Orthogonal Transforms and their Application to Image Coding

by

NASSER MOHAMMADI NASRABADI

June 1984

A Thesis submitted for the degree of
Doctor of Philosophy in the Faculty of Engineering
and the Diploma of Imperial College (D.I.C.)

Department of Electrical Engineering
Imperial College of Science & Technology
(University of London)
London

ORTHOGONAL TRANSFORMS AND THEIR APPLICATIONS TO IMAGE CODING

ABSTRACT

The properties of orthogonal transforms and the problems of the image coding techniques with the applications of linear transforms have been mentioned.

The linear transforms that have been studied are the Discrete Fourier transform, Number theoretic transforms, P-adic transforms, Cosine transform, Hadamard transform and Polynomial transform. Number theoretic transforms have been discussed because they have the ability to transform a sequence of data without introducing any arithmetic errors. However, these transforms are defined modulas an integer and so the transfomm length as well as the dynamic range is limited. Thus a new transform has been introduced in the P-adic field, which will give a larger dynamic range. Error free arithmetic can be performed in the P-adic field even with rational numbers.

Extension fields of P-adic numbers have been discussed and new transforms in these fields have been developed. Only complex P-adic fields and quadratic fields have been investigated. Hadamard and Cosine transforms and their applications to digital image coding have been studied. Polynomial transform algorithms have also been mentioned, since these algorithms can implement linear transforms very efficiently.

Digital image coding algorithms such as intraframe, interframe, predictive and hybrid techniques have been studied and simulated for several digital pictures. A two-dimensional block transform coding system has been simulated and applied to a test image to reduce the bit rate. Adaptive intraframe and a predictive coding system have been discussed and some

new techniques have been developed.

Hybrid transform/DPCM coding techniques have been studied since they are computationally more efficient than the two-dimensional block transform coders. A new hybrid coder has also been developed which consists of a transformation and a vector quantization scheme. The vector quantizer scheme is a look-up table containing the most probable vector patterns of the transformed coefficients, which are identified by a codeword that is transmitted or stored. This new hybrid coder has been simulated for digital pictures, and results have been compared with that of conventional hybrid techniques.

Finally, interframe coding techniques have been discussed. Most of the available techniques are reviewed and compared. A new three-dimensional hybrid technique has been developed which consists of a two-dimensional linear transform and a vector quantization scheme. Simulation has been performed on an image sequence for comparison with other coding techniques. Objects within digital image sequences move at different speeds so an adaptive hybrid transform/vector quantization technique has been developed. This uses several codebooks of different dimensions in order to exploit the motion variations within the image sequence. The adaptive hybrid coding technique has been simulated for a digital image sequence and the results are compared with the non-adaptive hybrid technique.

ACKNOWLEDGEMENTS

I am very pleased to acknowledge my indebtedness to my supervisor, Dr. R.A. King for his valuable assistance and guidance during my stay at Imperial College. His encouragement and very valued criticisms are most sincerely appreciated.

I would like to express my gratitude for all the help and advice received from my colleagues and friends at Imperial College.

I would also like to thank Miss C. Collins for skillfully typing and correcting the manuscript.

Finally, it gives me special pleasure to express my deep love to my parents for all their help and support during my study.

LIST OF SYMBOLS AND ABBREVIATIONS

The following system of numbering and cross references is used in this thesis. Each chapter is labelled with a numeral and sub-divided into sections. All sections, figures, tables and equations within a chapter are numbered consecutively starting from 1. At the end of the volume there is a list of references. When such a reference is made in the thesis, it is denoted by its list number in brackets.

The following is a list of symbols and abbreviations appearing in the thesis.

M	Integer
P	Prime integer
F_t	Fermat
I_p	Field of integers with P an integer prime
Z_M	Ring of integers with M an integer
(mod M)	Modulo M
G	The group G
$\equiv$	Equivalence sign
a|b	a is a divisor of b
axb	a is not a divisor of b
$<a>_b$	Residue of a modulo b
o	Convolution operation
NxN	Denotes a matrix with N rows and N columns
$\otimes$	Element by element multiplications (for matrices)
$GF(p^2)$	Galios field with p a prime integer
$GF(M^2)$	Galios ring with M an integer
Q_p	Field of p-adic numbers

$\hat{Q}_p$	Segmented p-adic field
H(p.r,α)	Hensel code representation of a number α
Q_g	g-adic ring
$Q_p(\sqrt{-1})$	Complex fields with p a prime integer
$Q_p(\sqrt{m})$	Quadratic fields with p a prime integer
$Q_p(\sqrt{m}) = K_p$	Quadratic extension field of Q_p
$\hat{Q}_p(\sqrt{m}) = \hat{K}_p$	Segmented quadratic extension field of Q_p
φ or Φ	Euler's totient function
NTT	Number Theoretic Transform
MNT	Mersenne Number Transform
FNT	Fermat Number Transform
CNTT	Complex Number Theoretic Transform
PT	Polynomial Transform
HT	Hadamard Transform
CT	Cosine Transform
DFT	Discrete Fourier Transform
IDFT	Inverse Discrete Fourier Transform
FFT	Fast Fourier Transform
KVT	
PAT	P-Adic Transform
CPAT	Complex P-Adic Transform
WFT	Winograd Fourier Transform
IIR	Infinite Impulse Response
FIR	Finite Impulse Response
P(Z)	Polynomial in terms of Z

$P(z)$	Polynomial in terms of z
$\frac{R(z)}{P(z)}$	Polynomial ring
(Mod P(z))	Modulo polynomial P(z)
CRT	Chinese remainder theorem
$(\frac{a}{p})$	Lagendre symbol
$<x_n>$	P-adic sequence
$<\hat{x}_n>$	Canonical P-adic sequence
f(x)	A function with variable x
f´(x)	Derivative of f(x)
CCP	Circular convolution property

AUTHOR'S PUBLICATIONS

1. Nasrabadi, N.M., King, R.A., "Computationally efficient discrete cosine transform algorithms", Electron. Letts., 6 January 1983, vol. 19, no. 1, pp.24.

2. Nasrabadi, N.M., King, R.A., "Entropy-coded hybrid differential pulse-code modulation", Electron. Letts., 20th January 1983, vol. 19, no. 2, pp. 83.

3. Nasrabadi, N.M., King, R.A., "Transform coding using vector quantization", 1983 Conf. on Information Science and System, 23-25 March 1983. The John Hopkins University, Baltimore, Maryland, U.S.A.

4. Nasrabadi, N.M., King, R.A., "Image coding using vector quantization in the transform domain", BPRA Second Int. Conf. on Pattern Recognition and Pattern Recognition Letters, 19-21 September 1983, University of Oxford, England.

5. Nasrabadi, N.M., King, R.A., "Computationally efficient adaptive block-transform coding", Proc. of EUSIPCO-83, 2nd European Conf. on Signal Processing, 12-13 September, 1983.

6. Nasrabadi, N.M., King, R.A., "Fast Digital Convolution using P-ADIC transforms", Electron. Letts., 31 March 1983, vol. 19, no. 7, pp. 266.

7. Nasrabadi, N.M., King, R.A., "Interframe coding of moving image sequences", Pattern Recognition in Photogrammetry, 27-29 September, 1983, Gnaz, Austria.

8. Nasrabadi, N.M., King, R.A., "Complex number theoretic transform in P-Adic field", 1984 IEEE Int. Conf. on Acoustics, Speech and Signal Processing, 19-21 March, 1984, San Diego, California.

9. Nasrabadi, N.M., King, R.A., "A new image coding technique using transform vector quantization", 1984 IEEE Int. Conf. on Acoustic, Speech and Signal Processing, 19-21 March, 1984, San Diego, California.

10. Ibikunle, J.O., Nasrabadi, N.M., King, R.A., "Design of codec for video conferencing", First GRETSI-CESTA Image Symposium, Biarritz, France, 21-25 May, 1984.

11. Nasrabadi, N.M., King, R.A., "Design of a new codec system for video conferencing",Int. Conf. on Digital Signal Processing, Florence, Italy, 5-8 September, 1984.

12. Henein, K.M., Nasrabadi, N.M., Gorgui-Naguib, N.R., King, R.A., "Efficient technique for implementing fermat number transforms on microprocessors", 1984 IEEE Int. Symposium on Circuits and Systems, 7-10 May, 1984, Montreal, Canada.

CONTENTS

TABLE OF CONTENTS

Page

CHAPTER ONE

CHAPTER ONE

INTRODUCTION

Orthogonal transforms have been applied in many applications such as in digital image processing. One of the major applications of orthogonal transforms is in digital medical image processing. The introduction of computerised tomography (CT) scanners [A1] was a major advance in digital image processing whereby three-dimensional images were reconstructed from their two-dimensional projections. CT scanners of head and body have resulted in useful techniques for detection of tumours and infectious diseases in medical diagnosis. Medical images are usually reconstructed from X-ray [A2], single photo [A3] or ultrasound computerised tomography [A4] projections by using Radon transform [A5]. The Radon transform or parallel projection of a two-dimensional object $f(x,y)$ is given by

$$P_\theta(t) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\ \delta(x\cos\theta + y\sin\theta - t)\,dxdy \quad (1.1)$$

where δ represents the line in xy plane.

It has been shown that the Fourier transform of $f(x,y)$ in polar coordinates $F(w,\theta)$ is equal to the one-dimensional Fourier transform of the projections $S_\theta(w)$. Thus

$$F(w,\theta) = S_\theta(w)$$

where (w,θ) are polar coordinates.

In order to reconstruct images some two-dimensional Fourier transforms have to be implemented, so fast transform algorithms are welcomed. Other applications of reconstruction from projections include radioastronomy, optical interferometry, electron microscopy and geophysical exploration.

Another application of image processing is in Remote sensing, where a three-dimensional transform is performed on a multi-spectral band image to investigate inter-band correlation and image information.

Multi-spectral images are obtained from the Landsat satellites. The first Landsat-1 was launched in 1972 [A6], data acquired by these Multi-spectral Scanners (MSS) have been utilised for a number of applications. The Scanners on the first Landsat series had only four spectral bands (0.5-1.1 μm) with pixel resolution of 75 m square. The new MSS on Landsat-D, launched in 1982, has seven spectral bands (0.45-12.5 μm) with pixel resolution of 30 m. The images formed by these MSS's produce a huge quantity of data which are either transmitted to a communication satellite or stored on tapes. The applications of these images are in agriculture, topographic maps, crop types, infestation, soil texture, forest texture, flood control and many more [A7].

Digital image processing has been used for classification and identification of crops. Image processing techniques, such as enhancement, restoration and coding, have been applied to these images. Transform coding techniques have been investigated for reduction of data obtained from Landsat Scanners [A7]. Fast two- or three-dimensional transforms have to be performed for coding algorithms.

Other major applications of image processing are in robot vision, for tasks such as inspection of industrial parts [A8]. A large number of applications have been investigated, such as visual inspection of integrated circuits, recognition of agricultural objects, locating surface defects in wood, visual inspection system for hot steel slabs, and

many more applications [A9], [A10].

Digital recursive and non-recursive filters have been designed for many years for filtering. Noise removal, low-pass and high-pass filtering have been applied to digital images for many years;these are implemented by Fourier transform. In digital image enhancement digital convolution has to be implemented very efficiently. For example, a one-dimensional convolution can be represented by

$$y(\ell) = \sum_{n=0}^{N-1} x(n)\, h(\ell - \imath) \qquad \text{for} \quad \ell = 0,1,\ldots N-1 \tag{1.2}$$

where x(n) is the input sequence of N-points and h(n) is the impulse response of the filter. To implement expression (1.2) directly N^2 multiplications are required. However if the two sequences are transformed by a linear transform which has digital convolution properties,the convolution becomes N-point multiplications as given by

$$Y(K) = X(K) \cdot H(K) \tag{1.3}$$

where Y(K), X(K) and H(K) are the transformed data. Discrete Fourier transform (DFT) and Number theoretic transform (NTT) are the two examples of Orthogonal transforms which can implement expression (1.2) very efficiently. The bulk of the computation is the amount of arithmetic required to convert the input sequences into the transform domain and back. Thus a large number of algorithms [A11] has been developed to implement these transforms efficiently. Because of immense applications of expression (1.2) it has motivated us to investigate its implementation in detail by making investigations into some orthogonal transforms.

The first four chapters of this thesis are devoted to some orthogonal transforms.

