\\ \cdot\cdot\\ 0\\ i_{m-r+1}\\ \cdot\cdot\\ i_m \end{array}\right]$$

$\leftarrow m-r \rightarrow | \longleftarrow r \longrightarrow$

$$= [\, j_1 , \ldots , j_m \,] \left[\begin{array}{cccc} &&1&1\\ &1&1&\\ 1&1&&\\ 1&&&\\ \hline &&&\\ &&&\end{array}\right] \left[\begin{array}{c} i_{m-r-1}\\ \cdot\cdot\\ \cdot\cdot\\ \cdot\cdot\\ i_m \end{array}\right]$$

$\longleftarrow r \longrightarrow$

$$= [\, j_1 , \ldots , j_r \,] \left[\begin{array}{cccccc} &&&&1&1\\ &&&1&1&\\ &&1&1&&\\ &1&1&&&\\ &\cdot&\cdot&&&\\ 1&1&&&&\\ 1&&&&&\end{array}\right] \left[\begin{array}{c} i_{m-r+1}\\ \cdot\cdot\\ \cdot\cdot\\ \cdot\cdot\\ i_m \end{array}\right] \text{----} (5.1)$$

$\longleftarrow r \longrightarrow$

Therefore, if we want to calculate transform coefficient c_i where i is within the range $[\ 0,2^r -1\ ]$, we can use equation 5.1 instead of equation 3.9. In the other words, the first 2^r transform coefficients of the 2^m-order Walsh transform can be obtained by first dividing 2^m

data elements into 2^r groups, adding the data elements of the groups together with proper scaling to form dc coefficients,and finally applying the $2^r \times 2^r$ Walsh transform.

For example, consider a set of 16 elements $[x_0, x_1, \ldots, x_{15}]$ which are divided into 4 blocks, $[x_0, \ldots, x_3]$, $[x_4, \ldots, x_7]$, $[x_8, \ldots, x_{11}]$ and $[x_{12}, \ldots, x_{15}]$. Let the dc coefficients of the 4 blocks be d_0, d_1, d_2 and d_3. Also, let the Walsh transform coefficients of $[x_0, x_1, \ldots, x_{15}]$ be $[c_0, c_1, \ldots, c_{15}]$. Then we have

$$\begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{bmatrix} = [H] \times \begin{bmatrix} d_0 \\ d_1 \\ d_2 \\ d_3 \end{bmatrix}$$

where $[H]$ is the 4-order Walsh transform.

This theorem implies that the first 2^r of the 2^m sequency-ordered Walsh transform coefficients can be estimated using dc coefficient restoration schemes of block size 2^{m-r}, and therefore, that a dc coefficient restoration scheme can be viewed as a low order coefficient restoration scheme.

5.3 ELEMENT ESTIMATION

5.3.1 Description

The dc coefficient of block (1,1), a(1,1), is set to an arbitrary level and the grey levels in block (1,1) are adjusted accordingly. The dc coefficient of block (1,2) is then chosen as that which minimizes the square magnitude of the edge difference vector between these two blocks. Similarly,dc coefficient a(1,j) is estimated from a(1,j-1), and a(i,1) estimated from a(i-1,1) until all the dc coefficients in the first row and first column have been estimated. The next step is the estimation of a(i,j) from a(i-1,j) and a(i,j-1). This is done by minimizing the square magnitudes of the two edge difference vectors between these three blocks. When all the coefficients have been estimated, the overall grey level of the picture is brought within the desired range for display.

5.3.2 Theoretical development

The method of estimation of a(i,j) from the edge difference vectors and the previously estimated dc coefficients a(i-1,j) and a(i,j-1) is now developed. Consider a picture of N by N blocks, each block having n by n pels. Let [x(p,q)] and [c(r,s)] be blocks of original pels and transform coefficients respectively.

$$[\ c(r,s)\ ] = [\ T\ ]\ [\ x(p,q)\ ]\ [\ T\ ]^{t} \qquad \text{-----------------}(\ 5.2\ )$$

$$[\ T\ ]^{t} = [\ T_{1},T_{2},T_{3},\ldots\ldots,T_{n}\ ]^{t}$$

where [T] is an orthogonal transformation with

$$T_{1}^{t} = [\ 1,1,\ldots,1\ ]\ /\ \sqrt{n} \qquad \text{--------------------}(\ 5.3\ )$$

Both the DCT and Walsh transform satisfy equation 5.3. The block of pels having zero dc level is given by the inverse transformation

$$[\ u(p,q)\ ] = [\ T\ ]^{t}\ [\ f(r,s)\ ]\ [\ T\ ]$$

$$\text{where}\quad f(r,s) \begin{array}{ll} = 0 & r=s=1 \\ = c(r,s) & \text{otherwise} \end{array}$$

The basis picture of the dc coefficient is given by

$$T_{1} \times T_{1}^{t} = (1/n) \times \begin{bmatrix} 1, & \ldots\ldots, & 1 \\ 1, & \ldots\ldots, & 1 \\ 1, & \ldots\ldots, & 1 \\ 1, & \ldots\ldots, & 1 \end{bmatrix}$$

Therefore, the vectors at the four edges of the basis picture are all equal to

$$V = [\ 1/n,\ \ldots\ldots\ldots,\ 1/n\ ]^{t} \qquad \text{--------------}(\ 5.4\ )$$

Let $u(p,q)_{i,j}$ be the (p,q)th pel in the (i,j)th block. The two edge difference vectors, as shown in Fig.5.2, considered in the estimation of a(i,j) are a) the vertical edge difference vector:

$$D_{1,i,j} = a(i,j-1) \times V + \begin{bmatrix} u(1,n)_{i,j-1} & - u(1,1)_{i,j} \\ u(2,n)_{i,j-1} & - u(2,1)_{i,j} \\ \ldots\ldots\ldots\ . & \ldots\ldots\ldots\ldots \\ \ldots\ldots\ldots\ . & \ldots\ldots\ldots\ldots \\ u(n,n)_{i,j-1} & - u(n,1)_{i,j} \end{bmatrix} \quad (\ 5.5\ )$$

where i and j lie in the range [2,N], and b) the horizontal edge difference vector:

$$D_{2,i,j} = a(i-1,j) \times V + \begin{bmatrix} u(n,1)_{i-1,j} & - u(1,1)_{i,j} \\ u(n,2)_{i-1,j} & - u(1,2)_{i,j} \\ \cdots\cdots\cdots & \cdots\cdots\cdots \\ \cdots\cdots\cdots & \cdots\cdots\cdots \\ u(n,n)_{i-1,j} & - u(1,n)_{i,j} \end{bmatrix} \quad (5.6)$$

where i and j lie in the range [2,N].

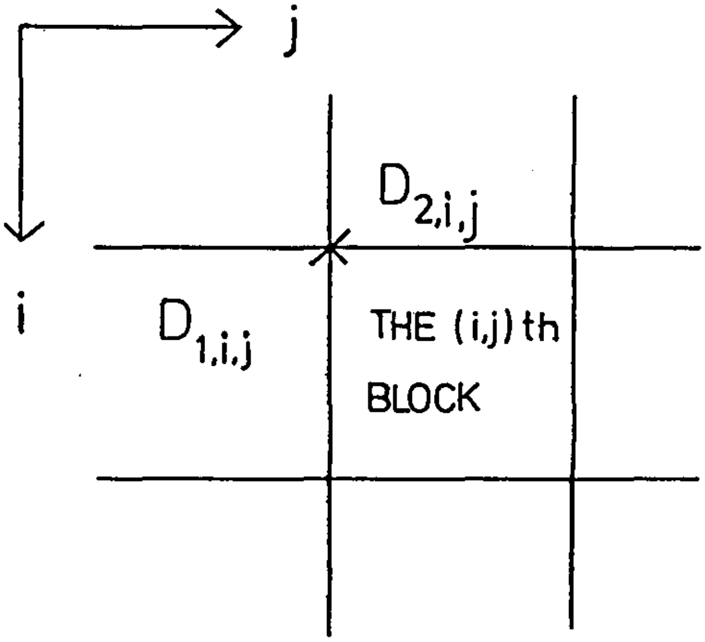


Fig.5.2

The two edge difference vectors considered in ELEMENT ESTIMATION.

If a dc coefficient, a(i,j), is added to block (i,j), then the two edge difference vectors are changed to a) new vertical edge difference vector:

$$W_{1,i,j} = D_{1,i,j} - a(i,j) \times V \text{ ----------------- } (5.7)$$

and b) new horizontal edge difference vector:

$$W_{2,i,j} = D_{2,i,j} - a(i,j) \times V \quad \text{---------------(5.8)}$$

For simplicity of notation, the indices i and j will now be dropped. The sum of the square magnitudes of these vectors is

$$e = \sum_{p=1}^{2} | D_p - a V |^2 \quad \text{-------------------------(5.9)}$$

where e is function of the estimated dc coefficient a. It can be shown (see Appendix B) that e is minimum when

$$a = (1/2) \times \sum_{p=1}^{2} \sum_{i=1}^{n} d(p,i) \quad \text{----------------(5.10)}$$

where d(p,i) is the i th element of the vector D_p. Equation 5.10 implies that setting the new dc coefficient a equal to one half of the sum of the grey level difference d(p,i) at the two edges minimizes the edge difference. It should be noted that the grey level differences, d(p,i), are computed from pels in block (i-1,j) and block (i,j-1) whose levels have been adjusted in accordance with dc coefficients estimated previously. In general, it can be shown that

$$a(i,j) = (1/P) \times \sum_{p=1}^{P} \sum_{i=1}^{n} d(p,i) \quad \text{-------------(5.11)}$$

where P is the number of edges taken in consideration.

5.4 ROW ESTIMATION

5.4.1 Description

In this method, the dc coefficients in the first row are estimated using element estimation and the pels in the first row are adjusted accordingly. The next row of dc coefficients is then determined as the set of dc coefficients which minimizes the sum of the square magnitudes of the edge difference vectors between the first and second rows, and also those between the individual blocks in the second row. Similarly, the dc coefficients in the i+1 th row are estimated from the i th row until all the dc coefficients are found. Finally, the overall picture grey level is brought within the desired range for display.

5.4.2 Theoretical development

The method of estimation of the dc coefficients in the i th row from those in the i-1 th row is given. Terms defined by equations 5.1 to 5.4 are used. Also, it is assumed that the receiver has the following information:

(i) $[\ u(k,l)\ ]_{i,j}$: the (i,j)th block of pels having zero dc level.

(ii) $[\ v(k,l)\ ]_{i-1,j}$: the (i-1,j)th block of pels whose dc levels have been adjusted according to the estimated dc coefficients.

$$[\ v(k,l)\ ]_{i-1,j} = [\ u(k,l)\ ]_{i-1,j} + b_{i-1,j} \times V \times V^{t} \quad \text{----(5.12)}$$

where $b_{i-1,j}$ is the estimated dc coefficient for block (i-1,j).

Now, we are to estimate the N-dimensional dc coefficient vector

$$A = [\, a_1 , \ldots, a_N \,]$$

from $[\, u(k,l) \,]_{i,j}$ and $[\, v(k,l) \,]_{i-1,j}$. We define vertical and horizontal edge difference vectors for block (i,j) as shown in Fig.5.3.

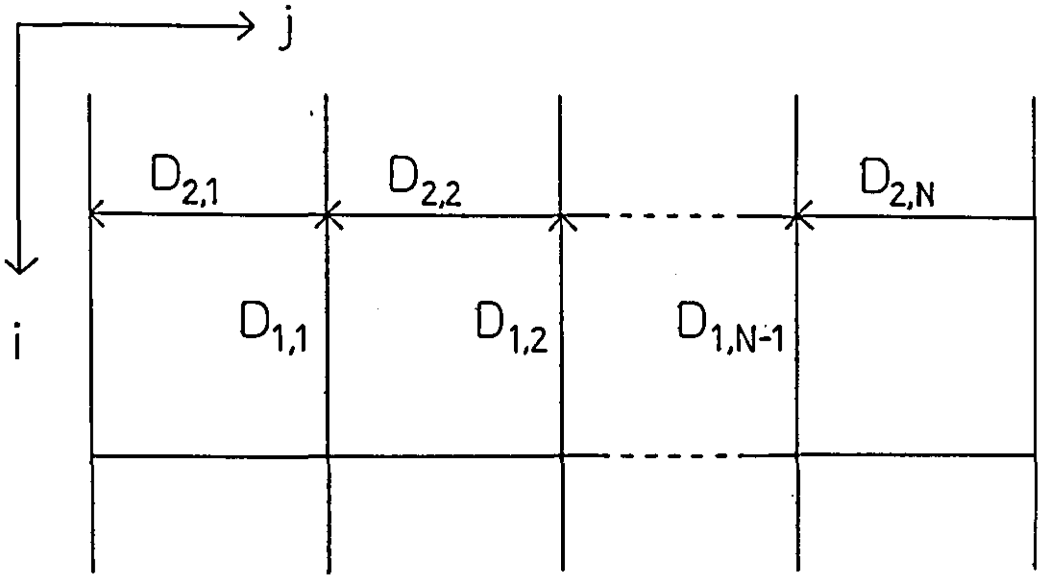


Fig.5.3 The vertical and horizontal edge difference vectors considered in ROW ESTIMATION.