NTT [A12] is believed to implement expression (1.2) very efficiently with no arithmetic errors. In Chapter two NTT's are defined in detail. In Section 2.2 NTT's in the field or ring of integers are introduced. Since arithmetic is performed modulo an integer no error is introduced in the computation of the transform. However the input sequence has to be scaled and rounded to an integer sequence. Thus a large dynamic range is required. In Section 2.4 we introduce complex NTT's for implementation of complex convolution directly instead of performing several real transforms. Digital complex convolutions are usually used to form Radar-images using Doppler effect [A13]. Fast implementation techiques and hardware implementations of NTT are not described since Chapter two is believed to be a short review. Appendix A is included which introduces Elementary Number Theory to complement Chapter two.

Since NTT's have to be scaled and rounded to integer value, this has motivated us to investigate other fields such as P-adic fields where rational numbers can be represented exactly. In Chapter three P-adic transforms are introduced. Appendix B is included to give introduction to P-adic numbers and some of the properties of the P-adic field. In Section 3.2 segmented P-adic field is introduced. In Section 3.3 the general P-adic transform is introduced which is proved to be orthogonal with convolution properties. In the sub-sections the derivation of the root of unity is discussed and,for practical purposes, Mersennes and Fermat P-adic transforms are developed. No simulation results are given, only

theoretical definitions of the P-adic transform are given. Implementation techniques for P-adic transform are discussed and comparisons with NTT are studied.

In Chapter four complex P-adic transforms are introduced. In Appendix B complex P-adic numbers are explained. In Section 4.2.1 complex P-adic convolution is introduced and its decomposition into real P-adic convolution is discussed. In Section 4.2.2 implementation of complex P-adic field via Fermat number P-adic transforms are discussed. In Section 4.3 P-adic transforms in extension fields are discussed. It is shown that complex P-adic transforms exist by choosing primes such as prime Mersenne integers. Finally, in Section 4.6, g-adic transforms are introduced which could be defined as the direct sum of several P-adic fields.

In Chapter five several orthogonal transforms are introduced. Discrete Fourier, Hadamard and Cosine transforms are discussed. In Section 5.4, implementation of discrete Fourier transform is explained. Makhoul's algorithm is introduced and compared with a new algorithm. This new algorithm employs a polynomial transform algorithm discussed in Section 5.5. In Section 5.5 the concept of polynomial transform algorithm is discussed, several polynomial transform algorithms are also explained.

One major application of Orthogonal transforms is in digital image coding [A14]. The high inter-pixel correlation in digital images is representative of a large redundancy in digital images which can be removed by two-dimensional transforms. Orthogonal transforms have the property that they decorrelate image-pixels and concentrate the energy around the low frequency

coefficients. This energy compaction property makes linear transforms useful for image coding. The discrete Cosine transform is believed to have the best energy compaction compared with other linear transforms. Transmission of digital TV images in digital Cable TV network is one application of digital image coding where transmission bandwidth is reduced by coding algorithms. Due to diverse applications several coding algorithms have been developed. In Chapter six image coding is introduced. In Section 6.3 Transform image coding concept is explained. In Section 6.3.1 Block transform coding is defined and in sub-sections adaptive block transform coding is explained. In Section 6.3.2 a new adaptive block transform coding is designed where inter-block correlation is investigated. Chapter six is devoted to intra-frame transform coding, transmission bit rate of R = 1.0 bits per pixel are shown to be achieved.

In Chapter seven predictive coding techniques are investigated. These coding techniques are believed to be very simple to implement. However, the transmission bit rates achieved are not as low as that of block transform coding. Entropy coders can be employed with predictive coders (DPCM) to reduce the bit rates. In Section 7.3 a new predictive coder is designed that modifies the histogram of the difference signal for its application to entropy coder.

Chapter eight is devoted to Hybrid image coders. Since DPCM systems have superior coding performances at high bit rates with less complex hardware implementation, and greatly reduced storage requirements compared to transform coders, they can be combined with transform coders, which produce higher compaction of energy and in turn lower bit rates,to introduce

hybrid transform/DPCM systems. The computational complexity of these hybrid systems is reduced compared with two-dimensional transform coders. In Section 8.2 and 8.3 one- and two-dimensional hybrid coders are introduced respectively. DPCM coders suffer from their lack of immunity to noise, and their compression ability is not as good as transform coders. Thus, in Section 8.4, a new hybrid coding technique is introduced, where transform and vector-quantizer coders are employed. Each line of the image data is transformed and a vector quantization scheme is applied in column direction. The vector-quantizer is more efficient than DPCM because the noise due to transmission is localised and does not produce the line effects as in DPCM. The vector-quantizer is a look-up table so it is computationally very fast and efficient. In Section 8.5 a two-dimensional hybrid coder is introduced. The concepts of these new hybrid coders are very similar to the conventional hybrid coders. The new hybrid coders are believed to be more efficient than conventional hybrid coders. No adaptivity was used in the hybrid coder, although adaptivity could be introduced. The two-dimensional hybrid coder is used to exploit the inter-block correlation.

In Chapter nine digital image sequence is introduced. Three-dimensional or inter-frame coders are studied. The applications where image coding is desirable are mentioned in Section 9.2. Human visual system is investigated in section 9.3. It is seen that the visual system behaves like a three-dimensional bandpass filter. In Section 9.4 several inter-frame coders are reviewed and the advantages and disadvantages are considered. In sub-section 9.4.3 movement compensated predictive coder is introduced. This coder has recently

been developed where the adaptivity is based upon the motion estimation. In sub-section 9.4.6 three-dimensional transform coders are reviewed and in Section 9.4.7 several inter-frame coders are referenced. A comparison between the coders is also given. In Section 9.5 a new inter-frame hybrid coder is designed. The hybrid coder involves a two-dimensional transform and a vector-quantiser coder. In sub-section 9.5.2 the adaptive version of the hybrid coder is developed, the adaptivity is based upon the motion variation in the image sequence. The image coding algorithm is based upon three-dimensional blocks of the image sequence. Thus several processors in parallel can be used to encode each three-dimensional blocks in parallel. The only difficulty with the proposed coder is the requirement for a large amount of memory.

The highly adaptive version of the proposed coder is believed to be as good as three-dimensional transform coders with less computational complexity.

CHAPTER TWO

CHAPTER TWO

NUMBER THEORETIC TRANSFORM

2.1 INTRODUCTION

Recently new transforms defined over a finite ring of integers with arithmetic carried out modulo an integer have been introduced. These transforms are believed to implement circular convolution very efficiently since only integer arithmetic is to be performed. In this chapter number theoretic transform (NTT) is reviewed. The conditions for existence of NTT over integer fields and rings are discussed in section 2.2. In section 2.3 NTT's with Mersenne and Fermat primes are discussed, together with their computational simplicity.

Complex number theoretic transforms are discussed in section 2.4. Transform over quadratic extension fields are discussed in section 2.5. Galois ringsare discussed in section 2.6. Introduction to elementary number theory is given in Appendix A, since this chapter is planned to be a short review.

2.2 NUMBER THEORETIC TRANSFORM [15], [16]

(a) Definition

The forward and inverse number theoretic transform (NTT) of an n-point sequence x(n) is defined respectively as:

$$X(K) = \sum_{n=0}^{N-1} x(n)\ \alpha^{nK} \qquad \text{modulo M} \qquad (2.1)$$

$$x(n) = N^{-1} \sum_{K=0}^{N-1} X(K)\ \alpha^{-nK} \qquad \text{modulo M} \qquad (2.2)$$

where X(K) is the transformed sequence, M an integer and $N.N^{-1} = 1$ modulo M. Number theoretic transform is said to be defined over a ring or a field of integers if arithmetic is performed modulo a rational integer or a prime integer respectively.

(b) Number Theoretic Transform over a Field, I_M

If transform is defined over a prime integer, the conditions for existence and having a circular convolution property are given by:

$$\alpha^N \equiv 1 \qquad \text{Mod M} \qquad (2.3)$$

$$N.N^{-1} \equiv 1 \qquad \text{Mod M} \qquad (2.4)$$

where α is a root of unity of order N, and M is a prime integer. The parameters α, N and M are all inter-related by expression (2.3). The maximum transform length is given by Euler's totient function $\varphi(M)$ defined as the number of integers smaller than M and relatively prime to it. Thus:

$$\varphi(M) = M - 1$$

In order for the transform to exist it must satisfy:

$$\alpha_\varphi^{M-1} \equiv 1 \qquad \text{Mod M} \qquad (2.5)$$

where α_φ is known as primitive root giving the maximum transform length. The α_φ is not the only primitive root, there are $\varphi(\varphi(M))$other primitive roots by Euler's functions[41]. From the above it is seen that an integer α_φ can be found in the field of a prime integer.

By Fermat-Euler theorem (Appendix A) roots of unity of order N where N must divide $\varphi(M)$ can be obtained from the expression

$$\alpha = \alpha_{\varphi}^{(M-1)/N} \tag{2.6}$$

Thus eqn. (2.3) can be solved for transform length N where N is divisible by M-1 denoted as $N|M-1$.

(c) Number Theoretic Transform over a Ring of Integers Z_M

If the transform pair in (2.1) and (2.2) is defined over a rational integer ring Z_M, the ring is decomposed into several fields and the condition for existence of transform is derived over each field.

A ring Z_M can be decomposed uniquely into several fields as defined by:

$$Z_M = I_{P_1} \otimes I_{P_2} \otimes \quad \ldots \quad I_{P_n} \tag{2.7}$$

where

$$M = P_1 P_2 \ldots P_{\ell} \tag{2.8}$$

The condition for existence with α as roots of unity is given by :

$$\begin{aligned} &(1)\quad \alpha^N \equiv 1 \quad \text{Mod } P_i \\ &(2)\quad NN^{-1} \equiv 1 \quad \text{Mod } P_i \qquad \text{for } i=1,2,\ldots \ell \\ &(3)\quad N|(P_i-1) \end{aligned} \tag{2.9}$$

If O(M) is defined as the greatest common divisor (gcd) of the (P_i-1) for $i=1,2 \ldots \ell$, then the maximum transform length in Z_M is

$$N_{max} = O(M)$$

The advantage of defining NTT's over a composite modulo is that a larger dynamic range is obtained and by using Chinese Remainder Theorem the NTT can be decomposed and implemented modulo each prime P_i in parallel and combined together finally by the Chinese Remainder Theorem. This has the advantage where arithmetic is evaluated modulo small primes P_i's in parallel using small word length processors.

The NTT in Z_M is given by

$$(X_1(K),\ X_2(K),\ \ldots X_\ell(K)) = \sum_{n=0}^{N-1} (x_1(n), x_2(n),\ \ldots\ x_\ell(n))(\alpha_1, \alpha_2, \ldots \alpha_\ell)^{nK} \tag{2.10}$$

It can also be written as

$$(X_1(K),\ X_2(K), \ldots X_\ell(K)) = \left(\sum_{n=0}^{N-1} x_1(n)\alpha_1^{nK},\ \sum_{n=0}^{N-1} x_2\alpha_2^{nK}, \ldots \sum_{n=0}^{N-1} x_\ell\ \alpha_\ell^{nK}\right) \tag{2.11}$$

where each sumation is evaluated modulo corresponding prime P_i. The data are represented by their decomposed form as

$$\begin{aligned} X(K) &= (X_1(K),\ X_2(K),\ X_i(K), \ldots\ X_\ell(K)) \\ x(n) &= (x_1(n),\ x_2(n),\ x_i(n), \ldots\ x_\ell(n)) \\ \alpha &= (\alpha_1,\ \alpha_2,\ \alpha_i, \ldots\ \alpha_\ell) \\ M &= (P_1,\ P_2,\ P_i \ldots\ P_\ell) \end{aligned} \tag{2.12}$$

where

$$X_i(K),\ x_i(n),\ \alpha_i\ \varepsilon\ I_{P_i}$$

2.3 MERSENNE AND FERMAT NUMBER TRANSFORMS [17], [18]

In previous sections NTT was defined over a field and ring. Here we consider hardware implementation of NTT's. Since arithmetic is performed modulo an integer prime, certain prime integers can be implemented very efficiently. Prime

integers of power of 2 can perform modulo arithmetic by only shifting and addition. The attractive primes are Mersenne primes defined as $M = 2^P-1$ where $P = 1,3,5,7,13,17,...61$, and Fermat primes defined as $M = 2^{2^t} + 1$ for $t = 1,2,3,4$. In the following sub-section transforms in each of these primes are considered.