The vertical edge difference vector between the j th block an j+1 th block in the i th row of blocks is

$$D_{1,j} = \begin{bmatrix} u(1,n)_{i,j} & - & u(1,1)_{i,j+1} \\ u(2,n)_{i,j} & - & u(2,1)_{i,j+1} \\ \cdots\cdots\cdots & . & \cdots\cdots\cdots\cdots \\ u(n,n)_{i,j} & - & u(n,1)_{i,j+1} \end{bmatrix} \quad j \varepsilon [1,N-1] \qquad \text{----(5.13)}$$

The horizontal edge difference vector between the (i-1,j)th block and (i,j) th block is

$$D_{2,j} = \begin{bmatrix} u(n,1)_{i,j} & - & v(1,1)_{i-1,j} \\ u(n,2)_{i,j} & - & v(1,2)_{i-1,j} \\ \cdots\cdots\cdots & \cdot & \cdots\cdots\cdots \\ u(n,n)_{i,j} & - & v(1,n)_{i-1,j} \end{bmatrix} \quad j \in [1,N] \qquad \text{-----(5.14)}$$

If the pels in the i th row are adjusted by the N dc coefficients $a_1, a_2, a_3, \ldots, a_N$, then the edge difference vectors $D_{1,j}$ and $D_{2,j}$ are changed to $W_{1,j}$ and $W_{2,j}$ respectively:

$$W_{1,j} = D_{1,j} + (a_j - a_{j+1}) \times V$$

$$W_{2,j} = D_{2,j} + a_j \times V \qquad \text{-----------------(5.15)}$$

Therefore, the sum of the squares of the magnitudes of these edge difference vectors becomes

$$e = \sum_{j=1}^{N-1} \left| D_{1,j} + (a_j - a_{j+1}) \times V \right|^2$$

$$+ \sum_{j=1}^{N} \left| D_{2,j} + a_j \times V \right|^2 \qquad \text{-----------------(5.16)}$$

or

$$e = \sum_{p=1}^{2} \sum_{j=1}^{N} \left| D_{p,j} + [\,R\,]_{p,j} \times A \right|^2 \qquad \text{-----------(5.17)}$$

and the n by N dimension matrices $[\,R\,]_{p,j}$ are as follows:

(A) When p=1 (the vertical edge difference is being considered)

j	$[R]_{1,j}$
1	[V, -V, 0, ..., 0]
2	[0, V, -V, 0,..., 0]
.	....
.	....
N-1	[0, ..., 0, V, -V]
N	[0, ..., 0]

(B) When p=2 (the horizontal edge difference is being considered)

j	$[R]_{2,j}$
1	[V, 0, ..., 0]
2	[0, V, 0, ..., 0]
.	....
.	....
N	[0, ..., 0, V]

it can be shown (see Appendix A) that e is minimum when

$$A = - [\,RR\,]^{-1} \times C \qquad \text{-----------------------------}(\ 5.18\ )$$

$$\text{where } [\,RR\,] = \sum_{p=1}^{2} \sum_{j=1}^{N} [\,R\,]^{t}_{p,j} \; [\,R\,]_{p,j} \qquad \text{--------------}(\ 5.19\ )$$

and $$C = \sum_{p=1}^{2} \sum_{j=1}^{N} [\ R\ ]^{t}_{p,j} \quad D_{p,j} \qquad \text{----------------}(\ 5.20\ )$$

As

$$\sum_{j=1}^{N} [\ R\ ]^{t}_{1,j} \quad [\ R\ ]_{1,j}$$

$$= \ (1/n) \ \ x \ \left[\begin{array}{ccccccccc} 1 & -1 & 0 & 0 & & & & & \\ -1 & 2 & -1 & 0 & & & & & \\ 0 & -1 & 2 & -1 & 0 & & & & \\ & 0 & \cdot & \cdot & \cdot & 0 & & & \\ & & \cdot & \cdot & \cdot & \cdot & \cdot & & \\ & & & \cdot & \cdot & \cdot & \cdot & \cdot & \\ & & & & \cdot & \cdot & \cdot & \cdot & 0 \\ & & & & & 0 & -1 & 2 & -1 \\ & & & & & & 0 & -1 & 1 \end{array} \right]$$

$$\sum_{j=1}^{N} [\ R\ ]^{t}_{2,j} \quad [\ R\ ]_{2,j}$$

$$= \ (1/n) \ \ x \ \left[\begin{array}{ccccccc} 1 & & & & & & \\ & 1 & & & & & \\ & & 1 & & & & \\ & & & \cdot & & & \\ & & & & \cdot & & \\ & & & & & 1 & \\ & & & & & & 1 \end{array} \right]$$

Adding these two terms together, we have

$$[\ RR\ ] = (1/n) \ x \ \left[\begin{array}{ccccccccc} 2 & -1 & 0 & 0 & & & & & \\ -1 & 3 & -1 & 0 & 0 & & & & \\ 0 & -1 & 3 & -1 & 0 & & & & \\ & 0 & -1 & 3 & -1 & 0 & & & \\ & & \cdot & \cdot & \cdot & \cdot & \cdot & & \\ & & & \cdot & \cdot & \cdot & \cdot & \cdot & \\ & & & & 0 & -1 & 3 & -1 & 0 \\ & & & & & 0 & -1 & 3 & -1 \\ & & & & & & 0 & -1 & 2 \end{array} \right]$$

The elements of vector C are summarized in table 5.1 and 5.2 where the ith element of vector C is

$$c_i = c_{i,1} + c_{i,2} \quad \text{-----------------------------}(\ 5.21\ )$$

also,

$$d_{p,j} = \sum_{q=0}^{N-1} d_{p,j,q}$$

and $d_{p,j,q}$ is the q th element of the vector $D_{p,j}$.

j	$c_{1,1}$	$c_{2,1}$	$c_{3,1}$		$c_{N-1,1}$	$c_{N,1}$
1	$d_{1,1}$	$-d_{1,1}$				
2		$d_{1,2}$	$-d_{1,2}$			
3			$d_{1,3}$			
~	~	~	~	~	~	~
N-2					$-d_{1,N-2}$	
N-1					$d_{1,N-1}$	$-d_{1,N-1}$

Table 5.1 : $c_{i,1}$ (equation 5.21) which is equal to the summation of the terms, $d_{1,j}$, below it.

j	$c_{1,2}$	$c_{2,2}$	$c_{3,2}$	~~~~	$c_{N-1,2}$	$c_{N,2}$
1	$d_{2,1}$			~~~~		
2		$d_{2,2}$		~~~~		
3			$d_{2,3}$	~~~~		
~	~	~	~	~~~~	~	~
N-1				~~~~	$d_{2,N-1}$	
N				~~~~		$-d_{2,N}$

Table 5.2 : $c_{i,2}$ (equation 5.21) which is equal to the summation of the terms, $d_{2,j}$, below it.

5.5 PLANE ESTIMATION

5.5.1 Description

In this method, the N x N blocks of pels having zero dc level are grouped into (N/2) x (N/2) groups as shown in Fig.5.4. The difference over all four edges within each group is then minimized by inserting new dc coefficients. Therefore, each group of four blocks can be regarded as one single larger block of (2xn) by (2xn) pels, and there are (N/4) x (N/4) of them in the picture. The same process can then be applied to minimize the difference at the four edges and to form even larger blocks. This process will continue until the whole picture is merged into one block. Finally, the restored picture grey levels are brought within the desired range for display.

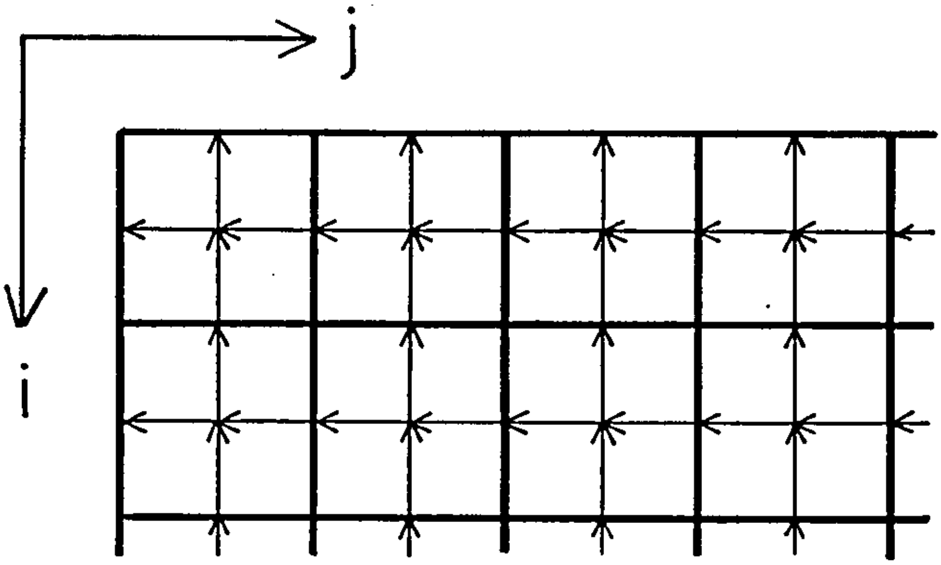


Fig.5.4 The way that the NxN blocks of pels are grouped into (N/2)x(N/2) groups in PLANE ESTIMATION.

5.5.2 Theoretical development

Without losing generality, we may consider the four block group at the top left corner as shown in Fig.5.5. This group is the same as a picture which contains only two by two blocks, each block of pels having zero dc level. Define the four edge difference vectors as

$$D_1 = \begin{bmatrix} u(n,1)_{1,1} & - & u(1,1)_{2,1} \\ u(n,2)_{1,1} & - & u(1,2)_{2,1} \\ \cdots\cdots\cdots & . & \cdots\cdots\cdots \\ u(n,n)_{1,1} & - & u(1,n)_{2,1} \end{bmatrix}$$

$$D_2 = \begin{bmatrix} u(1,n)_{2,1} - u(1,1)_{2,2} \\ u(2,n)_{2,1} - u(2,1)_{2,2} \\ \cdots\cdots\cdots \\ u(n,n)_{2,1} - u(n,1)_{2,2} \end{bmatrix}$$

$$D_3 = \begin{bmatrix} u(n,1)_{1,2} - u(1,1)_{2,2} \\ u(n,2)_{1,2} - u(1,2)_{2,2} \\ \cdots\cdots\cdots \\ u(n,n)_{1,2} - u(1,n)_{2,2} \end{bmatrix}$$

$$D_4 = \begin{bmatrix} u(1,n)_{1,1} - u(1,1)_{1,2} \\ u(2,n)_{1,1} - u(2,1)_{1,2} \\ \cdots\cdots\cdots \\ u(n,n)_{1,1} - u(n,1)_{1,2} \end{bmatrix} \qquad \text{-----------------(5.22)}$$

If the pels in blocks (2,1), (2,2) and (1,2) are adjusted in accordance with the three estimated dc coefficients a_1, a_2 and a_3, then the four edge difference vectors will become

$$W_1 = D_1 - V \times a(2,1)$$

$$W_2 = D_2 + V \times (a(2,1)-a(2,2))$$

$$W_3 = D_3 + V \times (a(1,2)-a(2,2))$$

$$W_4 = D_4 - V \times a(1,2) \qquad \text{-------------------(5.23)}$$

We require the dc coefficient vector A

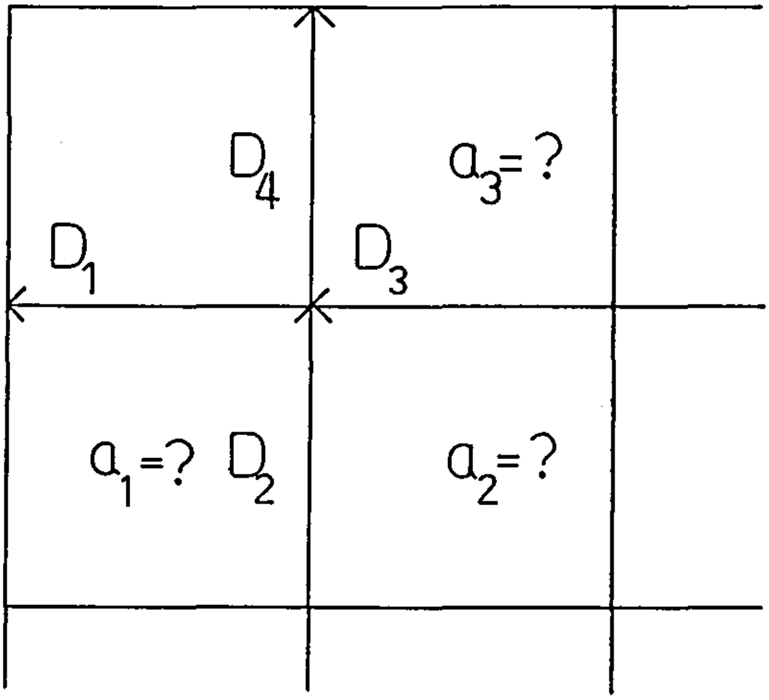


Fig.5.5 The four edge difference vectors considered in PLANE ESTIMATION.

$$A = [\, a(2,1),\ a(2,2),\ a(1,2)\,]^t \triangleq [\, a_1 ,\ a_2 ,\ a_3 \,]^t \qquad \text{------------------(5.24)}$$

such that e , the sum of the squares of the four new edge difference vector magnitudes, is a minimum.

$$e = \sum_{p=1}^{4} \left| D_p + [\, R \,]_p A \right|^2 \qquad \text{----------------(5.25)}$$

Equations 5.23 to 5.25 imply that

$$
\begin{aligned}
[\,R\,]_1 &= [\,-V,\;\; 0,\;\; 0\,] \\
[\,R\,]_2 &= [\;\; V,\; -V,\;\; 0\,] \\
[\,R\,]_3 &= [\;\; 0,\; -V,\;\; V\,] \\
[\,R\,]_4 &= [\;\; 0,\;\; 0,\; -V\,]
\end{aligned}
\qquad \text{--------------------}(\ 5.26\ )
$$

It can be shown (see appendix A) that the dc coefficients that minimize e are given by

$$A = [\,RR\,]^{-1} \times C \qquad \text{---------------------}(\ 5.27\ )$$

$$\text{where} \quad [\,RR\,] = \sum_{p=1}^{4} [\,R\,]_p^t \times [\,R\,]_p$$

$$= (1/n) \times \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix} \qquad \text{--------------}(\ 5.28\ )$$

$$\text{and} \quad [\,RR\,]^{-1} = (n/4) \times \begin{bmatrix} 3 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 3 \end{bmatrix} \qquad \text{--------------}(\ 5.29\ )$$

$$\text{Also,} \quad C = \sum_{p=1}^{4} [\,R\,]_p^t \times D_p \qquad \text{---------------------}(\ 5.30\ )$$

$$\text{Let} \quad D_p = \begin{bmatrix} d(1)_p \\ d(2)_p \\ \cdots\cdot \\ d(n)_p \end{bmatrix} \qquad \text{---------------}(\ 5.31\ )$$

$$C = \begin{bmatrix} c(1) \\ c(2) \\ c(3) \end{bmatrix} \text{----------------}(5.32)$$

$$s(p) = (1/n) \times \sum_{i=1}^{n} d(i)_{p} \text{---------}(5.33)$$

Equations 5.26, 5.30, 5.31, 5.32 and 5.33 indicate that

$$\begin{aligned} c(1) &= -s(1) + s(2) \\ c(2) &= -s(2) - s(3) \\ c(3) &= s(3) - s(4) \quad \text{-----------------}(5.34) \end{aligned}$$

and equations 5.27, 5.29 and 5.34 indicate that

$$\begin{aligned} a(1) &= (s(4) + s(3) - s(2) + 3 \times s(1))/4 \\ a(2) &= (s(4) + s(3) + s(2) + s(1))/2 \\ a(3) &= (3 \times s(4) - s(3) + s(2) + s(1))/4 \end{aligned}$$

$$\text{------------------------------}(5.35)$$

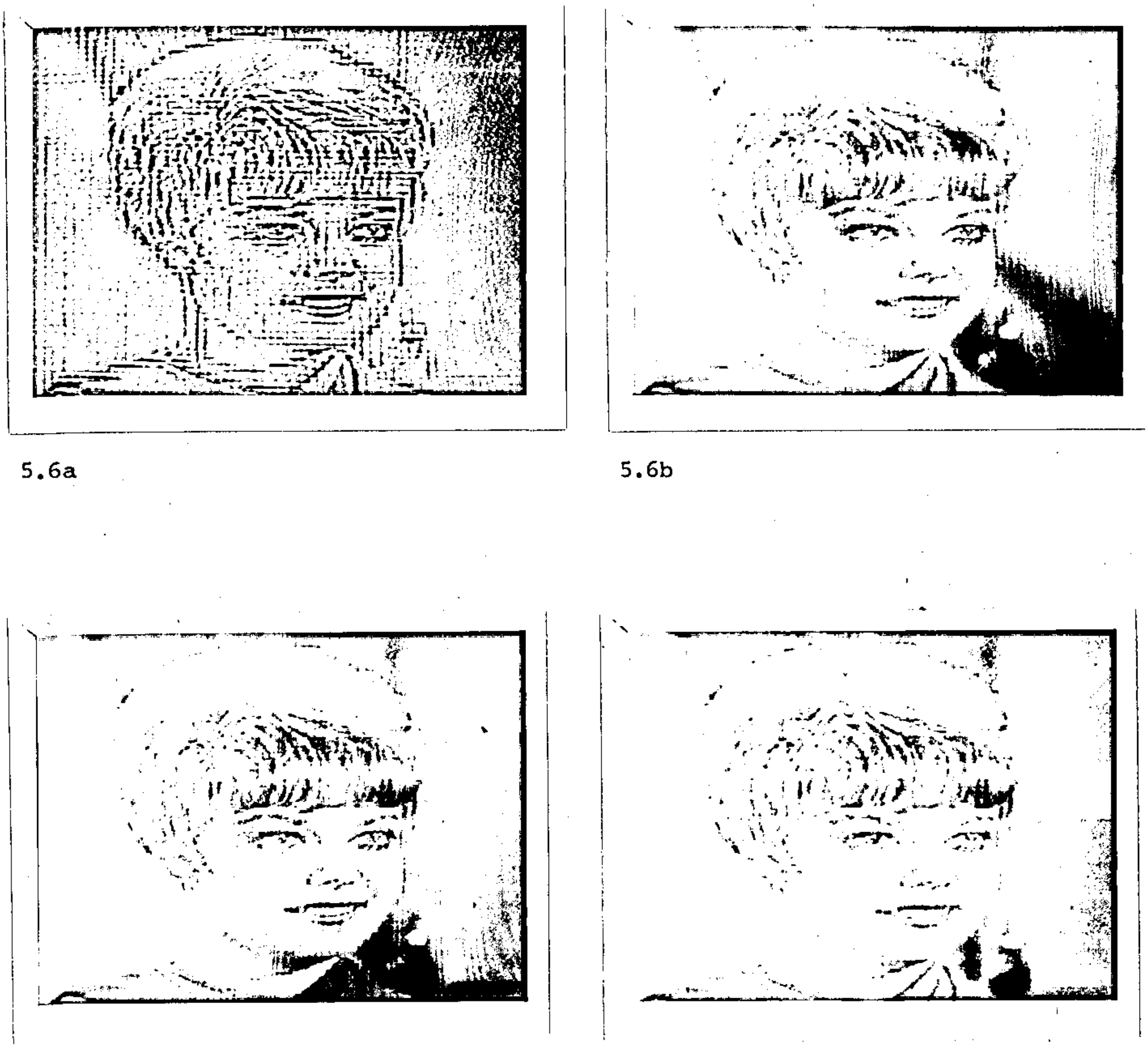
5.6 EXPERIMENTAL RESULTS

Evaluation of the restoration schemes was carried out using computer simulation. A picture was first divided into blocks of size n by n. Each block was then transformed using the Walsh transform, and the dc coefficient set equal to zero. All blocks were then inverse transformed to return to the picture domain. Fig.5.6a to Fig.5.14a are the pictures after inverse transformation with dc coefficients set to zero.

The three dc coefficient restoration schemes were then applied to obtain the restored pictures as well as the sets of estimated dc coefficients. These procedures were repeated for block sizes 4x4, 8x8 and 16x16, and for the pictures 'Girl', 'House' and 'Testcard' described in section 4.5.2.1. No coefficient quantization was undertaken.

For the 'Girl' picture, the restored pictures are shown in Fig.5.6 to 5.8. When the block size is 4x4, there are severe edging effects in all the three restored pictures. Furthermore,the accumulation of error due to each estimation produces impairment effects along the direction of estimation. In the picture restored by Element Estimation, if a block is very bright or very dark, this brightness or darkness tends to diffuse diagonally from top left to bottom right. In the picture restored by Row Estimation, the diffusion runs vertically from top to bottom and is less severe than that given by Element Estimation. In contrast, the picture restored by Plane Estimation shows no such effect. However, accumulation of estimation errors makes the edging effects more prominent as the block size increases.

When the block size is 8 x 8, the pictures restored by Element and Row Estimation still have edging effects but not the apparent diffusion effect. Noticeable edging effects still remain in the picture restored using Plane Estimation. When the block size is 16x16, Row Estimation restored the picture without perceptible error whilst Element Estimation produced a reasonably good picture. Again, there are noticeable edging effects in the picture restored by Plane Estimation.



5.6a

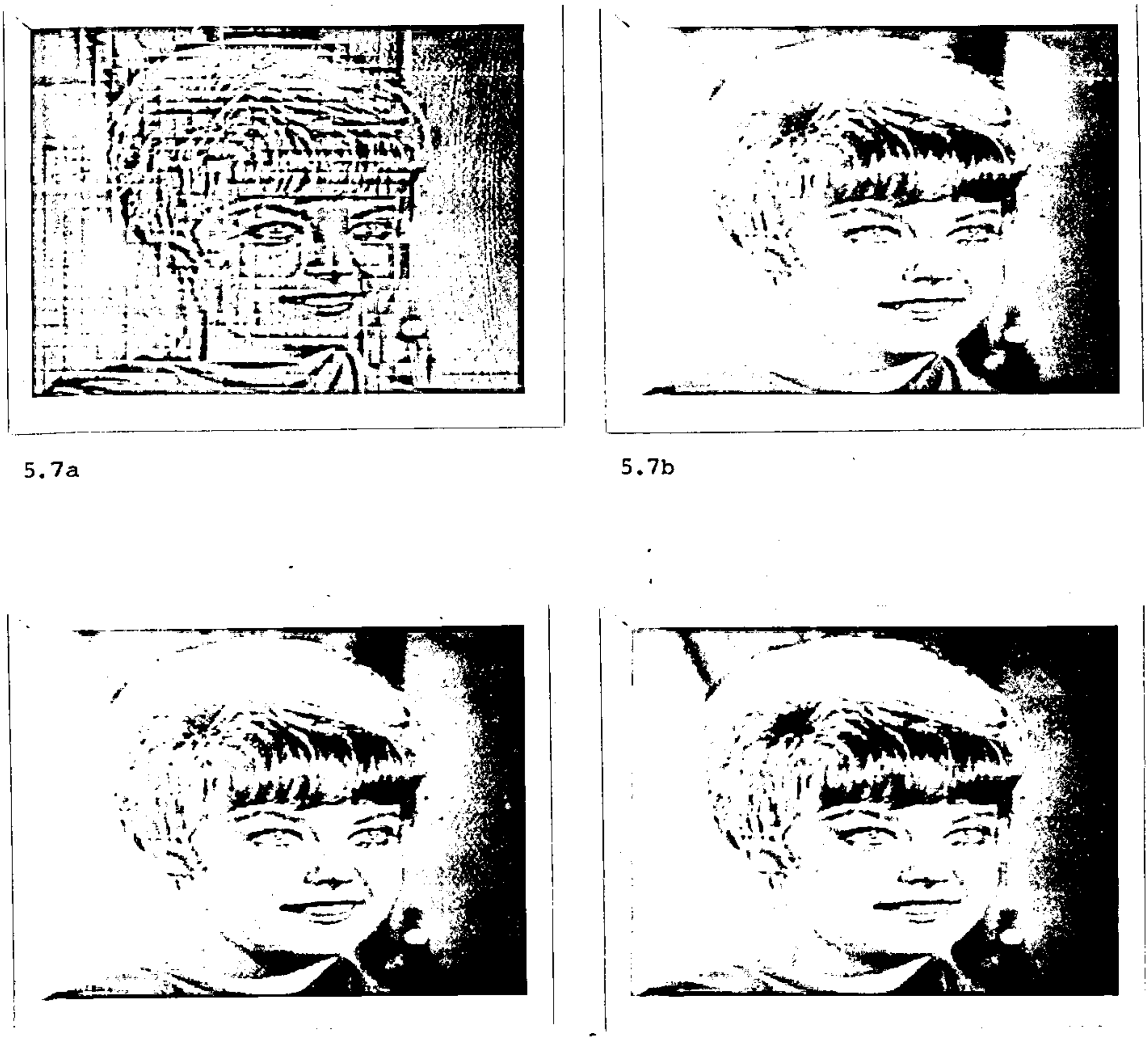
5.6b

5.6c

5.6d

Fig.5.6

The 'girl' picture with dc coefficients a) set equal to zero, b) restored with ELEMENT ESTIMATION, c) restored with ROW ESTIMATION and d) restored with PLANE ESTIMATION for block size 4x4.



5.7a

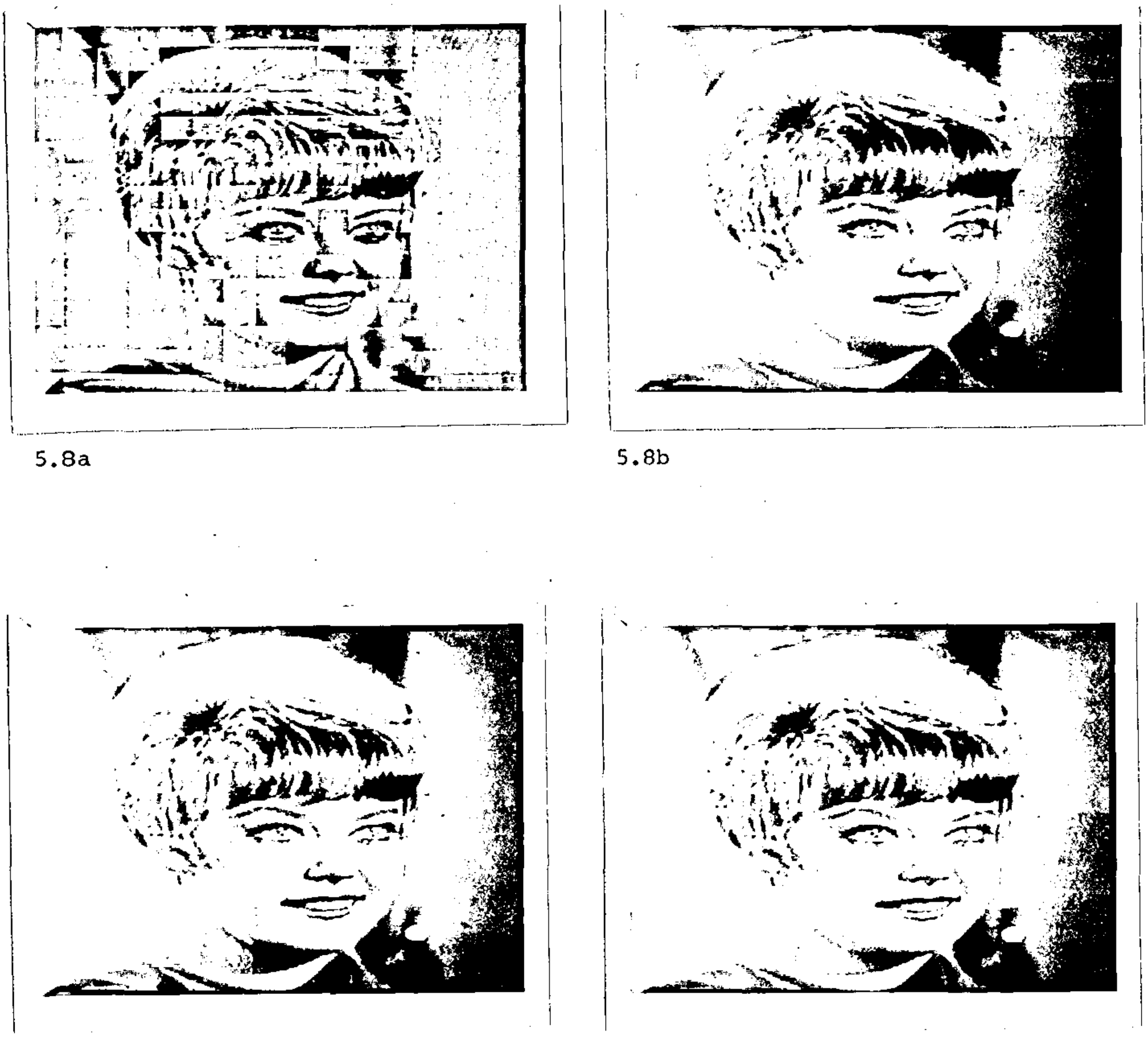
5.7b

5.7c

5.7d

Fig.5.7

The 'girl' picture with dc coefficients a) set equal to zero, b) restored with ELEMENT ESTIMATION, c) restored with ROW ESTIMATION and d) restored with PLANE ESTIMATION for block size 8x8.