(a) Mersenne Transform

Consider arithmetic modulo primes of form $M = 2^P-1$; then transforms in field I_M have maximum transform length given by $N_{max} = \varphi(M) = 2^P-2$, and primitive root of order N_{max} can be found. However, transforms of length P are found to have roots of $\alpha = 2$. These transforms are known as Mersenne Transforms defined as:

$$X(K) \equiv \sum_{n=0}^{P-1} x(n)\, 2^{nK} \quad \text{Mod}(2^P-1) \qquad (2.13)$$

$$\text{for } K = 1,\ldots P-1$$

These transforms have the disadvantage of being short length since P determines the processor word-length. Thus NTT's with large transform lengths require large word-length processors. Also transform length is not highly composite so simple FFT algorithm cannot be used. However, they can be implemented by more complex algorithms such as Winograd's and prime factor algorithms [19],[20]. These transforms can be used to implement circular convolutions of P-point. The inverse transform is defined as:

$$x(n) \equiv P^{-1} \sum_{K=0}^{P-1} X(K)\, 2^{-nK} \quad \text{Mod}(2^P-1) \qquad (2.14)$$

where

$$2^{-nK} \equiv 2^{P-nK} \quad \text{Mod}(2^P-1)$$

$$\text{for } n = 1,\ldots P-1$$

Mersenne transforms can also be defined over a composite modulo which can be factorised into several Mersenne prime factors. Mersenne transforms with $\alpha = 2$ can be implemented by $P(P-1)$ additions and $(P-1)^2$ shifts.

(b) Fermat Transform

If the prime integer is chosen as $F_t = 2^{2^t} + 1$ with $t = 1,2,3,4$, the maximum transform length is given by $N_{max} = 2^{2^t}$ which is highly composite and can be implemented by Radix-2 FFT algorithm. The primitive root is given by $\alpha_\varphi = 3$ which cannot be implemented by just shifting. However, transform of length $N = 2^t$ has roots of $\alpha = 2$ which can be implemented by shifting and Radix-2 FFT algorithm. These transforms are known as Fermat Transforms defined as:

$$X(K) = \sum_{n=0}^{N-1} x(n) \; 2^{nk} \qquad (\text{Mod } F_t) \qquad (2.15)$$

having an inverse of form:

$$x(n) = N^{-1} \sum_{K=0}^{N-1} X(K) \; 2^{-nK} \qquad (\text{Mod } F_t) \qquad (2.16)$$

where $N = 2^t$ which divides $\varphi(F_t)$.

Fermat transforms can implement circular convolution by only addition and shifting. The disadvantage of Fermat transform, as in Mersenne transform, is that the transform length is limited by word-length of the processor. Thus large data-sequences cannot be transformed directly, however it is possible to convert the sequence into a multi-dimensional sequence and then Fermat transform [A20]. Hardware implementation of Mersenne and Fermat transforms are considered by several researchers [A16], [A19].

2.4 COMPLEX NUMBER THEORETIC TRANSFORM, COMPLEX FIELD, RING, QUADRATIC FIELDS [A21], [A22]

In some applications like Synthetic aperature radar image formation, complex convolution is required. Thus to implement complex convolution by real NTT's several NTT's have to be implemented [A23]. So it is advantageous to define a complex number theoretic transform where complex sequence can be transformed directly. A complex NTT pair is defined as:

$$X(K) = \sum_{n=0}^{N-1} x(n) \quad \alpha^{Kn} \qquad \text{in } Q_p \tag{2.17}$$

$$x(n) = \frac{1}{N} \sum_{K=0}^{N-1} X(K) \quad \alpha^{-Kn}$$

where $x(n)$, $X(K)$, $\alpha \in Q_p$ and P a prime integer.

The field Q_p is known as the second order extension of integer field I_p. An element A of extension field Q_p can be represented as $A = a + \hat{j}b$, where $a,b \in I_p$ and $\hat{j} = \sqrt{-1}$ in Q_p. The number of elements in a finite field Q_p is P^2. Q_p is also known as Galois fields denoted by $GF(P^2)$
The complex extension field Q_p which can be represented also as $Q_p(\sqrt{-1})$ only exists if expression $x^2 + 1 = 0$ is irreducible over I_p.

The above argument can be generalised to all quadratic extension fields $Q_p(\sqrt{m})$ of I_p if eqn. $x^2 + m = 0$ is irreducible over I_p . By Euler theorem [A41] ;

$$m^{(P-1)/2} \equiv \left(\frac{m}{p}\right) \qquad \text{Mod } P \tag{2.18}$$

The quadratic eqn. $x^2 + m = 0$ Mod P is reducible over I_p if

$(\frac{m}{p}) = +1$ and $x^2 + m = 0$ and P is said to be irreducible over I_p if $(\frac{m}{p}) = -1$ and $Q_p(\sqrt{m})$ is a field of P^2 elements isomorphic to $GF(P^2)$[41] .

In this section complex number theoretic transforms are considerd for Mersenne and Fermat primes ; quadratic extension fields are considered in the next section. Consider $F_t = 2^{2^t} + 1$ a Fermat prime it can be shown that eqn. $x^2 + 1 = 0$ Mod F_t is reducible in I_{F_t} since expression $(-1)^{F_t-1/2} = (\frac{-1}{F_t}) = +1$. Thus there is no complex extension field with a Fermat prime, however there are quadratic extension fields which are explained in the next section. If Mersenne primes $P = 2^q-1$ are considered, it can easily be shown that eqn. $x^2 + 1 = 0$ Mod M is irreducible in I_p since expression $(-1)^{P-1/2} = (\frac{-1}{P}) = -1$ so complex number theoretic transform can be defined. The condition for existence of complex number theoretic transform is given below.

(a) α is root of unity or order N in $Q_p(\sqrt{-1})$

(b) $N.N^{-1} = 1$ in $Q_p(\sqrt{-1})$

(c) $N \mid (P^2-1)$ since there are P^2-1 distinct elements in $Q_p(\sqrt{-1})$

An algorithm to calculate the primitive roots in $Q_p(\sqrt{-1})$ are given in [A22].

2.5 QUADRATIC NUMBER THEORETIC TRANSFORMS [A22]

Number theoretic transform can be defined in quadratic extension fields as:

$$X(K) = \sum_{n=0}^{N-1} x(n)\ \alpha^{nK}$$

$$x(n) = \sum_{n=0}^{N-1} X(K)\ \alpha^{-nK} \qquad \text{in } Q_p(\sqrt{m}) \qquad (2.18)$$

where $Q_p(\sqrt{m})$ is quadratic extension field of I_p of P^2 distinct elements with P a prime integer. X(K), x(n), are elements of $Q_p(\sqrt{m})$ where each element is represented as $Q_p(\sqrt{m}) = a+\sqrt{m}b$ with $a, b \in I_p$.

In [A22] it is seen that quadratic extension fields exist with primes as Fermat and Mersenne integers. For example with Fermat primes $F_t = 2^{2^t} + 1$ the maximum transform length is $N_{max} = F_t^2 - 1 = 2^{2^t+1}(2^{2^t-1} + 1)$. Transforms of order 2^{2^t+1} can be defined and can be implemented by Radix-2 FFT algorithms. Similarly quadratic Number theoretic transform can be defined with Mersenne primes which are implemented by Winograds algorithm [A19].

2.6 NUMBER THEORETIC TRANSFORM IN GALOIS RINGS $GF(q^2)$ [A24]

In the previous two sections transforms were defined over Galois fields $GF(P^2)$ where P is a prime integer. Here we introduce Number theoretic transform in $GF(q^2)$ where q is a composite integer which can be factorised into its prime factors given as:

$$q = P_1 P_2 \ \ldots\ P_\ell \quad \text{and } x^2 + 1 = 0 \ \text{Mod } P_i$$

is irreducible for $i = 1,2,\ldots.\ell$.

Thus $GF(q^2)$ can be decomposed into its Galois fields represented as

$$GF(q^2) = GF(P_1^2) \otimes GF(P_2^2) \otimes \ldots \otimes GF(P^2) \qquad (2.19)$$

where every element A in $GF(q^2)$ can be represented in its factorised form $A = (A_1, A_2, \ldots, A_i, \ldots, A_\ell)$ in which $A_i \in GF(P_i^2)$.

Thus a number theoretic transform defined in $GF(q^2)$ must be satisfied in every $GF(P_1^2)$.

The condition for existence is given by:

$$\begin{aligned} &(1)\quad \alpha^N \equiv 1 \qquad \text{Mod } P_i \\ &(2)\quad NN^{-1} \equiv 1 \qquad \text{Mod } P_i \qquad \text{for } i=1,2,\ldots\ell \\ &(3)\quad N \mid P_i^2 - 1 \end{aligned} \tag{2.20}$$

where α is the root of unity of order N in $GF(q^2)$. The detailed proof is given in .

2.7 CONCLUSION AND DISCUSSION

In this chapter a review of number theoretic transform in different fields was given. Primes such as Mersenne and Fermat were discussed because of their implementation simplicity. However, other prime integers have been considerd to provide larger dynamic range [A25]. Mersenne and Fermat trnsforms are certainly very efficient to implement compared to DFT's. However, transform length is limited by word-length of the processor, but their implementation only requires shifting and addition. Along data sequence can be transformed by NTT's if it is first converted into a multi-dimensional sequence by the Chinese Remainder Theorem and then transformed using efficient algorithms such as Winograd's, prime factor and polynomial algorithms. One advantage of NTT over DFT is that arithmetic is done error free, although data has to be scaled and rounded into integers.

Since NTT's have circular convolution properties, they can be used in digital signal processing applications. Application of NTT to digital image processing has been considered in [A26]. Its use to implement circular convolution for radar signal is considered in [A27]. Other NTT's, such as pseudo-Mersenne and Fermat transforms, are discussed in [A28], [A29].

CHAPTER THREE

CHAPTER THREE

P-ADIC TRANSFORMS

3.1 Introduction

In this chapter the P-Adic number field Q_p is first introduced. Then a finite-segment P-Adic number field is defined which is known as a Hensel code. In Section 3.3 a number theoretic like transform is defined in the field of segmented P-Adic numbers. This transform is called P-Adic transform. The conditions for existence of such a transform are also derived. In Section 3.4, such P-Adic transforms are defined where prime P is chosen to be a Fermat or Mersenne prime number. Computational complexity to implement these transforms by prime factor, Winograd and polynomial transform algorithms are also considered.

3.2 P-Adic field Q_p and Segmented P-Adic field $\hat{Q}_p$

3.2.1 Introduction to P-Adic Numbers

The idea of a P-Adic field Q_p was originated by Hensel [A29]. Hensel stated that every P-Adic number α can be uniquely represented by an infinite series of the form

$$\alpha = \sum_{n=-m}^{\infty} a_n P^n , \quad a_n \in I_p \qquad (3.2.1)$$

where P is a prime integer and m an interger.

This infinite series (3.2.1) converges to α, with respect to the P-Adic norm [A30]. In Appendix B a more detailed property of the P-Adic field Q_p is given with some examples.

Any P-Adic number is represented by an infinite series given by expression (3.2.1), but in order to be able to do arithmetic a finite segment P-Adic number system has to be defined. Krishnumurthy introduced a finite segment P-Adic field, $\hat{Q}_p$, by truncating the infinite P-Adic expansion series of a P-Adic number to a fixed number of digits r [A31]. This representation of a finite segmented P-Adic expansion of a P-Adic number α is also known as a Hensel code [A32] and is denoted by $H(p,r,\alpha)$. The conditions for construction of such codes are [A33]:

(i) the numerator and denominator of the rational numbers to be represented have a prescribed bound, given by

$$-\sqrt{\frac{p^r-1}{2}} \leq |a| , |b| \leq +\sqrt{\frac{p^r-1}{2}} \qquad (3.2.2)$$

(ii) the P-Adic expansions are terminated at the right such that r is even. Thus any rational number α within the prescribed bound (3.2.2) can be represented by a Hensel code as

$$H(p,r,\alpha) = \sum_{i=-m}^{r} a_i p^i \qquad (3.2.3)$$

where $a_i \in I_p$.

3.2.2 Arithmetic Operations in Segmented P-Adic Field

The basic arithmetic operations such as Add/Subtract/Multiply/Divide are valid in $\hat{Q}_p$ provided that no overflow occurs even in computation. Consider two rational numbers $H(p,r,\alpha)$, $H(p,r,\beta)$ represented by their corresponding Hensel codes, then

$$H(p,r,\alpha * \beta) = H(p,r,\alpha) * H(p,r,\beta) \qquad (3.2.4)$$

where $*$ represents an operation.
Expression (3.2.4) is valid provided that the absolute value of the numerator and denominator of α, β, and $\alpha * \beta$ do not exceed $\sqrt{\frac{p^r-1}{2}}$. The arithmetic operations in $\hat{Q}_p$ (or Hensel codes) are almost identical to the p-ary arithmetic, since it is essentially modulo p^r arithmetic realised as a simple recursion of modulo p operations. The P-Adic arithmetic operations are error free. In Appendix B the procedures for basic P-Adic arithmetic operations are explained.

3.3 P-Adic Transforms

3.3.1 General P-Adic Transform [A34]

In this section number theoretic like transforms are defined in the finite segmented P-Adic field, $\hat{Q}_p$, and its properties are explained. We define the P-Adic transform of N point-sequence x(n) represented by their Hensel codes as $H(p,r, x(n))$ for a given (p,r) as;

$$H(p,r,X(k)) = \sum_{n=0}^{N-1} H(p,r, x(n)) \{H(p,r,\gamma)\}^{nk} \qquad (3.3.1)$$

$$\text{for } 0 \leqslant k \leqslant N-1$$

and its inverse transform

$$H(p,r,\ x(\ell)) = H(p,r,\ \frac{1}{N}\ ) \sum_{K=O}^{N-1} H(p,r,X(k)\ \{H(p,r,\gamma)\}^{-\ell k}$$

$$\text{for } O \leqslant \ell \leqslant N - 1 \qquad (3.3.2)$$

where $H(p,r,\gamma)$ is the Nth root of unity in $\hat{Q}_p$; its existence and derivation is discussed in the next section.

To derive the conditions for orthogonality (or existence of the inverse transform) we substitute eqn. (3.3.·) into eqn.(3.3.2), then

$$H(p,r,x(\ell)) = H(p,r,\tfrac{1}{N}) \sum_{k=O}^{N-1} \sum_{n=O}^{N-1} H(p,r,x(n))\ \{H(p,r,\gamma)\}^{K(n-\ell)} \qquad (3.3.3)$$

then reordering

$$= H(p,r,\tfrac{1}{N}) \sum_{n=O}^{N-1} H(p,r,x(n)) \sum_{k=O}^{N-1} \{H(p,r,\gamma)\}^{K(n-\ell)} \qquad (3.3.4)$$

Let

$$H(p,r,S) = H(p,r,\tfrac{1}{N}) \sum_{k=O}^{N-1} \{H(p,r,\gamma)\}^{K(n-\ell)} \qquad (3.3.5)$$

We get

$$H(p,r,S) = H(p,r,1) \text{ when } (n-\ell) \equiv O \text{ Mod } N$$

Since

$$\{H(p,r,\gamma)\}^{(n-\ell)N} = H(p,r,1) \text{ in } \hat{Q}_p.$$

$$H(p,r,S) = O \text{ when } (n-\ell) \not\equiv O \text{ Mod } N$$

Because we have

$$\{H(p,r,\gamma)\}^{(n-\ell)} \neq 1 \tag{3.3.6}$$

thus

$$\{H(p,r,\gamma)\}^{n-\ell} - 1 \neq 0 \tag{3.3.7}$$

Multiply H(p,r,S) by expression (3.3.7) we get

$$\begin{aligned} &= H(p,r,\tfrac{1}{N}) \; [\{H(p,r,\gamma)\}^{n-\ell} - 1] \sum_{n=0}^{N-1} \{H(p,r,\gamma)\}^{k(n-\ell)} \\ &= H(p,r,\tfrac{1}{N}) \; [\{H(p,r,\gamma)\}^{(n-\ell)N} - 1 \;] \qquad (3.3.8) \\ &= 0 \end{aligned}$$

Since

$$\{H(p,r,\gamma)\}^{N} = H(p,r,1) \text{ in } \hat{Q}_p \tag{3.3.9}$$

and since expression (3.3.7) is not zero, then H(p,r,S) = 0 is valid.

Thus conditions for the transform having DFT structure or the properties of cyclic convolution are, if and only if,

(1) $H(p,r,\gamma)$ is a root of unity of order N in the field of $\hat{Q}_p$, thus

$$\{H(p,r,\gamma)\}^{N} = H(p,r,1) \text{ in } \hat{Q}_p.$$

(2) $H(p,r,\frac{1}{N})$ should exist (or be representable) in $\hat{Q}_p$.

(3) $H(p,r,\gamma^{-1})$ should be representable.

In the next section existence and derivation of $H(p,r,\gamma)$ in $\hat{Q}_p$ are given.

3.3.2 Existence and Derivation of $H(p,r,\gamma)$ in $\hat{Q}_p$

In the previous section it was shown that one of the conditions for the existence of the P-Adic transform is that a root of unity $H(p,r,\gamma)$ of order N exists in Q_p, in other words

$$\{H(p,r,\gamma)\}^N = H(p,r,1) \quad \text{in } \hat{Q}_p \tag{3.3.10}$$

Bachman [A30] has shown that the equation $x^{p-1} = 0$ has exactly p-1 distinct roots in $\hat{Q}_p$. Thus the maximum transform length which can be used is $N_{max} = p-1$ which is determined by the chosen prime, p. We call the root of unity of order N_{max} the primitive P-Adic root $H(p,r,\gamma_\varphi)$, the powers of this primitive root will generate all the roots of unity in the given field, Q_p.

The number theoretic properties, such as the number of primitive roots is given by $\varphi(\varphi(P))$, the number of roots of order N is given by $\varphi(N)$ and the Nth root γ_N can be obtained from the primitive root γ_φ as given by $\gamma_N = \gamma_\varphi^{P-1/N}$, are all valid in the P-Adic field where φ is defined as Euler's function [A35].

3.3.2.1 To Find $H(p,r,\gamma_\varphi)$

We have to find the solution for

$$\{H(p,r,x)\}^{P-1} = H(p,r,1) \text{ in } \hat{Q}_p \tag{3.3.11}$$

or in general

$$\{H(p,r,x)\}^N = H(p,r,1) \text{ in } \hat{Q}_p$$

Let $\gamma = \{a_o a_1 \ldots a_n \ldots\}$ be the Nth root of unity in $\hat{Q}_p$. By P-Adic multiplication N times we must get

$$\gamma^N = \{a_o^N , N\, a_o^{N-1}\, a_1, \ldots\} \quad = \{\, 1,0,0 \ldots 0 \ldots \,\}$$

then by considering each digit we get

$$a_o^N = 1 \quad \text{Mod } P$$

But by Fermat-Euler's theorem [A35] $a_o^{\varphi(P)} = 1$ Mod P and since $\varphi(P) = P-1$ then $a_o^{P-1} \equiv 1$ Mod P, then for $N = P-1$ there exists a root a_o of order P-1. Once the congruence equation $a_o^{P-1} \equiv 1$ Mod P is solved for a_o, the P-Adic primitive root $H(p,r,\gamma)$ can be evaluated by using Newton's iterative method [A30] (see Appendix B). In the following section P-Adic transforms are defined for Mersenne and Fermat integer primes and the primitive roots are given.

3.4 Mersennes and Fermat's P-Adic Transforms

The basic arithmetic operations in a segmented P-Adic field, $\hat{Q}_p$, is done modulo P where P is a prime number. But operation Mod P is very costly thus simple primes such as Mersenne $P = 2^n-1$ where $n = 2,3,5,11,13,\ldots$ and Fermat prime integers such as $p = 2^{2^t} + 1$ for $t = (1,2,3,4)$ are usually chosen in order to reduce the complexity of the processor [A16]. In the following sub-sections Mersenne and Fermat P-Adic transform in $\hat{Q}_p$ are introduced.

3.4.1 Fast-Mersenne P-Adic Transform

Consider the segmented P-Adic field, $\hat{Q}_p$, with P a Mersenne prime $P = 2^n-1$ for $n = 2,3,5,11,13,\ldots$. The P-Adic transform in this field has the maximum length of $N_{max} = P - 1 = 2^n - 2 = 2(2^{n-1} - 1)$ which is not highly

composite. Thus simple radix 2-FFT algorithm cannot be used. But more complex efficient algorithms such as prime factors or Winograd algorithms can be used. In these algorithms the transform length is put into prime factorization form as shown in Table 3.4.1 for several Mersenne primes.

n	$2^n - 2$	$P_1 P_2 \ldots P_i$
2	2	2
3	6	2.3
5	30	2.3.5
7	126	$2x3^2x7$
13	8190	$2x3^2x5x7x13$

Then by the Chinese Remainder Theorem this one-dimensional P-Adic transform is mapped into a multi-dimensional P-Adic transform such that it is cyclic with prime transform length in every dimension [A36],[A38]. This multi-dimensional P-Adic transform is then implemented by Rader algorithm [A38] which is a conversion of prime length P_i transforms into circular convolutions of P_i-1 points. The circular convolutions are implemented by FFT algorithms, short convolutions or polynomial product algorithms [A39].

The P-Adic transform can also be implemented by polynomial transform algorithms introduced in Chapter five [A11].

3.4.2 Fermat P-Adic Transforms

Consider a P-adic transform with prime p a Fermat prime. A Fermat prime is given by $F_t = 2^{2^t} + 1$ for t=1,2,3,4. The maximum transform length N_{max} is given by $N_{max} = F_t - 1 = 2^{2^t}$ obtained from eqn.(3.3.10). This transform length is highly composite, thus a radix-2 FFT algorithm can be employed, so Fermat P-adic transforms can be implemented by FFT algorithms.

3.5 Conclusions

In this chapter a number theoretic like transform is defined in the field of P-adic numbers. In section 3.2.1 P-adic field and segmented P-adic field are introduced. Arithmetic in segmented P-adic field is discussed in section 3.2.2. Detailed properties of P-adic field are given in Appendix B. P-adic transforms are defined in section 3.3. The conditions for orthogonality are given in section 3.3.1. It is seen that the P-adic transforms have the same properties as that of NTT. In section 3.4 Mersenne and Fermat P-adic transforms are defined. It is seen that in section 3.4.1 the fast radix-3 FFT algorithms cannot be used for Mersenne P-adic transforms. However, they can be implemented by more complicated techniques as explained in section 3.4.1. In section 3.4.2 Fermat P-adic transforms are discussed. Hardware implementation of these transforms was not considered but the techniques discussed in [A16] can be used to implement such transforms. Only Mersenne and Fermat prime integers were discussed, other primes can also be considered.

CHAPTER FOUR

CHAPTER FOUR

COMPLEX P-ADIC TRANSFORM

4.1 Introduction

In many instances of digital signal processing, digital filtering of complex signals is required, or a complex convolution has to be implemented. In this chapter it is shown that a complex convolution can be decomposed into four real convolutions which are then implemented by four real P-adic transforms. In section 4.2.2 we show that owing to special representation of complex numbers in a P adic field with P a Fermat prime, the complex convolution reduces to two real convolutions. In sections 4.3 and 4.4 we define complex P-adic fields, K_p. It is shown that the action of this transform over K_p is equivalent to the discrete Fourier transform of a sequence of complex numbers of finite dynamic range. In section 4.5 P-adic transform is mathematically defined in G-adic fields.

4.2 Implementation of Complex Convolution Via P-adic Transforms

4.2.1 Decomposition of Complex Convolution

Consider a complex P-adic sequence $H(p,r,y_n)$ to be filtered by a complex sequence having N terms $H(p,r,b_n)$, in which $H(p,r,u_n)$ is the output P-adic sequence given by

$$H(p,r,u_m) = \sum_{n=0}^{N-1} H(p,r,b_n) \quad H(p,r,y_{m-n}) \qquad (4.2.1.1)$$

$$\text{for } n = 0,1,\ldots N-1$$

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

where

$$H(p,r,b_n) = H(p,r,h_n) + \hat{j}\, H(p,r,\hat{h}_n) \qquad (4.2.1.2)$$
$$H(p,r,y_n) = H(p,r,x_n) + \hat{j}\, H(p,r,\hat{x}_n)$$
$$H(p,r,u_m) = H(p,r,z_m) + \hat{j}\, H(p,r,\hat{z}_m)$$

where

$$\hat{j} = \sqrt{-1}$$

The complex convolution (4.2.1.1) can be decomposed into four real convolutions as given by eqn.

$$H(p,r,u_m) = \sum_{n=0}^{N-1} [H(p,r,h_n)\, H(p,r,x_{m-n}) - H(p,r,\hat{h}_n)\, H(p,r,\hat{x}_{m-n})]$$
$$+ \hat{j} \sum_{n=0}^{N-1} [H(p,r,\hat{h}_n)\, H(p,r,x_{m-n}) + H(p,r,h_n)\, H(p,r,\hat{x}_{m-n})] \qquad (4.2.1.3)$$

then

$$H(p,r,z_m) = \sum_{n=0}^{N-1} [H(p,r,h_n)\, H(p,r,x_{m-n}) - H(p,r,\hat{h}_n)\, H(p,r,\hat{x}_{m-n})] \qquad (4.2.1.4)$$

$$H(p,r,\hat{z}_m) = \sum_{n=0}^{N-1} [H(p,r,\hat{h}_n)\, H(p,r,x_{m-n}) + H(p,r,h_n)\, H(p,r,\hat{x}_{m-n})] \qquad (4.2.1.5)$$

The expression (4.2.1.3) consists of four real convolutions which can be implemented by Fast Mersenne number P-adic transforms, introduced in the previous chapter, section 3.4.1.