5.8a

5.8b

5.8c

5.8d

Fig.5.8

The 'girl' picture with dc coefficients a) set equal to zero, b) restored with ELEMENT ESTIMATION, c) restored with ROW ESTIMATION and d) restored with PLANE ESTIMATION for block size 16x16.

For the pictures 'House' and 'Testcard' which contain many regions of high activity, the restored pictures are shown in Fig.5.9 to Fig.5.14. These pictures show that all the three dc coefficient restoration schemes fail to produce sets of dc coefficients that result in satisfactory pictures. All the pictures restored from the 'House' and 'Testcard' have severe edging effects, compared with the pictures restored from the 'Girl' picture. Reasonable results can only be obtained when the block sizes are as large as 16x16 -- i.e. where the dc coefficients require only a small fraction of the total number of bits.

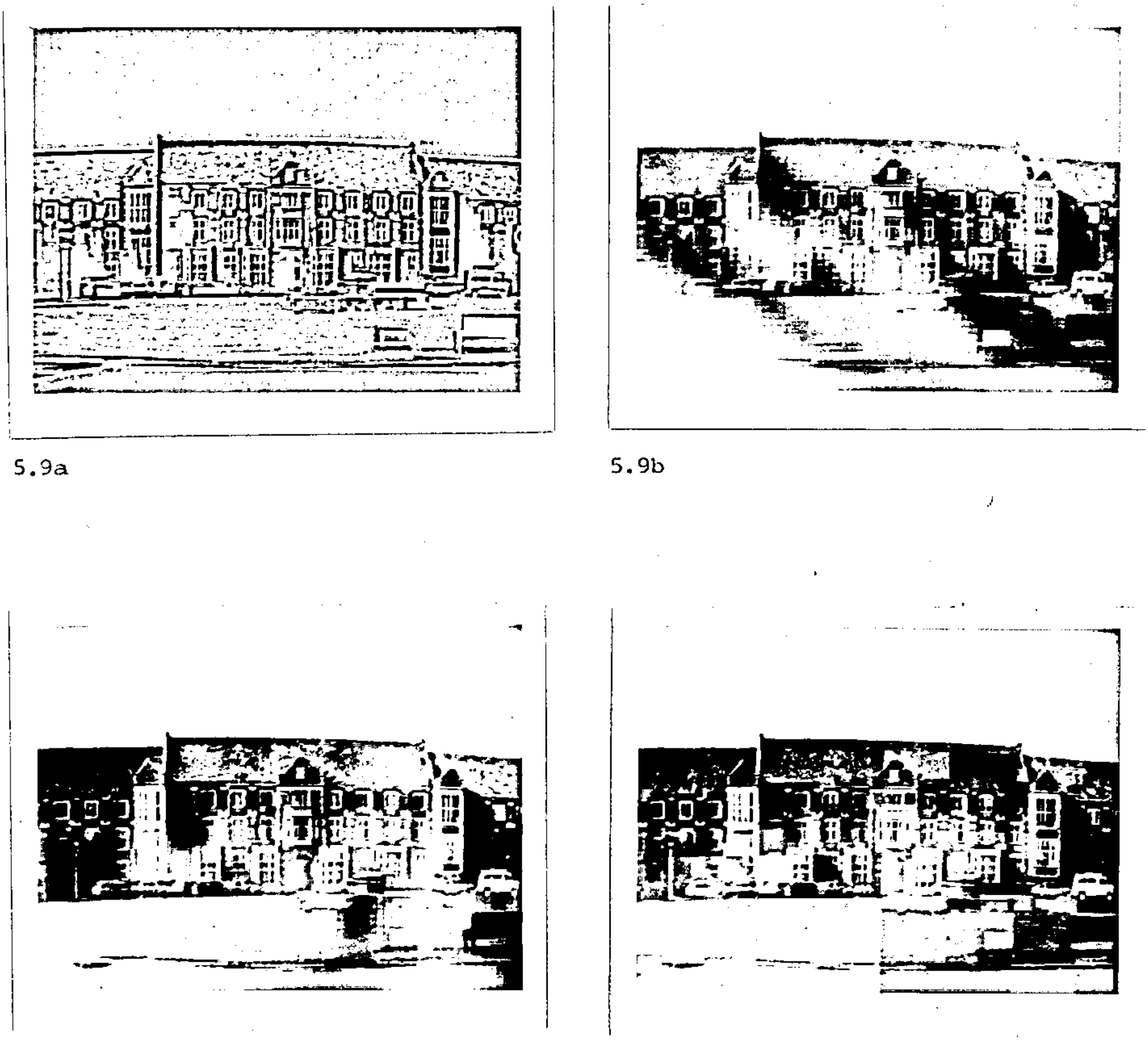
5.7 CONCLUSIONS

In a transform coding system using a small block size, a large proportion of the coding bits is required by the dc coefficients. Three schemes are proposed, in which the dc coefficients are not transmitted, but estimated at the receiver. This allows a reduction in bit rate, and the possibility of eliminating the serious effect of channel error on those coefficients. Computer simulation on real pictures showed that when using a large block size or a low activity picture, satisfactory results can be obtained. Also, of the three estimation schemes considered, Row Estimation gives the best result, followed by Element Estimation and finally Plane Estimation.

5.8 NOTE ON PUBLICATION

A paper based on the material described in this chapter and entitled 'DC Coefficient Restoration in Transform Image Coding' has been submitted to IEE Proceedings. The paper is jointly authored with R.J.Clarke.

Also, a U.K. patent application (No. 8229420) entitled 'Image Transmission', in co-authorship with Mr R J Clarke and Dr R C Nicol, has been made to claim originality of invention on techniques developed in Chapter 5.



5.9a

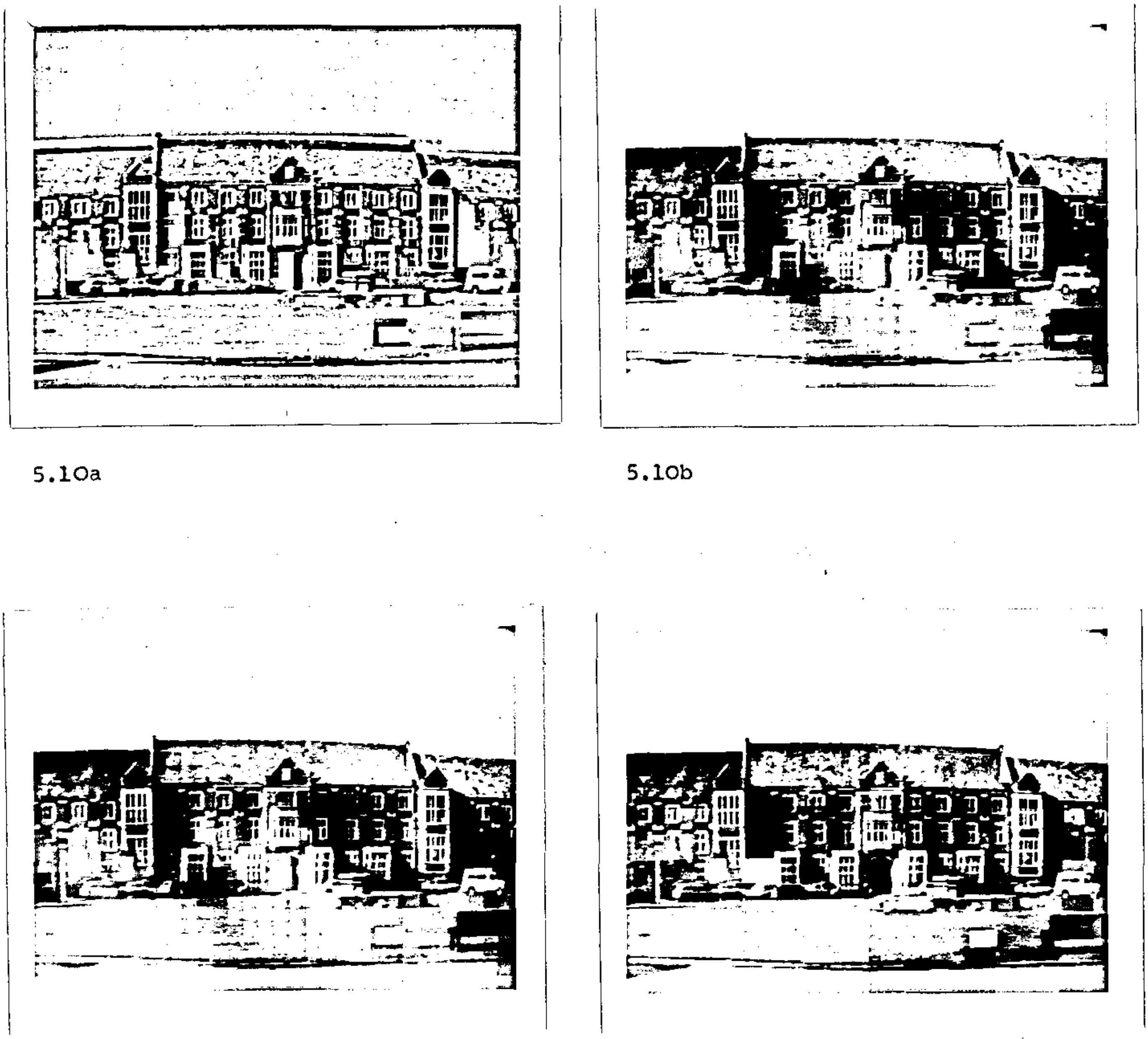
5.9b

5.9c

5.9d

Fig.5.9

The 'house' picture with dc coefficients a) set equal to zero, b) restored with ELEMENT ESTIMATION, c) restored with ROW ESTIMATION and d) restored with PLANE ESTIMATION for block size 4x4.



5.10a

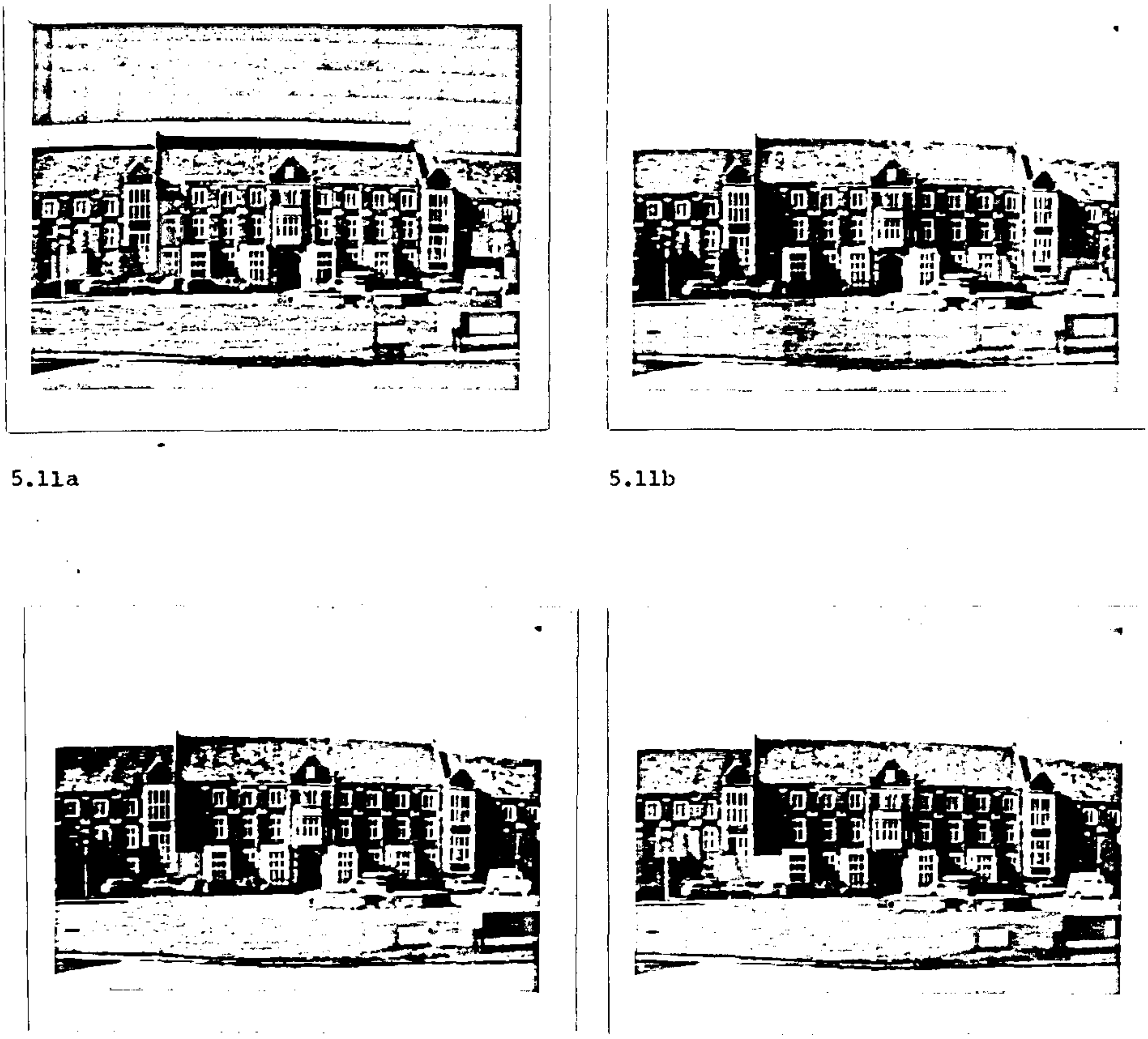
5.10b

5.10c

5.10d

Fig.5.10

The 'house' picture with dc coefficients a) set equal to zero, b) restored with ELEMENT ESTIMATION, c) restored with ROW ESTIMATION and d) restored with PLANE ESTIMATION for block size 8x8.