4.2.2 Complex Convolution via Fermat Number P-Adic Transforms [A 40]

Consider a P-adic transform defined in Q_p where p is chosen to be a Fermat number $p = 2^q + 1$ with $q = 2^t$ for $t = 1,2,3,4$. But $\hat{j} = \sqrt{-1}$ has a special representation in Q_p with p a Fermat prime. Since equation $x^2 = -1$ is solvable in $\hat{Q}_p$ the square root of -1 can be represented by a P-Adic sequence, say, $H(p,r,\hat{j})$ where p is a Fermat prime. Thus the expression (4.2.1.1) becomes an N-point real P-adic convolution.

$$H(p,r,u_m) = \sum_{n=0}^{N-1} [H(p,r,h_n) + H(p,r,\hat{j})\, H(p,r,\hat{h}_n)]$$

$$[H(p,r,x_{m-n}) + H(p,r,\hat{j})\, H(p,r,\hat{x}_{m-n})] \qquad (4.2.2.1)$$

where

$$H(p,r,u_m) = H(p,r,z_m) + H(p,r,\hat{j})\, H(p,r,\hat{z}_m) \qquad (4.2.2.2)$$

To recreate the in-phase and quadrature components $H(p,r,z_m)$ and $H(p,r,\hat{z}_m)$ of the output sample given by expression (4.2.2.2) we consider the auxiliary convolution given by (4.2.2.3);

$$H(p,r,v_m) = \sum_{n=0}^{N-1} [H(p,r,h_n) - H(p,r,\hat{j})\, H(p,r,\hat{h}_n)]$$

$$[H(p,r,x_{m-n}) - H(p,r,\hat{j})\, H(p,r,\hat{x}_{m-n})] \qquad (4.2.2.3)$$

where

$$H(p,r,v_m) = H(p,r,z_m) - H(p,r,\hat{j})\, H(p,r,\hat{z}_m) \qquad (4.2.2.4)$$

combining expressions (4.2.2.2) and (4.2.2.4) we get

$$H(p,r,z_m) = H(p,r,\tfrac{-1}{2})\ [H(p,r,u_m) + H(p,r,v_m)] \quad (4.2.2.5)$$

$$H(p,r,\hat{z}_m) = H(p,r,\tfrac{-1}{2})\ [H(p,r,u_m) - H(p,r,v_m)] \quad (4.2.2.6)$$

Thus in the P-adic field $\hat{Q}_p$ where p is a Fermat number an N-points complex P-adic convolution is implemented by two N-points real P-adic convolutions (4.2.2.2) and (4.2.2.3), instead of a conventional approach with four real P-adic convolutions. The number of operations (multiplications and additions) is reduced but the arithmetic operation must be performed in $\hat{Q}_p$ with p a Fermat number. The P-adic convolutions (4.2.2.1) and (4.2.2.3) are then implemented by Fermat number P-adic transform.

4.3 P-adic Transform in Extension Fields of Q_p [A40]

4.3.1 Introduction to Extension Fields of Q_p

The P-adic field Q_p has infinitely many distinct algebraic extension fields, all the fields being generated by roots of algebraic equations $x^n - p = 0$ ($n = 2,3,4,\ldots$). For simplicity only quadratic extension fields of Q_p are considered. In [A42] Mahler has shown that for $p \geq 3$ there are exactly three distinct quadratic extensions of Q_p and these may be represented by

$$Q_p(\sqrt{N_p}) \ , \quad Q_p(\sqrt{p}) \ , \quad Q_p(\sqrt{pN_p}) \quad (4.3.1.1)$$

where N_p is the smallest positive quadratic non-residue, or on the other hand the equation $x^2 - N_p = 0$ is said to have no solution in Q_p. By Euler's theorem [A41] this is further

equivalent to

$$\left(\frac{N_p}{P}\right) = (N_p)^{(N_p-1/2)} = -1 \qquad (4.3.1.2)$$

where $\left(\frac{m}{p}\right)$ is the Legendre symbol defined by

$$\left(\frac{m}{p}\right) = +1 \quad \text{if m is a quadratic residue modulo p.}$$
$$= -1 \quad \text{if m is a quadratic non-residue modulo p.}$$

Any one of the three quadratic extension fields (4.3.1.1) can be denoted by $K_p = Q_p(\sqrt{d})$. Every element z of K_p can be written in the form

$$z = x + \hat{j}y$$

where

$$x,y \in Q_p$$

and

$$\hat{j} = \sqrt{d} \quad .$$

In the following sections((4.3.2) and (4.3.3)) we will define the P-adic transforms in the three extension fields of Q_p with P a prime Mersenne number.

4.3.2 P-adic Transform in $Q_p(\sqrt{N_p})$ with P a Mersenne Prime [A40]

4.3.2.1 Introduction

Consider the quadratic extension field $K_p = Q_p(\sqrt{N_p})$ with P a Mersenne prime

$$P = 2^q - 1 \qquad q = 2,3,5,7,13,...$$

Since $x^2 \equiv -1$ Mod P is not soluble in Q_p, or by Euler's theorem

$$(-\frac{1}{p}) = (-1)^{\frac{p-1}{2}} = (-1)^{\frac{2^q-2}{2}} = (-1)^{2^{q-1}-1} = -1$$

(4.3.2.1)

Thus $N_p = -1$ is the smallest quadratic non-residue modulo Mersenne prime, and the polynomial $P(x) = x^2 + 1$ is said to be irreducible in Q_p.

Let us informally represent $\hat{i} = \sqrt{-1}$ as a root of the polynomial $P(x) = x^2 + 1$ satisfying $\hat{i}^2 = -1$ where $\hat{i}$ is an element of the extension field $Q_p(\sqrt{-1})$ which is composed of the set

$$K_p = Q_p(\sqrt{-1}) = \{A + \hat{i}B\}$$

where A,B Q_p

$\hat{i} \in K_p$ plays a similar role over the finite field Q_p that $\sqrt{-1} = i$ plays over the field of rationals Q. For example the basic arithmetic operations in K_p are similar to complex arithmetic

$$(a + \hat{i}b) \pm (c + \hat{i}d) = (a \pm c) + \hat{i}(b \pm d)$$
$$(a + \hat{i}b)(c + id) = (ac-bd) + \hat{i}(bc + ad)$$

where $(a + \hat{i}b)$, $(c + \hat{i}d) \in K_p$ and $a,b,c,d \in Q_p$. Thus $(a + \hat{i}b)$ and $(c + \hat{i}d)$ behave similarly to complex number. The other two distinct extension fields are $Q_p(\sqrt{P})$ and $Q_p(\sqrt{-P})$.

4.3.2.2 P-adic Transforms in Extension Field $Q_p(\sqrt{-1})$

We define a complex P-adic transform pair in $Q_p(\sqrt{-1})$ as

$$\langle X_K \rangle = \sum_{n=0}^{d-1} \langle x_n \rangle \langle \gamma \rangle^{nK} \quad \text{for } 0 \leq K \leq d-1$$

$$\langle x_\ell \rangle = \langle \frac{1}{d} \rangle \sum_{K=0}^{d-1} \langle X_K \rangle \langle \gamma \rangle^{-\ell K} \quad \text{for } 0 \leq \ell \leq d-1$$

(4.3.2.2)

where

$$\langle x_n \rangle \ , \ \langle X_K \rangle \quad \langle \gamma \rangle \ , \quad \varepsilon \ Q_p(\sqrt{-1})$$

and P is a Mersenne prime.

Conditions for existence are similar to that of complex number theoretical transforms [A15].

(a) γ is a root of unity of order d in $Q_p(\sqrt{-1})$ in order to have cyclic convolution property (CCP), thus $\gamma^d = 1$ in $Q_p(\sqrt{-1})$.

(d) $\frac{1}{d}$ exists in $Q_p(\sqrt{-1})$.

(c) d divides (p^2-1).

The number of elements in $Q_p(\sqrt{-1})$ is p^2, as in Galois field of p^2 elements $GF(p^2)$ [A22], the maximum order t of the multiplicative group with α as the generator is

$$t = p^2-1 = (2^q-1)^2 - 1 = 2^{q+1}\,(2^{q-1}-1) \qquad (4.3.2.3)$$

thus

$$\alpha^{p^2-1} = 1 \text{ in } Q_p(\sqrt{-1}).$$

γ as the generator of order d can be obtained by $\gamma = \alpha^{\frac{(p^2-1)}{d}}$ as in the number theoretic transform, which gives the third condition given above.

For $d = 2^{q+1}$ the radix-2 FFT algorithm can be used.

In the previous chapter we defined a segmented P-adic field $\hat{Q}_p$ in order to be able to do P-adic arithmetic on a finite wordlength processor. Thus to do the arithmetic in the extension fields we have to form a finite segmented

extension field $\hat{K}_p = \hat{Q}(\sqrt{-1})$, which is a subset of K_p. This is obtained by truncating the P-adic series representation of the elements in K_p to r digits as in chapter three. This will produce a finite set $\hat{K}_p$ with elements presented by

$$H(p,r,z) = H(p,r,A) + \hat{i}\, H(p,r,B)$$

where

$$H(p,r,A) \text{ or } H(p,r,B) = > \sum_{n=-m}^{r} C_n P^n \qquad (4.3.2.4)$$

where $C_n \varepsilon I_p$.

$$H(p,r,z) = \sum_{n=-m}^{r} Z_n P^n \qquad (4.3.2.5)$$

where $Z_n = a_n + \hat{i} b_n$.

The P-adic transform pair defined in $\hat{K}_p$ have to satisfy the same conditions as the P-adic transform pairs defined in K_p, with an additional dynamic range constraint such that if $(\frac{a}{b} + \hat{i}\,\frac{c}{d}) \varepsilon \hat{K}_p$ then

$$-\sqrt{\frac{P^r-1}{2}} \leqslant a,b,c,d \leqslant \sqrt{\frac{P^r-1}{2}} \qquad (4.3.2.6)$$

Thus to take transforms over $\hat{K}_p$ of a d-points sequence of complex numbers, $H(p,r,x_n) \varepsilon \hat{K}_p$, we have to find the root of unity γ of order d. The complex convolution is now implemented by two forward complex P-adic transforms and one inverse transform.

4.3.3 P-adic Transform Defined in $Q_p(\sqrt{P})$ and $Q_p(\sqrt{PN_p})$, with P a Mersenne Prime

In the previous section we defined P-adic transforms in $Q_p(\sqrt{N_p})$ where $N_p = -1$ and P a Mersenne prime. The other

two distinct fields are $Q_p(\sqrt{P})$ and $Q_p(\sqrt{-P})$. These are known as ramified extension fields because the elements in this field have a special representation [A42] as

$$Z = x + \sqrt{P}\ y = \sum_{n=-m}^{\infty} Z_n \hat{P}^n \qquad (4.3.3.1)$$

where $Z_n \in I_p$, $\hat{P} = \sqrt{P}$ $P = E\hat{P}^2$ and E is given by

$$E = 1 \qquad \text{if } K_p = Q_p(\sqrt{P}) \qquad P > 3$$
$$E = -1 \qquad \text{if } K_p = Q_p(\sqrt{-P}) \qquad P > 3$$

Thus an element in these two fields is represented by a series of power $\hat{P}$ with real coefficients $Z_n \in I_p$ as given by expression (4.4.1). We define the transform pair in $Q_p(\sqrt{P})$ as

$$X_K = \sum_{n=0}^{d-1} x_n \beta^{nK} \qquad \text{for } 0 \leq K \leq d-1 \qquad (4.3.3.2)$$

$$x_\ell = \frac{1}{d} \sum_{K=0}^{d-1} X_K \beta^{-\ell K} \qquad \text{for } 0 \leq \ell \leq d-1 \qquad (4.3.3.3)$$

where x_n, X_K, $\beta \in Q_p(\sqrt{P})$ and P is a Mersenne prime. The conditions given in section(4.3.2.2)must be satisfied for this pair of transforms. β is a root of unity of order d in $Q_p(\sqrt{P})$. The arithmetic in $Q_p(\sqrt{P})$ is similar to the arithmetic in complex P-adic field where here $\sqrt{P}$ plays the role of $j = \sqrt{-1}$ in $Q_p(\sqrt{P})$.