5.11a

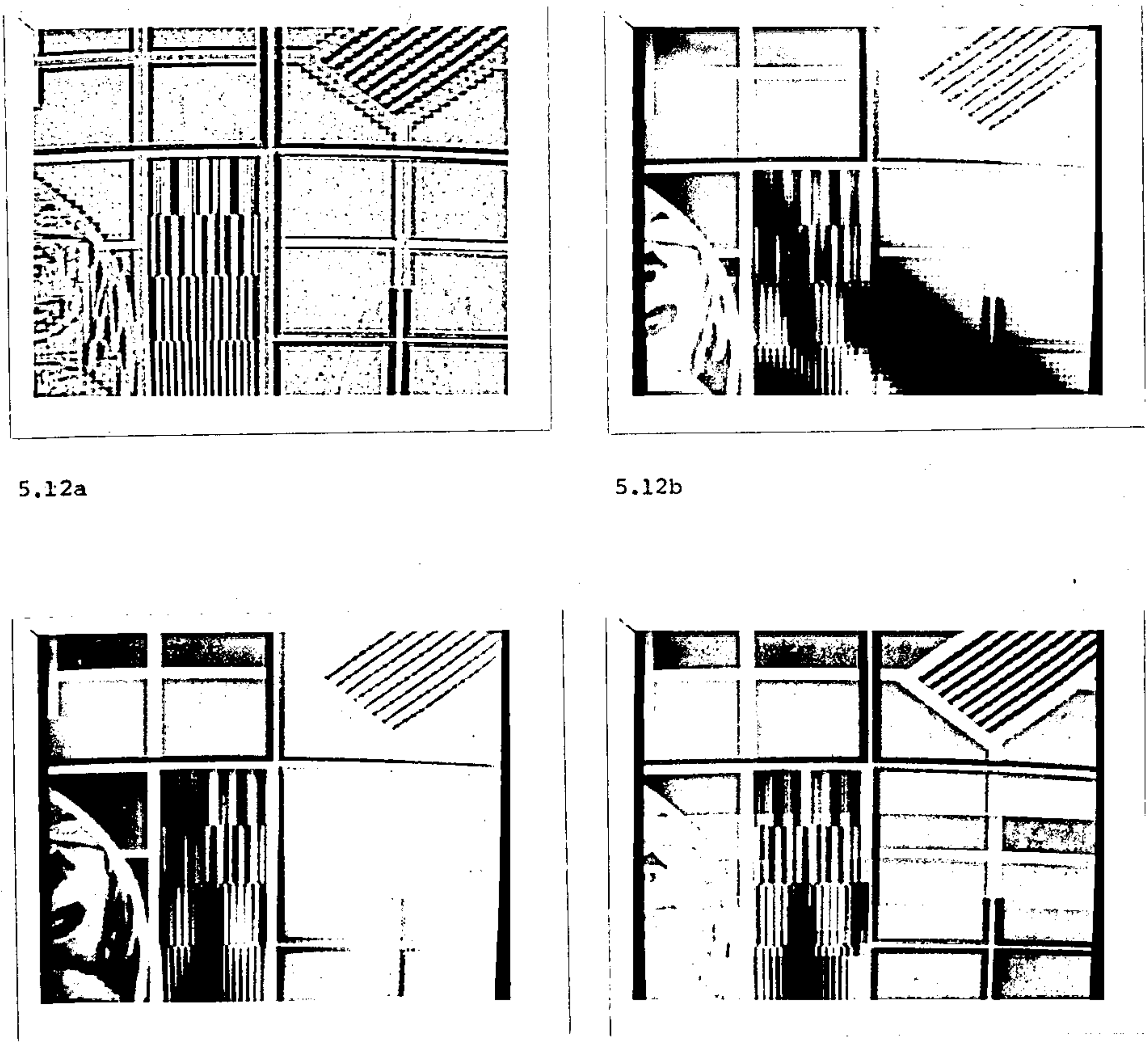
5.11b

5.11c

5.11d

Fig.5.11

The 'house' picture with dc coefficients a) set equal to zero, b) restored with ELEMENT ESTIMATION; c) restored with ROW ESTIMATION and d) restored with PLANE ESTIMATION for block size 16x16.



5.12a

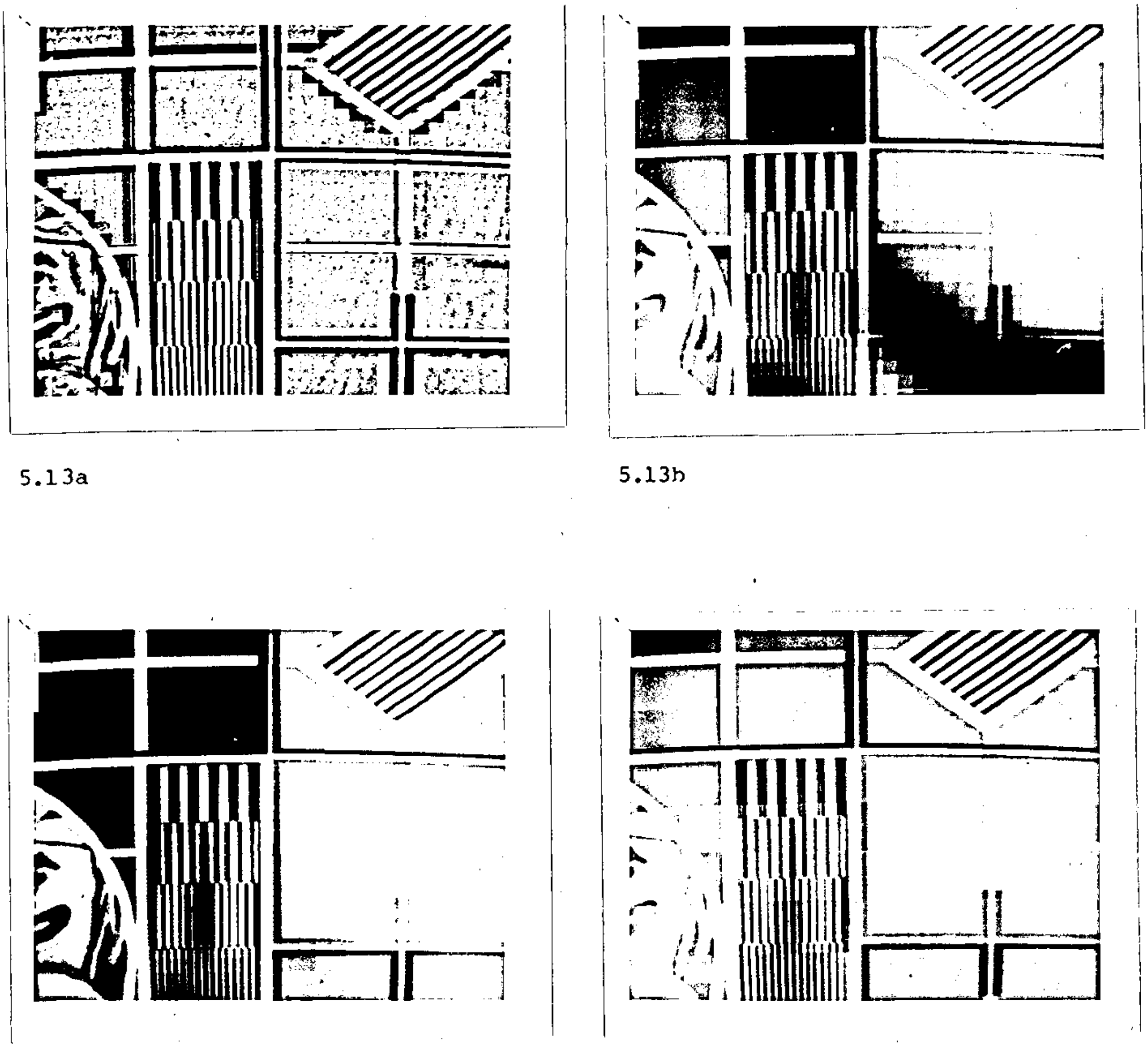
5.12b

5.12c

5.12d

Fig.5.12

The 'BBC testcard' with dc coefficients a) set equal to zero, b) restored with ELEMENT ESTIMATION, c) restored with ROW ESTIMATION and d) restored with PLANE ESTIMATION for block size 4x4.



5.13a

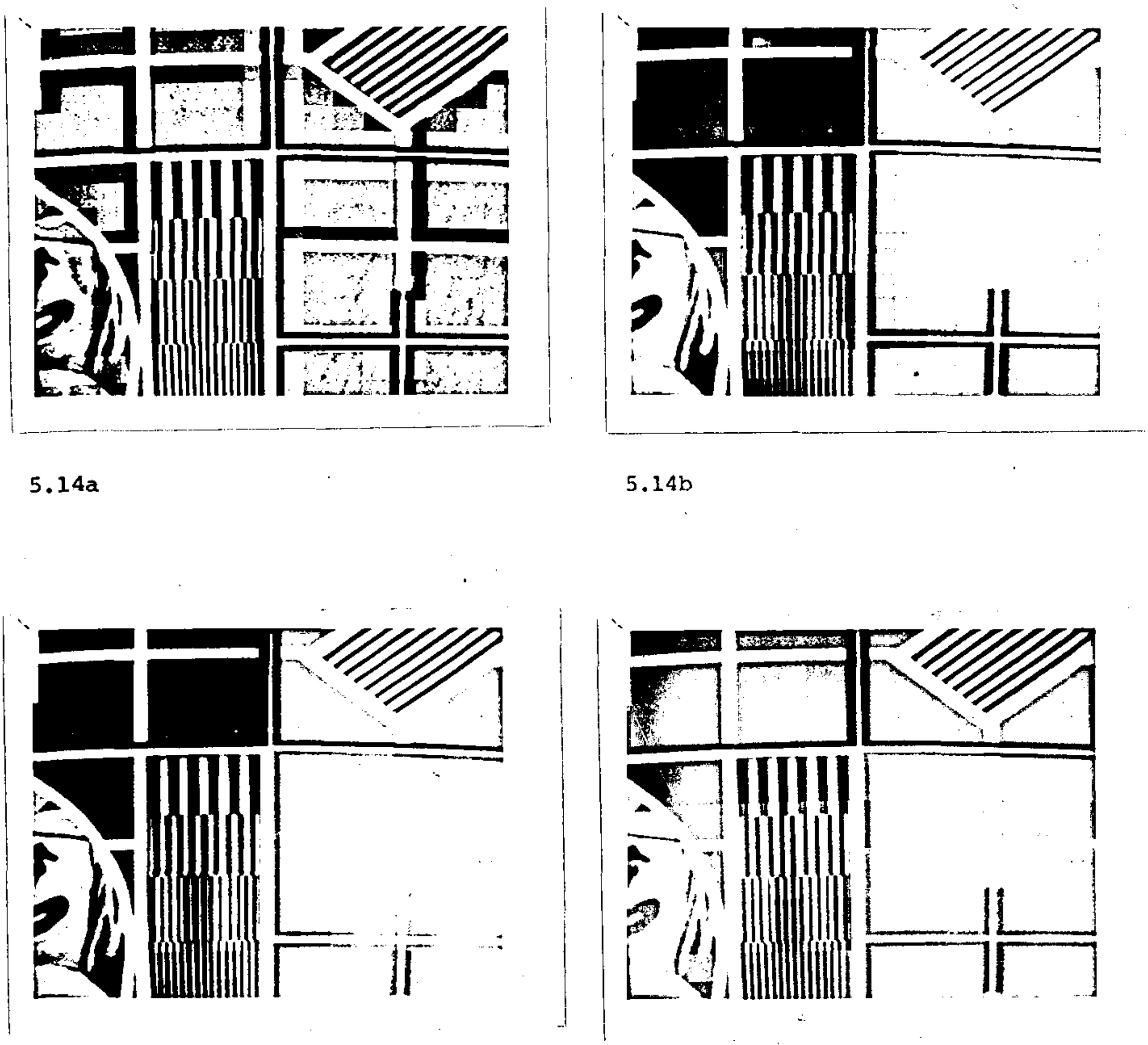
5.13b

5.13c

5.13d

Fig.5.13

The 'BBC testcard' with dc coefficients a) set equal to zero, b) restored with ELEMENT ESTIMATION, c) restored with ROW ESTIMATION and d) restored with PLANE ESTIMATION for block size 8x8.



5.14a

5.14b

5.14c

5.14d

Fig.5.14

The 'BBC testcard' with dc coefficients a) set equal to zero, b) restored with ELEMENT ESTIMATION, c) restored with ROW ESTIMATION and d) restored with PLANE ESTIMATION for block size 16x16.

CHAPTER SIX

RECAPITULATION AND SUGGESTIONS FOR FUTURE WORK

6.1 INTRODUCTION

At present, transmission of a 256 x 256 8-bit/pel picture over the British telephone network (1.2kbps) takes about 7.5 minutes. The transmission time, however, can be reduced to about 27 seconds by using transform coding. It achieves data compression first by transformation of the image into arrays of transform coefficients such that most of the energy is packed into a few coefficients. The use of appropriate bit allocation, optimal quantisation and a robust adaptive scheme then allows the image to be represented at 0.5 bit/pel with acceptable picture quality.