4.4 Quadratic P-adic Transform Defined in Extension Fields K_p with P a Fermat Prime

In this section a transform over the extension field, K_p, with P a Fermat prime $P = 2^{2^t} + 1$ for t = 1,2,3,4 is considered. As in the previous sections there are three distinct extension fields given by $K_p = Q_p(\sqrt{N_p})$, $Q_p(\sqrt{P})$ and $Q_p(\sqrt{PN_p})$ where N_p is the smallest integer non-residue modulo a Fermat prime. However a complex P-adic transform (where $N_p = -1$) does not exist with P a Fermat prime. Similar to the previous sections a transform pair can be defined in K_p. The maximum transform length t is given by $t = P-1 = 2^{2^n}$. Such a transform length is highly composite so radix 2-FFT algorithms can be used to implement these transforms.

4.5 Transform Defined in g-adic Ring over a Direct Sum of Several P-adic Fields

It is well known that any number α can be represented in a g-adic ring Q_g by an infinite series

$$\alpha = \sum_{n=-m}^{\infty} a_n g^n \tag{4.5.1}$$

Such an infinite series is convergent in g-adic norm [A42]. It is also known that a g-adic ring, Q_g, is a direct sum of P-adic fields Q_{P_k} for k = 1,2,...n. In the language of algebra

$$Q_g = Q_{P_1} \hat{\oplus} Q_{P_2} \hat{\oplus} Q_{P_3} \hat{\oplus} Q_{P_4} \hat{\oplus} \cdots Q_{P_n} \tag{4.5.2}$$

Where $\hat{\oplus}$ is the algebraic summation.

To define a Fourier like transform over Q_g is the same as defining it over the direct sum of the fields Q_{P_1}, Q_{P_2} ... Q_{P_n}.

Thus any element A of Q_g can be represented by its P-adic components

$$A = < A_1, A_2, \ldots\ldots A_n > \tag{4.5.3}$$

Then the basic arithmetic operations can be done in each P-adic field. For example two g-adic numbers made of n P-adic components,

$$\begin{aligned} A &= < A_{P_1}, A_{P_2}, \ldots A_{P_K} \ldots A_{P_n} > \\ B &= < B_{P_1}, B_{P_2}, \ldots B_{P_K} \ldots B_{P_n} > \\ \therefore \quad AB &= < A_{P_1} B_{P_1}, A_{P_2} B_{P_2}, \ldots A_{P_k} B_{P_k} \ldots A_{P_n} B_{P_n} > \end{aligned} \tag{4.5.4}$$

Therefore the arithmetic in each component can be done in parallel, provided that we can reconstruct the number from its components. Thus a Fourier-like transform defined over Q_g for a g-adic sequence a_n n = 0, ... d-1 is given by

$$\begin{aligned} A_\ell &= \sum_{n=0}^{d-1} a_n \gamma^{n\ell} \quad \text{in } Q_g \qquad \text{for } \ell = 0,1 \ldots d-1 \\ a_n &= \frac{1}{d} \sum_{\ell=0}^{d-1} A_\ell \gamma^{-n\ell} \quad \text{in } Q_g \qquad \text{for } n = 0,1,\ldots d-1 \end{aligned} \tag{4.5.5}$$

where

$$A_\ell, a_n, \gamma \in Q_g$$

or

$$\begin{aligned} A_\ell &= [(A_\ell)_{P_1}, \ldots (A_\ell)_{P_k}], \\ a_n &= [(a_n)_{P_1}, \ldots (a_n)_{P_k}] \end{aligned}$$

The condition for orthogonality is that

$$\gamma^d = 1 \text{ in } Q_g \tag{4.5.6}$$

and since γ is given by its components as

$$\gamma = [\gamma_{P_1}, \gamma_{P_2}, \cdots \gamma_{P_n}]$$

where

$$\gamma_{P_n} \varepsilon \; Q_{P_n}$$

then each component of γ should satisfy

$$\gamma_{P_k} = 1 \text{ in } Q_{P_k} \text{ for } k = 1,2,\ldots n$$

From above it is seen that the transform over Q_g is the same as a direct sum of transform over $\{Q_{P_1}, Q_{P_2}, \ldots, Q_{P_n}\}$ in parallel, giving a larger dynamic range.

4.6 Conclusions

In this chapter complex P-adic transforms are introduced. Implementation of complex convolutions is discussed in section 4.2, complex convolution in P-adic field is defined and in section 4.2.2 its implementation by Fermat P-adic transforms is considered. It is shown that complex P-adic convolution is implemented by two real P-adic convolutions if prime P is chosen to be a Fermat prime integer. This algorithm is similar to that of Nussbaumer [A23] where a complex convolution is implemented by Fermat transforms.

In section 4.3 extension fields of Q_p are considered. P-adic transforms are defined in the extension fields. Only Quadratic extensions are considered. It is seen that there are exactly three distinct quadratic extension fields of Q_p. With p a Mersenne prime integer, complex P-adic transforms are defined in section 4.3.2.2. In section 4.3.3 P-adic transforms in other extension fields are discussed. From section 4.3 it is seen that a complex P-adic convolution can be implemented directly by complex P-adic transforms.

In this chapter it is also shown that no complex P-adic transforms exist if p is chosen as a Fermat prime integer. A g-adic transform is defined in a g-adic ring in section 4.5. Since a g-adic ring is a direct sum of several P-adic fields, it is seen that a transform in a g-adic ring has to satisfy all its P-adic field components. The condition for existence of these transforms is also discussed in section 4.5.

Since rational complex number can be represented exactly in complex P-adic fields, complex P-adic transforms are preferred to complex number theoretic transforms where scaling is required to convert rational numbers into integers. Also, a larger dynamic range is achieved if P-adic transforms are used.

CHAPTER FIVE

CHAPTER FIVE

LINEAR TRANSFORMS

5.1 INTRODUCTION

The main objective of this chapter is to introduce some linear orthogonal transforms and some new implementation techniques. In section 5.2 Discrete Fourier transform is discussed; Hadamard transform is explained in section 5.3; Cosine transform is introduced in section 5.4,with some of its implementation techniques, and section 5.5 is devoted to polynomal transform algorithms and their application to implementation of convolution and transforms. This chapter is believed to be a short summary of the transforms and their implementations. The application of linear transforms to image coding are discussed in the following chapters, and other applications are discussed in Chapter one.

5.2 DISCRETE FOURIER TRANSFORM

The DFT of a sequence x(n) of length N is defined by:

$$X(K) = \sum_{n=0}^{N-1} x(n) \; W_N^{nK} \qquad (5.1)$$

where X(K) are the transformed coefficients and K = 0,1,... N-1, and $W_N = e^{\frac{j2\pi}{N}}$. The DFT has the property that x(n) and X(K) are uniquely related by a transform pair consisting of eqn. (5.1) and its inverse:

$$x(n) = N^{-1} \sum_{K=0}^{N-1} X(K) \; W_N^{-nK} \qquad (5.2)$$

The importance of Fourier transform pair is that it can implement digital convolution very efficiently. Consider a one-dimensional convolution

$$y(\ell) = \sum_{n=0}^{N-1} h(n)\, x(n-\ell) \qquad (5.3)$$

This is implemented in the transform domain by direct multiplication given as:

$$\gamma(K) = H(K)\ X(K) \qquad (5.4)$$

where $\gamma(K)$, $H(K)$ and $X(K)$ are the transformed coefficients of $y(n)$, $\ell(n)$ and $x(n)$.

The eqn. (5.2) can be implemented efficiently by fast algorithms such as Fast Fourier transform algorithm (FFT) [A12], Prime factor algorithm [A20], Winograd's algorithm [A19] and Polynomial algorithms [A11].

5.3 HADAMARD TRANSFORM

One-dimensional hadamard is defined as

$$H(u) = \sum_{x=0}^{N-1} f(x)\ (-1)^{\sum_{i=0}^{n-1} b_i(x) b_i(u)} \qquad (5.5)$$

$$f(x) = \frac{1}{N} \sum_{x=0}^{N-1} H(u)\ (-1)^{\sum_{i=0}^{n-1} b_i(x) b_i(u)} \qquad (5.6)$$

where the summation in the exponent is performed in modulo 2 and $b_K(z)$ is the Kth bit in the binary representation of z. Hadamard transform has a particular application in transform

coding because of its computational simplicity.

5.4 DISCRETE COSINE TRANSFORM

A one-dimensional discrete cosine transform (DCT) is defined as:

$$C(K) = 2 \sum_{n=0}^{N-1} x(n) \cos \frac{\pi(2n+1)K}{2N} , \qquad (5.7)$$

$$\text{for } 0 \leq K \leq N-1$$

and its inverse:

$$x(n) = \frac{1}{N} \left[\frac{C(0)}{2} + \sum_{K=1}^{N-1} C(K) \cos \frac{\pi(2n+1)K}{2N} \right] \qquad (5.8)$$

$$\text{for } 0 \leq n \leq N-1$$

where x(n) is a N-point real sequence.

It is known that the DCT of x(n) can be obtained from 2N-point DFT of x(n). x(n) is extended to 2N-point by padding N zeroes to it. DCT can be shown to be obtained from [A44].

$$C(K) = 2 \text{ real } \left[W_{2N}^{K/2} \sum_{n=0}^{2N-1} x(n) W_{2N}^{nK} \right] \qquad (5.9)$$

Similarly a two-dimensional cosine transform is defined as:

$$C(K_1,K_2) = 4 \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} x(n_1,n_2) \cos \frac{\pi(2n_1+1)K_1}{2N_1} \cos \frac{\pi(2n_2+1)K_2}{2N_2}$$

$$\text{for } 0 \leq K_1 \leq N_1-1 \ , \ 0 \leq K_2 \leq N_2 - 1$$

(5.10)

and similarly the inverse is given by

$$x(n_1,n_2) = \frac{1}{N_1 N_2} \sum_{K_1=0}^{N_1-1} \sum_{K_2=0}^{N_2-1} C'(K_1,K_2) \cos \frac{\pi(2n_1+1)K_1}{2N_1} \cos \frac{\pi(2n_2+1)K_2}{2N_2}$$

(2.11)

where

$$C'(K_1,K_2)=\begin{cases} C(0,0)/4 & K_1 = 0 \ , \ K_2 = 0 \\ C(K_1,0)/2 & K_1 \neq 0 \ , \ K_2 = 0 \\ C(0,K_2)/2 & K_1 = 0 \quad K_2 \neq 0 \\ C(K_1,K_2) & K_1 \neq 0 \quad K_2 \neq 0 \end{cases} \tag{5.12}$$

It can be shown that the two-dimensional DCT can be obtained from a $2N_1 \times 2N_2$ point DFT of $x(n_1,n_2)$ given by

$$C(K_1,K_2) = 4 \text{ real } \left[W_{2N_1}^{K_1/2} \ W_{2N_2}^{K_2/2} \sum_{n_1=0}^{2N_1-1} \sum_{n_2=0}^{2N_2-1} x(n_1,n_1) \ W_{2N_1}^{n_1K_1} \ W_{2N_2}^{n_2K_2} \right] \tag{5.13}$$

Since DCT is believed to have the best compression ability compared with other Orthogonal transforms [A44], its application in digital image processing is vital. Here some fast techniques of implementation of DCT are considered, because DCT has been used in our coding algorithms. Recently Makhoul has introduced a fast technique of implementation of two-dimensional DCT [A45], where the input sequence is rearranged as given by

$$V(n_1,n_2) = \begin{cases} x(2n_1,2n_2) & 0 \leq n_1 < \frac{N_1}{2} \ , \ 0 \leq n_2 < \frac{N_2}{2} \\ x(2N_1-2n_1-1,2n_2) & \frac{N_1}{2} \leq n_1 < N_1 \ , \ 0 \leq n_2 < \frac{N_2}{2} \\ x(2n_1,2N_2-2n_2-1) & 0 \leq n_1 < \frac{N_1}{2} \ , \ \frac{N_2}{2} \leq n_2 < N_2 \\ x(2N_1-2n_1-1,2N_2-2n_2-1) & \frac{N_1}{2} \leq n_1 < N_1 \ , \ \frac{N_2}{2} \leq n_2 < N_2 \end{cases} \tag{5.14}$$

Then $C(K_1,K_2)$ can be obtained from the two-dimensional $(N_1 \times N_2)$-point DFT of $v(n_1,n_2)$ as given by

$$V(K_1,K_2) = \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} v(n_1,n_2)\, W_{N_1}^{n_1K_1}\, W_{N_2}^{n_2K_2} \tag{5.15}$$

$$C(K_1,K_2) = 2 \text{ real } \{W_{4N_1}^{K_1} [W_{4N_2}^{K_2} V(K_1,K_2) + W_{4N_2}^{-K_2} V(K_1,N_2-K_2)]\} \tag{5.16}$$

The number of multiplications is thus reduced by a factor of 4 since only a $N_1 \times N_2$ point DFT is performed.

Eqn. (5.16) can be implemented by polynomial transform algorithm given in [A46], where expression (5.16) is implemented as

$$C(K_1,K_2) = 2 \text{ real } \{ E_{K_1,K_2} + j\, E_{K_1,N_2-K_2}\} \tag{5.17}$$

where $j = \sqrt{-1}$ and

$$E_{K_1,K_2} = W_{4N_1}^{K_1}\, W_{4N_2}^{K_2}\, V(K_1,K_2) \tag{5.18}$$

so the bulk of the computation is the implementation of E_{K_1,K_2}.

E_{K_1,K_2} can be implemented by first performing row-column permutation $(4n_1 + 1)n_2$ modulo $N_2 = 2^{t_2}$ and by computing N_2 odd DFT's along the lines, with

$$E(n_2,z) = \sum_{K_1=0}^{N_1-1} \left[\sum_{n_1=0}^{N_1-1} v(n_1,(4n_1+1)n_2)\, W_{2N_1}^{(4n_1+1)K_1} \right] z^{K_1} \tag{5.19}$$

$E(K_1,K_2)$ is then obtained by a complex inverse polynomial transform

$$\bar{E}(K_2,z) \equiv \sum_{n_2=0}^{N_1-1} E(n_2,z)\, z^{-\frac{4n_2K_2}{N_2}} \quad \text{Modulo } (z^{N_1}-j) \tag{5.20}$$

$$\bar{E}(K_2,z) = \sum_{K_1=0}^{N_1-1} E(K_1,K_2)\, W_{2N_2}^{-K_2}\, z^{K_1} \tag{5.21}$$

The expression (5.17) is implemented to obtain $C(K_1,K_2)$. By the above algorithm a large number of multiplications and additions are saved compared to Makhoul's method, the details of the algorithm can be found in [A46]. The above algorithm is implemented in Fortran for our image coding techniques where expression (5.20) is implemented by radix-2 FFT polynomial algorithm since $N_1 = 2^{t_1}$ and $N_2 = 2^{t_2}$. The concept of the polynomial transform is given in the next section.

5.5 POLYNOMIAL TRANSFORM ALGORITHMS

Recently several new techniques of implementing 2D-DFT have been introduced by Nussbaumer [A47], using polynomial transform algorithms. These algorithms are not explained here but the concept of polynomial transform is discussed. Polynomial transforms can be viewed as discrete Fourier like transforms defined in a residue class polynomial ring $R[z]/P(z)$, where R is a ring or a field. The polynomial algebra is performed modulo $P(z)$, in which the coefficients of the polynomials are taken to lie in R.

A general definition of polynomial transforms can be obtained by considering a polynomial convolution $Y_\ell(z)$ of length N defined modulo a polynomial $P(z)$, with

$$Y_\ell(z) = \sum_{m=0}^{N-1} H_m(z)\, X_{\ell-m}(z) \quad \text{Modulo } P(z) \tag{5.22}$$

$$H_m(z) = \sum_{n=0}^{b-1} h_{n,m} \, z^n \tag{5.23}$$

$$X_r(z) = \sum_{s=0}^{b-1} x_{s,r} \, z^s \tag{5.24}$$

where

$$\{Y_i(z), H_i(z), X_i(z)\} \; \epsilon \; \frac{R[z]}{P(z)}$$

with

R the field of rational numbers

and b the degree of P(z).

The one-dimensional polynomial convolution can be transformed into N element-by-element multiplications in R[z]/f(z) by a polynomial transform defined as [A48],

$$\bar{H}_K(z) \equiv \sum_{m=0}^{N-1} H_m(z) \, [G(z)]^{mK} \quad \text{Modulo } P(z), \quad K = 0,1,\ldots N-1 \tag{5.25}$$

and similarly for

$$\bar{X}_K(z) \equiv \sum_{r=0}^{N-1} X_r(z) \, [G(z)]^{rK} \text{ Modulo } P(z), \quad K = 0,1,\ldots N-1 \tag{5.26}$$

where

{G(z)} is an nth primitive root of unity in R[z]/P(z)

then

$$\bar{Y}_K(z) \equiv \bar{H}_K(z) \; . \; \bar{X}_K(z) \quad \text{Modulo } P(z) \quad \text{for } K = 0 \text{ to } N-1 \tag{5.27}$$

and $\{Y_\ell(z)\}$ can be recovered from $\{\bar{Y}_K(z)\}$ using the inverse polynomial transform given by

$$Y_\ell(z) = N^{-1} \sum_{K=0}^{N-1} \bar{Y}_K(z) \, [G(z)]^{-\ell K} \quad \text{Modulo } P(z) \quad \text{for } \ell=0 \text{ to } N-1. \tag{5.28}$$

To establish that the polynomial transforms support circular convolution the transforms $\bar{H}_K(z)$ and $\bar{X}_K(z)$ of $H_m(z)$ and $X_r(z)$ are calculated. Then element by element multiplications of $\bar{H}_K(z)$ by $\bar{X}_K(z)$ modulo P(z) is evaluated, and inverse transform of $\bar{Y}_K(z)$ is computed. This can be represented as

$$Y_\ell(z) \equiv N^{-1} \sum_{m=0}^{N-1} \sum_{r=0}^{N-1} H_m(z)\, X_r(z) \sum_{K=0}^{N-1} [G(z)]^{(m+r-\ell)K} \tag{5.29}$$

The expression (5.29) is valid provided the three following conditions are met [A49];

(1) G(z) is an nth primitive root of unity R[z]/P(z);

$$[G(z)]^N \equiv 1 \quad \text{Modulo } P(z) \tag{5.30}$$

(2) N and G(z) have inverses modulo P(z)

(3) $$S \equiv \sum_{K=0}^{N-1} [G(z)]^{(m+r-\ell)K} \quad \text{modulo } P(z) \equiv \begin{cases} 0 \text{ for } (m+r-\ell) \not\equiv 0 \text{ Modulo N} \\ N \text{ for } (m+r-\ell) \equiv 0 \text{ Modulo N} \end{cases}$$

The polynomial transforms have the same structure as DFTS, but with complex exponential roots of unity replaced by polynomials G(z) and with all operations defined modulo P(z). If the Kernel G(z) of the polynomial transform can be chosen as a power of z, the resulting transforms will involve only the multiplication of polynomials by powers of the nth root z. This multiplication is done by shifting the coefficients of polynomials which can be done very efficiently in hardware.

Arambepola and Rayner [A50] investigated the classes of polynomial rings that possess simple Kernel G(z). They found that there exists always a polynomial transform with simple primitive roots G(z) = z[or z^t where (t,r) = 1] in the ring $R[z]/C_r(z)$ which satisfies the above three conditions (5.30) and $C_r(z)$ is a cyclotomic polynomial of order N defined as

$$C_r(z) = \prod_{Y_i \epsilon S} (z - Y_i) \tag{5.31}$$

Here S is the set, containing all primitive r^{th} roots of unity. The principal application of polynomial transforms concerns the computation of two-dimensional circular convolutions. Nussbaumer [A51] has shown a circular convolution of size N x N which can be represented as a polynomial convolution of length N where all polynomials are defined modulo (z^N-1) as given by

$$y_{u,\ell} = \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} h_{n,m}\, x_{u-n,\ell-m} \qquad u = 0,\ldots,N-1 \tag{5.32}$$

$$Y_\ell(z) \equiv \sum_{m=0}^{N-1} H_m(z)\, X_{\ell-m}(z) \quad \text{Modulo } (z^N-1) \qquad \ell = 0,1,\ldots N-1 \tag{5.33}$$

$H_i(z)$ and $X_i(z)$ are defined by expressions (5.23) and (5.24) respectively.

For coefficients in the field of rationals R[z], z^N-1 is the product of d cyclotomic polynomials $C_{r_i}(z)$, where d is the number of divisors r_i of N, including 1 and N, with

$$z^N - 1 = \prod_{i=1}^{d} C_{r_i}(z) \tag{5.34}$$

The degree of each cyclotomic polynomial $C_{r_i}(z)$ is $\Phi(r_i)$, (where $\Phi(r_i)$ is Euler's totient function and is equal to the number of positive integers smaller than r_i, which are prime

to r_i including 1).

Since the various polynomials $C_{r_i}(z)$ are irreducible, the polynomial convolution defined modulo (z^N-1) can be computed separately modulo each cyclotomic polynomial $C_{r_i}(x)$, with reconstruction of the final result by the Chinese remainder theorem defined for polynomials as explained in Appendix A for integers.

Let us now consider expression (5.33), if N = P, where P is an odd prime then z^P-1 is the product of two cyclotomic polynomials.

$$\begin{aligned} z^P-1 &= (z-1)\,P(z) \\ P(z) &= z^{P-1} + z^{P-2} + \ldots\ldots + 1 \end{aligned} \tag{5.35}$$

Now the polynomial convolution is defined modulo P(z) and modulo (z-1)

Similarly if $N = 2^t$ then

$$z^{2^t} - 1 = (z-1) \prod_{i=1}^{t} z^{2^{t-i}} + 1 = (z^{2^{t-1}} + 1)\,(z^{2^{t-1}} - 1) \tag{5.36}$$

Thus the polynomial convolution is evaluated for each modulo $(z^{2^{t-i}} + 1)$ for $i \in (1,2,\ldots)$, the final result is obtained by the Chinese remainder theorem [A52].

Arambepola and Rayner [A53] have also defined a mapping which translates a circular convolution into skew circular one and vice versa, this mapping will result in a polynomial convolution modulo a cyclotomic polynomial which will remove the need for a Chinese remainder theorem decomposition. They have also defined several new algorithms given in [A49].

Recently Nussbaumer has defined inverse polynomial transforms [A55] with their applications to convolution and DFT.

5.6 CONCLUSION

Several orthogonal transforms were reviewed in this chapter. Hadamard transform was discussed because of its implementation simplicity. The application of Hadamard transform in coding will be discussed in the following chapters. Cosine transform was discussed in more detail and several implementation techniques were reviewed. The major application of DCT has been in coding which is discussed in transform coding chapters. Finally, polynomial transforms were discussed. These transforms are very efficient in implementing multi-dimensional transforms and convolutions. Other algorithms, such as prime-factor algorithms [A20], Winograds algorithm [A19] and other polynomial algorithms [A55] were not discussed. However, some of these algorithms are still to be investigated. Most of the algorithms are concerned with reducing the number of multiplications and additions. No attention is paid to the complexity of the algorithms such as rearrangement and permutation of data when the number of multiplications and additions are reduced.

IMAGE CODING

AND

ITS APPLICATIONS

CHAPTER SIX

CHAPTER SIX

TRANSFORM CODING

6.1 Introduction

In digital communication networks such as are used in the transmission of speech and data it is advantageous to develop techniques that exploit the redundancies in the digital signal, in order to reduce storage or transmission bit rate. This is known as source coding. In this chapter classical transform coding techniques are reviewed and several new techniques are developed. Other source coding techniques are discussed in Chapter Nine. In Section 6.2 a digital image communication system is introduced. Transform image source coders are reviewed in Section 6.3, and the effect of block size and overlapping is considered. In Section 6.4 a new zonal coding strategy is developed using vector quantization.

6.2 Introduction to Image Coding

A simple model for a digital system is shown in Fig. 6.2.1.