At the beginning of the research program, interblock redundancy (section 1.3.3) was examined. A technique which allows the dc coefficients to be estimated at the receiver was found. Three dc coefficient estimation schemes based on this technique were proposed and tested on real images. Attention then was concentrated on another important aspect of transform coding -- the transformation. When looking into the symmetry properties of the Walsh transform, a concept known as dyadic symmetry was discovered. This led to the development of the high correlation transform (HCT) and the low correlation transform (LCT) as well as a unified matrix treatment for Walsh matrices. For ease of discussion, these results have been presented in the order of a) a unified matrix treatment of the Walsh matrix, b) the HCT and LCT and finally c) the dc coefficient restoration schemes.

6.2 DYADIC SYMMETRY AND ITS APPLICATIONS TO THE WALSH TRANSFORM THEORY

The concept of dyadic symmetry described in chapter three provides a common framework for most areas of interest concerning Walsh transforms. These include a) Walsh matrix generation, b) fast computational algorithms and c) conversion of Walsh transform coefficients from one ordering to another.

Many solutions of these problems have in fact been found by different researchers using quite different approaches. The concept of dyadic symmetry provides simple and straightforward derivations of all the results. It is a viable alternative to the unified matrix treatment of the Walsh transform put forward by Fino and Algazi [134]. As the whole theory relates to a binary field with 'logical and' and modulo two addition as operations, both theory and practical implementation are very simple.

First of all, the concept of dyadic symmetry led to the generation of the Walsh matrix of any ordering by equation 3.9

$$b_{ij} = j^t [S]^{-1} i \quad \text{-----------------------------}(3.9)$$

where b_{ij} is the (i,j)th element and [S] is the dyadic symmetry matrix of a particular ordering. The mxm binary dyadic symmetry matrices of the $2^m \times 2^m$ natural-ordered, dyadic-ordered and sequency-ordered Walsh matrices are respectively

$$[N] = \begin{bmatrix} 1 & & & & \\ & 1 & & & \\ & & \cdot & & \\ & & & \cdot & \\ & & & & 1 \end{bmatrix}$$

$$[D] = \begin{bmatrix} & & & & 1 \\ & & & 1 & \\ & & \cdot & & \\ & \cdot & & & \\ 1 & & & & \end{bmatrix}$$

$$[Z] = \begin{bmatrix} & & & & & 1 \\ & & & & 1 & 1 \\ & & & 1 & 1 & 1 \\ & \cdot & \cdot & \cdot & \cdot & \cdot \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

If i_N , i_D and i_Z are the i (row index) in b_{ij} of natural-ordered, dyadic ordered and sequency-ordered Walsh matrices, conversions between the indices are given by equation 3.23

$$[Z]^{-1} * i_Z \quad = \quad [D]^{-1} * i_D \quad = \quad [N]^{-1} * i_N \quad \text{------------(3.23)}$$

Fast computational algorithms were then obtained by repeated application of the 2^m-1 dyadic symmetry decompositions of a $2^m \times 2^m$ Walsh transform. Fig.6.1 shows the seven dyadic symmetry decompositions of an 8x8 Walsh transform. The repeated application of the 2^{m-1}th dyadic symmetry decompositions results in Shank's [124] fast computational

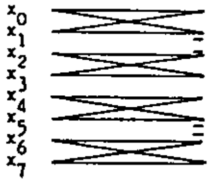
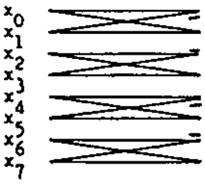
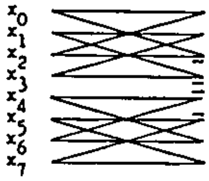
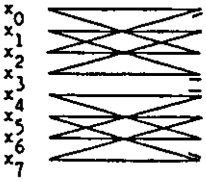
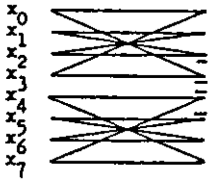
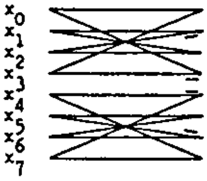
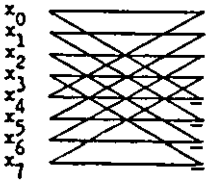
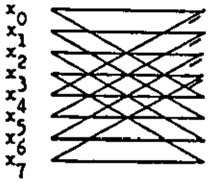
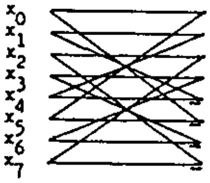
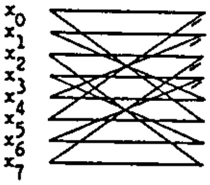
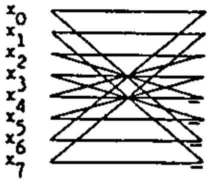
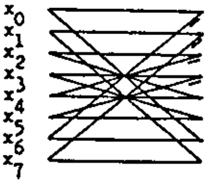
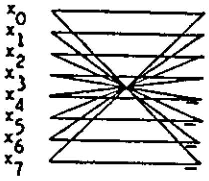
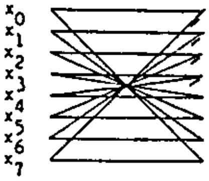
dyadic symmetry	U	V	[H]·U	[H]·V	signal flow diagrams
1	$x_0 + x_1$ $x_3 + x_2$ $x_4 + x_5$ $x_7 + x_6$	$x_0 - x_1$ $x_3 - x_2$ $x_4 - x_5$ $x_7 - x_6$	c_0 c_1 c_2 c_3	c_4 c_5 c_6 c_7	 
2	$x_0 + x_2$ $x_1 + x_3$ $x_6 + x_4$ $x_7 + x_5$	$x_0 - x_2$ $x_1 - x_3$ $x_6 - x_4$ $x_7 - x_5$	c_0 c_1 c_6 c_7	c_2 c_3 c_4 c_5	 
3	$x_0 + x_3$ $x_1 + x_2$ $x_6 + x_5$ $x_7 + x_4$	$x_0 - x_3$ $x_1 - x_2$ $x_6 - x_5$ $x_7 - x_4$	c_0 c_1 c_4 c_5	c_2 c_3 c_6 c_7	 
4	$x_0 + x_4$ $x_1 + x_5$ $x_2 + x_6$ $x_3 + x_7$	$x_0 - x_4$ $x_1 - x_5$ $x_2 - x_6$ $x_3 - x_7$	c_0 c_3 c_4 c_7	c_1 c_2 c_5 c_6	 
5	$x_0 + x_5$ $x_1 + x_4$ $x_2 + x_7$ $x_3 + x_6$	$x_0 - x_5$ $x_1 - x_4$ $x_2 - x_7$ $x_3 - x_6$	c_0 c_3 c_5 c_6	c_1 c_2 c_4 c_7	 
6	$x_0 + x_6$ $x_1 + x_7$ $x_2 + x_4$ $x_3 + x_5$	$x_0 - x_6$ $x_1 - x_7$ $x_2 - x_4$ $x_3 - x_5$	c_0 c_2 c_5 c_7	c_1 c_3 c_4 c_6	 
7	$x_0 + x_7$ $x_1 + x_6$ $x_2 + x_5$ $x_3 + x_4$	$x_0 - x_7$ $x_1 - x_6$ $x_2 - x_5$ $x_3 - x_4$	c_0 c_2 c_4 c_6	c_1 c_3 c_5 c_7	 

Fig.6.1 The seven dyadic symmetry decompositions.

algorithm (Fig.3.8) as well as those of Manz [126]. The repeated appli-application of the 1st dyadic symmmetry decompositions gives Larsen's [129] algorithm (Fig.3.9) and Fino's [127] algorithm. There are in fact many other fast computational algorithms. For example, Fig.3.10 and Fig.3.11 show two others which are obtained by the repeated application of the 2^m-1 th dyadic symmetry decompositions.

6.3 NEW TRANSFORMS

The two new transforms, HCT and LCT, can be used as substitutes for the Walsh transform. They have virtually the same computational requirements and implementation complexity as the Walsh transform, employing additions, substractions and binary shifts only but with an improved performance which lies between that of the Walsh transform and the discrete cosine transform (DCT).

Both transforms were obtained using a technique which can replace pair(s) of Walsh basis vectors by others to form a new set of linearly independent basis vectors. The HCT was designed to simulate the DCT whilst the LCT was found via a computer search. Fast computational algorithms have been developed for both forward and inverse HCT and LCT.

Tests on the HCT and LCT using the first-order Markov process of adjacent element correlation coefficient ρ show that the HCT has a better performance when ρ is close to unity whilst the LCT performs better when ρ is close to zero. Also, the two transforms have a better performance at a small block size than at a large block size. Tests on the two transforms using real images also give similar results. The HCT has a better performance on a low activity or highly correlated picture and the LCT performs better on a high activity or "artificial" picture (the 'BBC testcard', Fig.4.8c).

6.4 DC COEFFICIENT RESTORATION SCHEMES

The three dc coefficient restoration schemes provide a new approach for the exploitation of interblock redundancy. Unlike recursive block coding, the pinned sine transform or hybrid coding, the restoration schemes do not require additional computation at the transmitter, and the computational algorithms need only be implemented at the receiver.

Advantages provided by the dc coefficient restoration schemes are two-fold. First, they allow a further reduction in bit rate. In a two-dimensional transform coding system, the coding bits saved are 0.031, 0.125 and 0.5 bit/pel for block sizes 16x16, 8x8 and 4x4 respectively. Further, as dc coefficients are not transmitted, the serious degradation in the subjective quality of a picture due to the effect of channel errors on the dc coefficients is eliminated. It is proved that a dc coefficient restoration scheme is equvalent to a low sequency coefficient restoration scheme if the sequency-ordered Walsh transform is used. Tests of the three schemes on real images show that all the three schemes provide good results on the low activity 'girl' picture but fail to do so on the pictures 'house' and 'BBC testcard'. Also, it is found that Row Estimation has the best performance, then comes Element Estimation and Plane Estimation.

6.5 SUGGESTIONS FOR FUTURE WORK

The Walsh transform has a comparatively small computational requirement, sequency properties (p.39) which are similar to those of the familiar DFT, and it is statistically optimal for the class of processes defined by dyadic covariance matrices. To make use of these advantages, a special-purpose digital signal processor to compute the Walsh transform in the natural, dyadic and sequency orderings for time-series analysis has been proposed by Geadah and Corinthios [137].

As indicated in section three, the whole theory of dyadic symmetry relates to a binary field with 'logical and' and 'exclusive or' as operations. The derived algorithms based on this concept for the generation of Walsh matrices of different orderings, for the conversion of Walsh transform coefficients from one ordering to the other, and for the fast Walsh transform, all relate to this binary field and so can be implemented easily. Therefore, it is suggested that a Walsh transform processor designed using the theory of dyadic symmetry could have a simpler and more systematic arrangement than that of Geadah and Corinthios. This implies a cheaper and faster machine with more flexibility.

The two new transforms have not yet been used with any particular adaptive scheme. It is considered that an adaptive block classification coding scheme (section 1.3.2) employing the HCT for those blocks having a low activity index and the LCT for those blocks having a high activity index may produce even better results than a conventional block classification coding scheme using the same transform for all the blocks irrespective of activity. A dc coefficient restoration scheme could also be incorporated into the coding scheme to further reduce the bit rate.

REFERENCES & APPENDICES

REFERENCES

[1] Weszka, J.S., Dyer, C.R., Rosenfeld, A., "A Comparative Study of Texture Measures for Terran Classification," IEEE Trans. on Systems, Man, and Cybernetics, Vol.SMC-6, No.4., p.269-285, April 1976.

[2] Haralick, R.M., Shanmugam,K., Dinstein,I., "Textural Features for Image Classification," IEEE Trans. Syst.s Man and Cyber-ernetics, Vol.3, p.610-20, Nov.1973.

[3] Galloway, M.M., "Texture Classification using Grey Level Run Lengths, " Computer Graphics and Image Processing, Vol.4, p.172-179, June 1975.

[4] Davis, L.S., Johns, S.A., Aggarwal, J.K., "Texture Analysis using Generalized Co-occurrence Matrices," IEEE Trans. Vol.PAMI-1, No.3, p.251-259, July 79.

[5] Nishikawa, A., Massa, R.J., Mott-Smith, J.C., "Area properties of television pictures," IEEE Trans. Inform. Theory, Vol.IT-11, p.348-352, July 1965.

[6] Gattis, J.L., Wintz, P.A., "Automated techniques for data analysis and transmission," School of Electrical Engineering, Purdue University, Lafayette, Ind., Tech. Rep. TR-EE 71-37, Aug. 1971.

[7] Keskes, N., Maitre, H., Kretz, F., "Statistical Study of Edges in TV pictures, " IEEE Trans. Vol.COM-27, No.8, p.1239-47, Aug. 1979.

[8] Gupta, J.N., Wintz, P.A., "A Boundary Finding Algorithm and its Applications," IEEE Trans. Circuits and Systems, Vol.22, No.4, p.351-362, April 1975.

[9] Jahanshahi, M.H., "Statistical Scene Analysis: Boundary Estimation of Objects in Presence of Noise," USCIPI Report 590, University of Southern Calif., Aug. 1975.

[10] Graham, D.N., "Image Transmission by Two-Dimensional Contour Coding," Proc. IEEE, Vol.55, No.3, p.336-346, March 1967.

[11] Schreiber, W.F., Huang, T.S., Tretiak, O.J., "Contour Coding of Images," Picture Bandwidth Compressions, Huang, T.S., Tretiak, O.J., ED., Gordon and Breach, New York, p.443-448, 1972.

[12] Collins, P.V., "Segmentation and Texture Analysis for very Low Bit-rate Image Coding," International Picture Coding Symposium, PCS 83, UC, Davis, p.111-112, March 28-30, 1983.

[13] Forchheimer, R., Fahlander, O., "Low Bit-rate Coding through Animation," International Picture Coding Symposium, PCS 83, UC, Davis, p.113-4, March 28-30, 1983.

[14] Jain, A.K., "Image Data Compression: A Review," Proc. IEEE, Vol.69, No.3, p.349-389, March 1981.

[15] Kummerow, T., "DPCM System with Two-dimensional Predictor and Controlled Quantizer," NTG-Tagung Signalverarbeitung Erlangen, p.425-439, 1973.

[16] Inose,H., Ishizaka, "Adaptive Coding usng Two-dimensional Prediction for Image Signals," Trans. Instit. of electron. Commun. Eng., Japan, p.527-534, 1974.

[17] Aughenbaugh, G.W., Irwin, J.D., O'Neal,J.B., "Delayed Differential Pulse Code Modulation,", in Porc. 2nd Annu. Princeton Conf., p.125-130, Oct.1970.

[18] Stuller, J.A., Kurz, B., "Intraframe Sequential Picture Coding," IEEE Trans. Vol.COM-25, No.5, p.485-489, May 1977.

[19] Berger, T., "Rate Distortion Theory: A Mathematical Basis for Data Compression," Prentice-Hall, Inc. Englewood Cliffs, New Jersey, 1971.

[20] Davision, L.D., "Rate Distortion Theory and Applications," Proc. IEEE, Vol.60, No.7, p.800-808, July 1972.

[21] Wintz, P.A., Kurtenbach, A.J., "Waveform Error Control in PCM Telemetry," IEEE Trans. Inf. Th., Vol.14, No.5, p.650-661, Sept.1968.

[22] Chen, W., Smith, C.H., "Adaptive Coding of Monochrome and Color Images," IEEE Trans. Commun., Vol.COM-25, No.11, p.1285-1292, Nov. 1977.

[23] Enomoto, H., Shibata, K., "Features of Hadamard Transformed Television Signals," 1965 Nat. Conf. IECE, Japan, Paper

[24] Habibi, A., Wintz, P.A., "Optimum Linear Transformation for Encoding Two-dimensional Data," presented at Symp. Picture Bandwidth Compression, M.I.T., Cambridge, MA, 1969.

[25] Landau, H.J., Slepian, D., "Some Computer Experiments in Picture Processing for Bandwidth Reduction," Bell System Tech. J., Vol.50, p.1525-1540, May-June 1971.

[26] Wintz, P.A., "Transform Picture Coding," Proc. IEEE, Vol.60, No.7, p.809-820, July 1972.

[27] Woods, J.W., Huang, T.S., "Picture Bandwidth Compression by Linear Transformation and Block Quantization,", Symp. Picture Bandwidth Compression, M.I.T., Cambridge, MA, 1969.

[28] Karhunen, H., "Uber Lineare Methoden In Der Wahrscheinlichkeitsrechnung," Ann. Acad. Sci. Fenn., Ser. A.I.37, Helsinki, Finland, 1947.

[29] Loeve, M., "Functions Aleatoires De Seconde Ordre," in P.Levy, Processus Stochastiques et Mouvement Brownien, Paris, France, Hermann, 1948.

[30] Hotelling, H., "Analysis of a Complex of Statistical Variables into Principal Components," J.Educ. Psychol., Vol.24, p.417-441, 1933.

[31] Jain, A.K., "A Fast Karhunen-Loeve Transform for a Class of Stochastic Processes," IEEE Trans. Commun., Vol.COM-24, p.1023-1029, Sept.1976.

[32] Jain, A.K., Wang, S.H., Liao, Y.Z., "Fast KL Transform Data Compression Studies," Proc. NTC 1976 Dallas, Texas, Nov. 1976.

[33] Tasto, N., Wintz, P.A., "Image Coding of Adaptive Block Quantization," IEEE Trans. Comm. Tech., Vo.COM-19, No.6, p.957-971, Dec. 1971.

[34] Andrews, H., "Computer Techniques in Image Processing," New York: Academic Press 1970.

[35] Enomoto, H., Shibata, K., "Orthogonal Transform Coding System for Television Signals," IEEE Trans. Electromagn. Comp., Vol.EMC-13, p.11-17, Aug. 1971.

[36] Pratt, W.K. Welch, L.R. Chen, W.H., "Slant Transform for Image Coding," Proc. Symp. Application of Walsh Functions, Mar. 1972.

[37] Pratt, W.K., Chen, W., Welch, L.R., "Slant Transform Image Coding," IEEE Trans. Commun., Vol.COM-22, No.8, p.1075-1093, Aug. 1974.

[38] Haralick, R.M., Shanmugam, K., "Comparative Study of a Discrete Linear Basis for Image Data Compression," IEEE Trans. on Systems, man and Cybernetics, Vol.SMC-4, No.1, p.16-27, Jan. 1974.

[39] Fino, B.J., Algazi, V.R., "Slant Haar Transform," Proc. IEEE, Vol.62, p.653-654, May 1974.

[40] Andrews, H.C., Patterson, C.L., "Singular Value Decomposition (SVD) Image Coding," IEEE Trans. on Communication, p.425-432, April 1976.

[41] Kekre, H.B., Solanki, J.K., "Modified Slant and Modified Slant Haar Transforms for Image Data Compression," Comput. and Electr. Eng.(USA), Vol.4, No.3, p.199-206, Sept. 1977.

[42] Ahmed, N., Natarajan, T., Rao, K.R., "Discrete Cosine Transform," IEEE Trans. Computers, p.90-93, Jan. 1974.

[43] Chen, W., Smith, C.H., Fralick, S.C., "A Fast Computational Algorithm for the DCT," IEEE Trans., Vol.COM-25, No.9, p.1004-1009, Sept. 1977.

[44] Shanmugam, K.S., "Comments on 'Discrete Cosine Transform'," IEEE Trans., Vol.C-24, No.7, p.759, July 1975.

[45] Yemini, Y., Pearl, J., "Asymptotic Properties of Discrete Unitary Transforms," IEEE Trans. Vol.PAMI-1, No.4, p.366-371, Oct. 79.

[46] Clarke, R.J., "Relation between the Karhunen-Loeve and Cosine Transforms," IEE Proc., Vol.128, No.6, p.359-360, Nov.1981.

[47] McWhirter, J.G., Roberts, J.B.G., Simons, R.F., "A compact digital coder for extreme bit rate reduction of television pictures," Proc. No.49, IERE Inter. Conf. on Digital Processing of signals in Communications, Loughborough, 7th-10th April 1981.

[48] Dillard, G.M., "Application of Ranking Techniques to Data Compression for Image Transmission," NTC 1975 Conf. Rec., Vol.1, p.22.18-22, 1975.

[49] Turkington, R.D., "Two Dimensional Transform Coding of Colour Television Pictures using the Discrete Cosine Transform," IEE Colloquium on 'Transform Techniques in Image Processing, Digest No.1983/50, Paper No.11, 18 May 1983.

[50] Anderson, G.B., Huang, T.S., "Piecewise Fourier Transformation for picture Bandwidth Compression," IEEE Trans. Commun. Tech., Vol.COM-19, p.133-140, April 1971.

[51] Claire, E.T., "Bandwidth Compression in Image Transmission," Proc. of ICC 1972, p.39.2-13, 1972.

[52] Gimlett, T.I., "Use of Activity Classes in Adaptive Transform Image Coding," IEEE Trans. Commun., Vol.23, p.785-6, July 1975.

[53] Tescher, A.G., Andrews, H.C., Habibi, A., "Adaptive Phase Coding in Two and Three-dimensional Fourier and Walsh Image Compression," Picture Coding Symp., Goslar, Germany, Aug. 1974.

[54] Cox, R.V., Tescher, A.G., "Generalized Adaptive Transform Coding," Picture Coding Symp., Asilomar, CA, Jan. 1976.

[55] Tescher, A.G., Cox, R.V., "An Adaptive Transform Coding Algorithm," Proc. of ICC 1976, p.47.20-25, 1976.

[56] Wong, W.C., Steele, R., "Adaptive Coding of Discrete Cosine Transform Video Telephone Pictures," Picture Coding Symp., Ipswich, England, July 1979.

[57] Wong, W.C., Steele, R., "Adaptive Discrete Cosine Transformation of Pictures using a logarithmic Energy Distribution

Model," Proc. No.49, IERE Inter. Conf. on Digital Processing of Signals in Commun. Loughborough, 7th-10th April 1981.

[58] Habibi, A., "Hybrid Coding of Pictorial Data," IEEE Trans. Commun., Vol.COM-22, No.5, p.614-24, May 1974.

[59] Netravali, A.N., Prasada, B., Mounts, F.W., "Some Experiments in Adaptive and Predictive Hadamard Transform Coding of Pictures," Bell System Tech. J., Vol.56, No.8, p.1531-47, Oct. 1977.

[60] Ishii, M., "Picture BW Compression by DPCM in Hadamard Transform Domain," Fujitsu Sci. and Tech. J., Vol.10., No.3, p.51--65, Sept. 1974.

[61] Means, R.W., Whitehouse, H.J., Speiser, J.M., "Television Encoding using a Hybrid Discrete Cosine Transform and Differential Pulse Code Modulator in Real-Time," Proc. Nat. Telecom. Conf., San Diego, p.61-66, Dec. 1974.

[62] Jones, H.W., "A Conditional Replenishment Hadamard Video Compressor," SPIE Vol.119, Applications of Digital Image Processings, p.91-98, 1977.

[63] Roese, J.A., Pratt, W.K., Robinson, G.S., "Interframe Cosine Transform Image Coding," IEEE Trans. Commun., Vol.COM-25, No.11, Nov. 1977.

[64] Farrelle, P.M., "Recursive Block Coding," IEE Colloquium on 'Transform Techniques in Image Processing', Digest No.1983/50, Paper No.11, 18 May 1983.

[65] Meiri, A.Z., Yudilevich, E., "A Pinned Sine Transform Image Coder," IEEE Trans. Vol.COM-29, No.12, p.1728-1735, Dec.1981.

[66] Netravali, A.N., Limb, J.O., "Picture Coding: A Review," Proc. IEEE, Vol.68, No.3, p.366-406, March 1980.

[67] Sakrison, D.J., "On the Role of the Observer and a Distortion Measure in Image Transmission," IEEE Trans. Vol.COM-25, No.11, p.1251-1267, Nov. 1977.

[68] Campbell, F.W., "The Human Eye as an Optical Filter," Proc. IEEE, Vol.56, No.6, p.1009-1014, June 1968.

[69] Bryngdahl, O., "Characteristics of the Visual System: Psychophysical Measurement of the Response to Spatial Sine-Wave Stimuli in the Mesopic Region," J. Opt. Soc. Am., Vol.54, No.9, p.1152-1160, Sept. 1964.

[70] Lowry, E.M., Depalma, J.J., "Sine Wave Response of the Visual System, I. The Mach Phenomenon," J. Opt. Soc. Am., Vol.51, No.7., p.740-746, July 1961.

[71] Lowry, E.M., Depalma, J.J., "Sine Wave Response of the Visual System, II. Sine Wave and Square Wave Contrast Sensitivity," J. Opt. Soc. Am., Vol.52, No.3, p.328-335, Mar. 1962.

[72] Sachs, M.B., Nachmias, J., Robson, J.G., "Spatial Frequency Channels in Human Vision," J. Opt. Soc. Am., Vol.61, No.9, p.1176-1186, Sept. 1971.

[73] Stockham, T.G. Jr., "Image Processing in the Context of a Visual Model," Proc. IEEE, Vol.60, No.7, p.828-842, July 1972.

[74] Pearson, D.E., "A Realistic Model for Visual Communication Systems," Proc. IEEE, Vol.55, No.3, p.380-389, March 1967.

[75] Mannos, J.L., Sakrison, D.J., "The Effects of a Visual Fidelity Criterion on the Encoding of Images," IEEE Trans. Inf. Theory, Vol.20, p.525-536, 1974.

[76] Hall, E.L., Computer Image Processing and Recognition, Academic Press, New York, 1979.

[77] Rademacher,H., "Einige Satze von allgemeinen Orthogonalfunktionen," Math. Annalen 87, p.122-138, 1922.

[78] Rader, C.M., McClellan, J.H., "Number Theory in Digital Signal Processing," Prentice Hall, 1979.

[79] Kargapolov, M.I., Merzljakov, Ju.I., "Fundamentals of the Theory of Groups," Springer-Verlag, New York, Heidelberg, Berlin, 1979.

[80] Nering, E., "Linear Algebra and Matrix Theory," Wiley International Edition, 1970.

[81] Ayres, F., "Theory and Problems of Matrices," Schaum's outline Series McGraw-Hill Book Company, 1962.

[82] Hollingsworth, C.A., "Vectors, Matrices, and Group Theory for Scientists and Engineers," McGraw-Hill Book Company, 1967.

[83] Mirsky, L., "An Introduction to Linear Algebra," Oxford University Press, 1963.

[84] Pratt, W.K., "Digital Image Processing," John Wiley and Son, 1978.

[85] Ray, W.D., Driver, R.M., "Further decomposition of the Karhunen-Loeve series representation of a stationary ramdom process," IEEE Trans. Inf. Theory, Vol.IT-16, p.663-668, Nov. 1970.

[86] Jain, A.K., "Some new techniques in image processing," ONR Symposium on Current Problems in Image Science, Naval Post Graduate School, Monterey, California, p.201-223, Nov. 10-12, 1976.

[87] Jain, A.K., "A sinusoidal family of unitary transforms," IEEE Trans. Vol.PAMI-1, No.4, p.356-365, Oct. 1979.

[88] Kennett, B.L.N., "A note on the finite Walsh transform," IEEE Trans. Inf. Theory, Vol.IT-16, p.489-491, July 1970.

[89] Robinson, G.S., " Logical convolution and discrete Walsh and Fourier power spectra," IEEE Trans. Audio Electroacoust. (Special Issue on Digital Filtering), Vol.AU-20, p.271-280, Oct. 1972.

[90] Gulamhusein, M.N., "Simple matrix-theory proof of the discrete dyadic convolution theorem," Electron. Letter (GB), p.238-239, 17 May 1973.

[91] Haar, A., "Zur Theorie Der Orthogonalen Funktionen-System," Inaugural dissertation, Math. Annalen, 5, p.17-31, 1955.

[92] Max, J., "Quantizing for Minimum Distortion," IRE trans. Inf. Theory 6, 7-12, p.169-176, 1960.

[93] Roe, G.M., "Quantizing for Minimum Distortion," IEEE Trans. Inf. Theory, Vol.IT-10, p.384-385, 1964.

[94] Algazi, V.R., "Useful Approximations to Optimum Quantization," IEEE Trans. Vol.COM-14, No.3, p.297-301, June 1966.

[95] Ghanbari, M, Pearson, D.E., "Hadamard Coefficients in Transformed TV Pictures, Probability Density Functions," Electron. Lett. (GB), Vol.14, No.8, p.252-254, 13 April 1978.

[96] Ngan, K.N., "Adaptive Transform Coding of Video Signals," IEE Proc. Vol.129, No.1, Feb. 1982. papes

[97] Jury, E.I., " Sampled-Data Systems Revisited: Reflections, Recollections, and Reassessments," The Newsletter of the IEEE ASSP Society, No.54, p.14-20, June 81

[98] Cooley, J.W., Lewis, P.A., Welch, P.D., "Historical Notes on the Fast Fourier Transform," Proc. IEEE, 55, 10, p.1675-7, Oct. 1967.

[99] Andrews, H.C., Pratt, W.K., "Fourier Transform coding of images," Hawaii International Conf. on System Sciences, p.677-679, Jan. 1968.

[100] Harmuth, H.F., "A Generalized Concept of Frequency and some Applications," IEEE Trans. on Information Theory, Vol.IT-14, No.3, p.375-382, May 1968.

[101] Pratt,W.K., Kane,J., Andrews,H.C., "Hadamard Transform Image Coding," Proc. IEEE, Vol.57, No.1, p.58-68, Jan.1969.

[102] Proc. Symp. Applications of Walsh Functions, Washington, D.C., Mar.31-Apr.3, 1970.

[103] Proc. Symp. Applications of Walsh Functions, Washington, D.C., Apr. 1971.

[104] Proc. Symp. Applications of Walsh Functions, Washington, D.C., Apr. 1972.

[105] Proc. Symp. Applications of Walsh Functions, Washingtion, D.C., Apr. 1973.

[106] Blachman, N.M., "Sinusoids versusWalsh Functions," Proc. IEEE, Vol.62, No.3, p.346-354, Mar. 1974.

[107] Henderson, K.W., "Some Notes on the Walsh Functions," IEEE Trans. Electronic Computers, Vol.EC-13, p.50-52, Feb. 1964.

[108] Paley, R.E.A.C., "A Remarkable Series of Orthogonal Functions," Proc. London Math. Soc., Vol.2, No.34, p.241-279, 1934.

[109] Ahmed, N., Rao, K.R., "Orthogonal Transforms for Digital Signal Processing,", Berlin, Germany: Springer-Verlag, 1975.

[110] Ahmed, N., Schreiber, H.H., Lopresti, P.V., "On Notation and Definition of Terms related to a Class of Complete Orthogonal Functions," IEEE Trans. Electromag. Comp., Vol.EMC-15, p.75-80, May 1973.

[111] Ahmed, N., Rao, K.R., Abdussattar, A.L., "BIFORE or Hadamard Transform," IEEE Trans. Audio Electroacoust., Vol.AU-19, p.225-234, Sept. 1971.

[112] Yuen, C.K., "Remarks on the Ordering of Walsh Functions," IEEE Trans. Comput., Vol.C-21, p.1452, Dec. 1972.

[113] Ohnsorg, F.R., "Spectral Modes of the Walsh-Hadamard Transform," IEEE Trans. Vol.EMC-13, p.55-59, Aug. 71.

[114] Gonzalez, R.C., Wintz, P., Digital Image Processing, Addison-Wesley Publishing Company, Inc., Advanced Book Program.

[115] Larsen, R.D., Madych, W.R., "Walsh-like expansions and Hadamard Matrices, Orthogonal System Generation," IEEE Trans. Acoust., Speech and Signal Process, Vol.ASSP-24, No.1, p.71-75, Feb. 1976.

[116] Paley, R.E.A.C., "On Orthogonal Matrices," J. Math. Phys., Vol.12, p.311-320, 1933.

[117] Hubner, H., "Multiplex Systems using Sums of Walsh Functions as Carriers," IEEE Trans., Vol.EMC-13, p.180-191, Aug. 1971.

[118] Luke, H.K., "Lineare Signalverknupfungen in der Multiplex technik," Arch. elektr. Ubertr. 24, p.57-65, 1970.

[119] Walsh, J.L., "A closed Set of Orthogonal Functions," American J. of Mathematics, Vol.45, p.5-24, 1923.

[120] Fine, N.J., " On the Walsh Functions," Trans. Amer. Math. Soc., Vol.65, p.372-414, 1949.

[121] Alexits, G., Convergence Problems of Orthogonal Series, New York: Pergamon, p.51-62, 1961.

[122] Corrington, M.S., "Advanced Analytical and Signal Processing Techniques," Doc. RADC-TDR-62-200, AD277942, Apr. 1962.

[123] Swick, D.A., "Walsh Function Generation," IEEE Trans. Information Theory, Vol.IT-15, p.167, Jan. 1969.

[124] Shanks, J.L., "Computation of the fast Walsh-Fourier Transform," IEEE Trans. Comput., Vol.C-18, p.457-459, May 1969.

[125] Whechel, J.E.,Jr., Guinn, D.F., "The fast Fourier-Hadamard transform and its use in signal representation and classification," EASCON'68, Rec., p.561-573.

[126] Manz, J.W., "A Sequency Ordered Fast Walsh Transform," IEEE Trans. Audio Electroacoust., Vol.AU-20, p.204-205, Aug. 1972.

[127] Fino, B.J., "Relations between Haar and Walsh/Hadamard Transforms," IEEE Proc., Vol.60, No.5, p.647-648, May 1972.

[128] Fontaine, A.B., "Simple Dyadic and Sequency Fortran Mechanizations of the Fast Walsh Transform," Proc. Nat. Electron. Conf., Vol.28, p.271-273.

[129] Larsen, H., "An Algorithm to compute the Sequency Ordered Walsh Transform," IEEE Trans., Vol.ASSP-24, No.4, p.335-6, Aug. 1976.

[130] Lackey, R.B., Meltzer, D., "A Simplified Definition of Walsh Functions," IEEE Trans. Computers, Vol.C-20., p.211-213, Feb. 1971.

[131] Carl, J.W., " Comments on 'A simplified Definition of Walsh Functions'," IEEE Trans. Comput., Vol.C-20, p.1617, Dec. 1971.

[132] Pearl, J., "Application of Walsh Transform to Statistical Analysis," IEEE Trans. System Man Cybern, Vol.SMC-1, p.111-119, Apr. 1971.

[133] Yuen, C.K., "Comment on 'Application of Walsh Transfrom to Statistical Analysis'," IEEE Trans. System Man Cybern., Vol.SMC-2, p.294, Apr.1972.

[134] Fino, B.J., Algazi, V.R., "Unified Matrix Treatment of the Fast Walsh-Hadamard Transform," IEEE Trans. on Computer, Vol.C-25, p.1142-1146, Nov.1976.

[135] Wong, W.C., "Adaptive Transform Coding of Viewphone Signals," Ph.D. Thesis, Department of Electronic and Electrical Engineering, Loughborough University, Feb. 1980.

[136] Mitchell, O.R., Tabatabai, A.J., "Channel Error Recovery for Transform Image Coding," IEEE Trans. on Commun., Vol.COM-29, No.12, p.1754-1762, Dec. 1981.

[137] Geadah, Y.A., Corinthios, M.J.G., "Walsh Hadamard transform, natural, dyadic and sequency order algorithms," IEEE Trans. on Comput., Vol.C-26, No.5, p.415-442, May 1977.

Appendix A

Prove: Given vectors D_{pq} and matrices $[R]_{pq}$, show that the scalar

$$e = \sum_{p=1}^{P} \sum_{q=1}^{Q} \left| D_{pq} + [R]_{pq} \times A \right|^{2} \quad \text{------------------(A.1)}$$

is minimum when

$$A = - [RR]^{-1} \times C \quad \text{-----------------------------------(A.2)}$$

where

$$[RR] = \sum_{p=1}^{P} \sum_{q=1}^{Q} [R]^{t}_{pq} \times [R]_{pq} \quad \text{----------------------(A.3)}$$

$$C = \sum_{p=1}^{P} \sum_{q=1}^{Q} [R]^{t}_{pq} \times D_{pq} \quad \text{----------------------(A.4)}$$

Proof:

$$e = \sum_{p=1}^{P} \sum_{q=1}^{Q} \left| D_{pq} + [R]_{pq} \times A \right|^{2}$$

$$= \sum_{p=1}^{P} \sum_{q=1}^{Q} \{ D_{pq} + [R]_{pq} \times A \}^{t} \{ D_{pq} + [R]_{pq} \times A \}$$

$$= \sum_{p=1}^{P} \sum_{q=1}^{Q} \{ D^{t}_{pq} + A^{t} \times [R]^{t}_{pq} \} \{ D_{pq} + [R]_{pq} \times A \}$$

$$= \sum_{p=1}^{P} \sum_{q=1}^{Q} \{ D^{t}_{pq} D_{pq} + D^{t}_{pq} [R]_{pq} A + A^{t} [R]^{t}_{pq} D_{pq}$$

$$+ A^{t} [R]^{t}_{pq} [R]_{pq} A \}$$

$$= \sum_{p=1}^{P} \sum_{q=1}^{Q} \{ D^t_{pq} D_{pq} + 2(D^t_{pq} [R]_{pq} A) + A^t ([R]^t_{pq} [R]_{pq}) A \}$$

$$\text{---------------------------(A.5)}$$

First we represent equation A.5 in a simple form as

$$= b + 2 C^t A + A^t [RR] A \qquad \text{---------------------(A.6)}$$

$$\text{with} \quad b = \sum_{p=1}^{P} \sum_{q=1}^{Q} D^t_{pq} D_{pq} \qquad \text{-----------------------------(A.7)}$$

$$[RR] = \sum_{p=1}^{P} \sum_{q=1}^{Q} [R]^t_{pq} [R]_{pq} \qquad \text{----------------------------(A.8)}$$

$$C = \sum_{p=1}^{P} \sum_{q=1}^{Q} D^t_{pq} [R]_{pq} \qquad \text{----------------------------(A.9)}$$

Before we proceed with equation A.6, we first derive (from equations A.10 to A.13) an equation which will be used to represent the last two terms of equation A.6 in another form.

$$k = ([RR] A + C)^t [RR]^{-1} ([RR] A + C) \qquad \text{-----------(A.10)}$$

$$= (A^t [RR]^t + C^t) [RR]^{-1} ([RR] A + C)$$

$$= A^t [RR]^t [RR]^{-1} [RR] A + A^t [RR]^t [RR]^{-1} C$$

$$+ C^t [RR]^{-1} [RR] A + C^t [RR]^{-1} C \quad (A.11)$$

$$= A^t [RR] A + A^t C + C^t A + C^t [RR]^{-1} C \qquad \text{----------(A.12)}$$

The conversion between A.11 and A.12 is possible because [RR] as given by equation A.9 is a symmetrical matrix, i.e.: $[RR]^t = [RR]$. Therefore, we have

$$k \quad = \quad A^{t}[RR]\,A \;+\; 2\,C^{t}A \;+\; C^{t}[RR]^{-1}C \quad \text{-----------------}(\ A.13\ )$$

Now, we can proceed with equation A.6 which, by means of equation A.13, is converted into A.14:

$$e \quad = \quad b \;+\; \{\ (\ [RR]A + C\ )^{t}\,[RR]^{-1}\,(\ [RR]A + C\ ) \;-\; C^{t}[RR]^{-1}C$$

$$= \quad b \;+\; k \;-\; C^{t}[RR]^{-1}C \qquad \text{------------------}(\ A.14\ )$$

As defined by equation A.9, the matrix [RR] is a positive definite or semi-positive quadratic form and so the value of k is always positive (Similarly, $C^{t}[RR]^{-1}C$ and b are always positive). Thus, e is minimum when k is equal to zero, i.e.:

$$[RR]\,A \;+\; C \;=\; 0$$

or $$A \;=\; -[RR]^{-1}\,C$$

(Q.E.D)

Appendix B

Prove : Given vectors D_p and V, show that the scalar

$$e = \sum_{p=1}^{2} \left| D_p - a \times V \right|^2 \qquad \text{--------------------(5.9)}$$

is minimum when

$$a = (1/2) \times \sum_{p=1}^{2} \sum_{q=1}^{n} d(p,j) \qquad \text{-----------------(5.10)}$$

where $d(p,j)$ is the jth element of the vector D_p and
$V^t = (1/n, 1/n, \ldots, 1/n)^t$

Proof:

Equation 5.9 is a special case of equation A.1 with

$D_{pq} = D_p$, $[R]_{pq} = V$ and $A = a$.

By means of equation A.2 to A.4, we have e is minimum when

$$a = -[RR]^{-1} C \qquad \text{---------------------------------------(B.1)}$$

where

$$[RR] = \sum_{p=1}^{2} V^t V = 2/n \qquad \text{------------------------------(B.2)}$$

$$C = \sum_{p=1}^{2} V^t D_p = (1/n) \sum_{p=1}^{2} \sum_{j=1}^{n} d(p,j) \qquad \text{------------(B.3)}$$

Equations B.1 to B.3 imply

$$a = (1/2) \sum_{p=1}^{2} \sum_{j=1}^{n} d(p,j)$$

(Q.E.D)