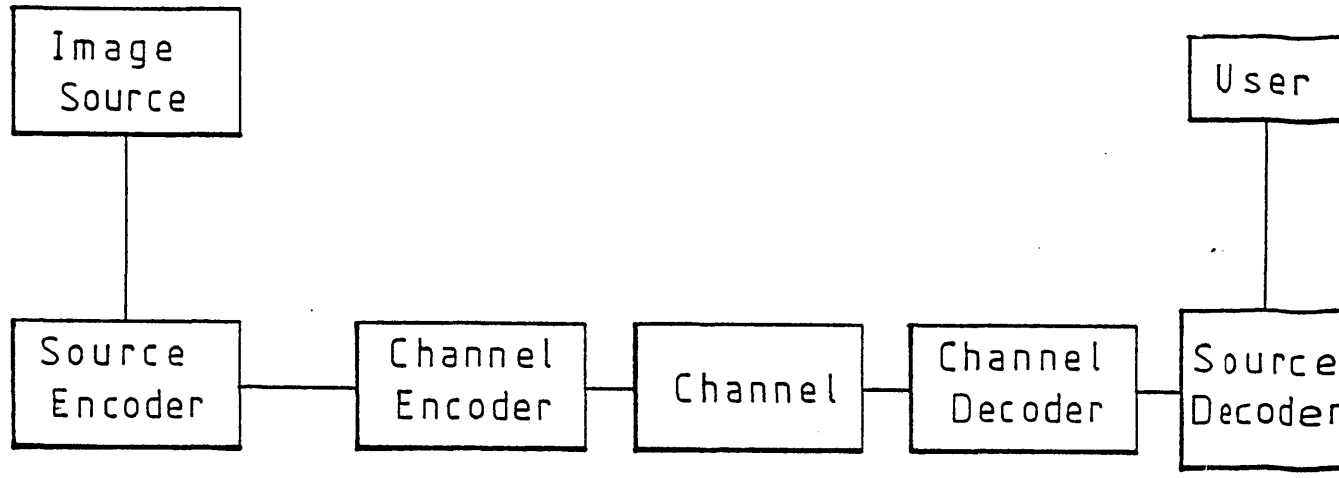


Fig. 6.2.1: A simple digital transmission system

A variable rate coding system is shown in Fig. 6.2.2, where a a buffer is used to smooth the data rate with the use of a controller.

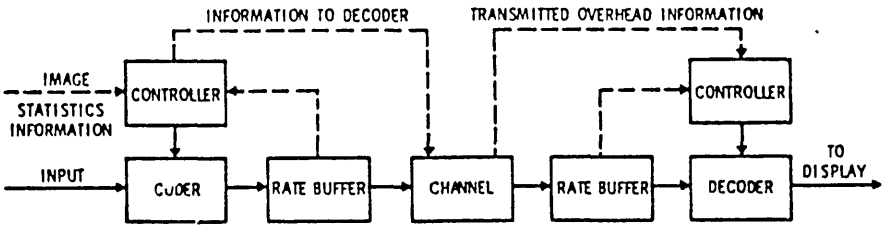


Fig. 6.2.2: A variable rate coding system [16B]

The model is made of an image source which is the sampled and quantized picture elements obtained from a television camera or facsimile scanner. The source encoder transforms the source data into a form that may reduce the number of bits required to represent each picture element. Next, the coded image source is converted to a format suitable for transmission. This step involves modulation of the transmission carrier and addition of check bits to each code-word in order to implement an error detection or correction at the decoder. The channel will introduce some noise independently to each bit of the transmitted code-word. The channel decoder and source decoders invert the coding processes to produce a reconstructed image source.

In our discussion we are only interested in the source coding; its algorithms; how they relate to the image signal it encodes; how the bit rate can be reduced by exploiting the source signal statistics and properties of human perception; the variety of quality criteria; the coder complexity;and, above all, how these phenomena are interrelated, and can be traded to approach an optimum design.

As mentioned above, the possibility of bandwidth reduction is indicated by two observations. First, there is a large amount of statistical redundancy or correlation in normal images. For example, two points that are spatially close together tend to have nearly the same brightness level. Encoding techniques that take into account the image statistics are known as statistical image coding [1B], [2B]. Unfortunately, statistical measures,means,covariances and first-order probability density functions are not a complete measure of picture structure. Pictures contain significantly more structure than is represented by the first and second order moments. Also pictorial data are not homogeneous, different regions of a picture contain different structures.

Second, there is a large amount of psychovisual redundancy in most images. The sensitivity of the human visual system to errors in the reconstructed picture depends on the frequency spectrum of the error, the gray level and amount of detail in the picture in the vicinity of the error. For example, the eye's sensitivity to distortion decreases with brightness and decreases with frequency. Hence it is possible to

increase the efficiency of the coder by allowing distortions that do not degrade subjective quality. Coding algorithms that take into account the human visual model are known as psychovisual coders [3B], [4B]. Hall [4B] has introduced a mathematical model for the human visual system (HVS) which consists of a low pass filter, a log function and a high pass filter.

In the next section transform coding is introduced which takes into account some aspects of both statistical and psychovisual coding.

6.3 Transform Image Coding

Transform source coder can be modelled as a sequence of three operations, a transformation, quantizer and coder, as illustrated in Fig. 6.3.1.

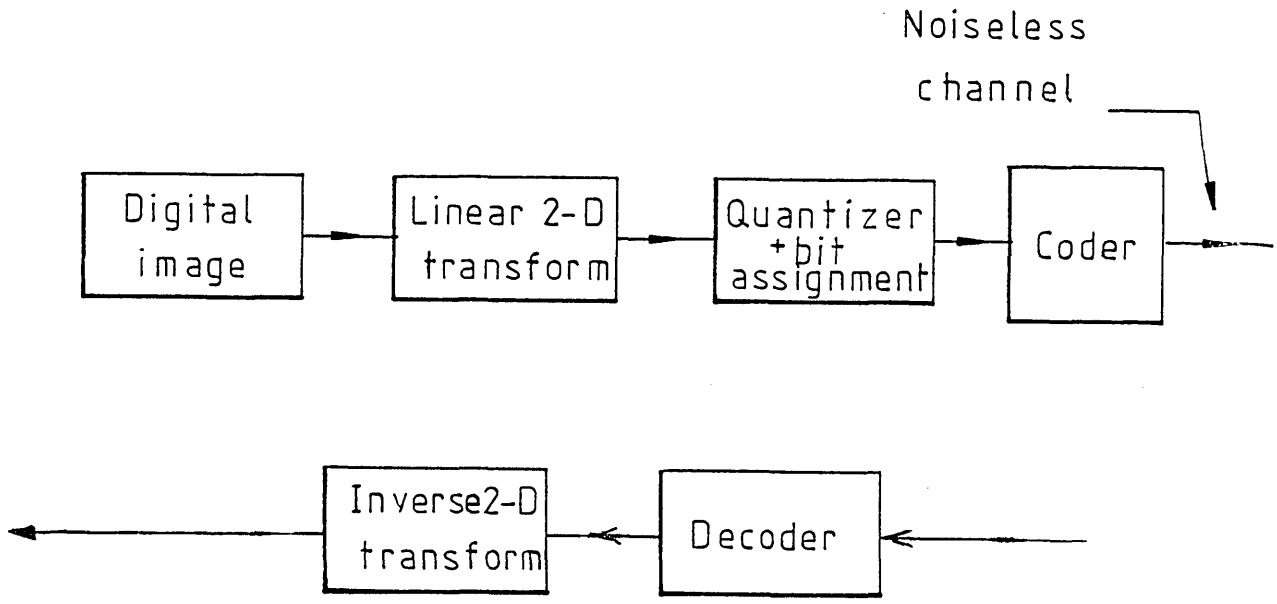


Fig. 6.3.1: A simple transform coder

The purpose of the two-dimensional linear transformation is to transform the set of statistically dependent pixels into a set of statistically independent variables. Such a linear transformation is a highly statistical transformation which cannot be found exactly because of the complex structure of images. The closest linear transformations that produce independent coefficients are the ones that produce uncorrelated coefficients. The resulting coefficients are uncorrelated but not necessarily statistically independent. However, for Gaussian image data, decorrelation ensures statistical independence. Several sub-optimal linear two-dimensional transformations have been employed in [5B], [6B], [7B]. Transforms like the discrete Karhunen-Loeve (KLT) [7B], which is based upon the statistical properties of an image, is found to uncorrelate image components quite well. Unfortunately such a statistical transformation does not have a fast implementation algorithm, so several deterministic fast transformations such as Fourier, Cosine, Sine, Hadamard and slant were investigated by several authors [5B], [6B], [8B], and [9B].

Ahmad and Rao [10B] investigated the performance of the discrete cosine transform with respect to the discrete KLT. They found that the cosine transform base functions are nearly equivalent to those of the discrete KLT. So, the discrete cosine transform is chosen throughout this chapter as our linear transformation which is implemented by the algorithm given in Chapter Five.

The quantizer is a mapping from the continuous variable domain of transform coefficients into the domain of

integers. These integers become the code-words that are transmitted through the channel. The actual coefficient quantization is performed in two steps: (a) the coefficient is normalized by its estimated variance and (b) the normalized variable is processed by the optimum quantizer based on the modelled probability density function of unit variance. The number of bits for a quantized coefficient is determined by relating the assumed prequantized variance to distortion [12B]. The optimum quantizer minimizes the mean square error between the original and quantized coefficients. The algorithm for designing a K-level quantizer was developed by Max [12B] for a Gaussian signal and subsequently for other probability distributions [13B].

Finally, the entropy coder is a reversible process which assigns a unique code-word to each possible input value. This entropy encoder exploits the redundancy that exists in the non-uniform probability distribution of the quantized data. In our algorithms a Huffman coder [14B] is used which assigns code-words of unequal lengths to the quantizer output levels.

6.3.1 Block-Transform Coding

As a consequence of the computational complexity involved in two-dimensional transform coding an image of size (NxN) is sub-divided into a set of (MxM) blocks, where each block is coded as a unit, independent of all other blocks. Block size is an important practical consideration. The argument is often made that no benefit is obtained by choosing subimages larger than the image correlation distance, assuming it is known. For most images the pixel correlation

distance is likely not to exceed 8 or 20 pixels [15B]. This is not a valid assumption since an image has a non-homogeneous structure, thus pixel correlation distance is different for different parts of the image. However, for practical reasons, it is advantageous not to exceed 16x16 or 32x32 block size. To obtain maximum decorrelation, increasing block size is beneficial. Conversely, to adapt to the local image structure, a smaller block size is preferred. In addition, the overheads associated with an adaptive transform coding algorithm are likely to become more important with decreasing transform block size.

In block-transform coding there are two branches:

(a) Non-adaptive two-dimensional block transform coding in which each block is coded using the same encoder.

(b) Adaptive two-dimensional block transform coding where each subimage is coded using the best encoder for that subimage content.

In the following sections both adaptive and non-adaptive block-transform coding systems will be discussed and simulated.

6.3.1.1 Non-Adaptive Block Transform Coding with Experimental Results

The test images in Figs. 6.3.2 and 6.3.3 of resolution 128x128 quantized to 8 bits were partitioned into blocks of size (8x8), then each block f(i,j) was cosine

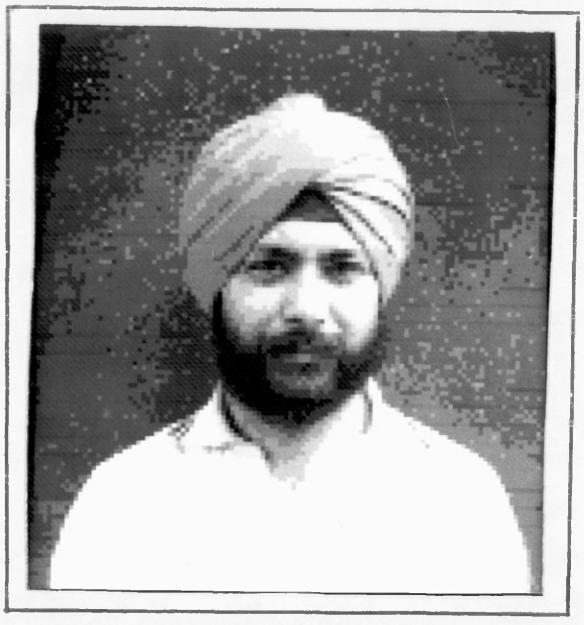


Fig. 6.3.2: Digital test image of size (128 x 128) with amplitude resolution of 8 bits, shown with only 16 levels.

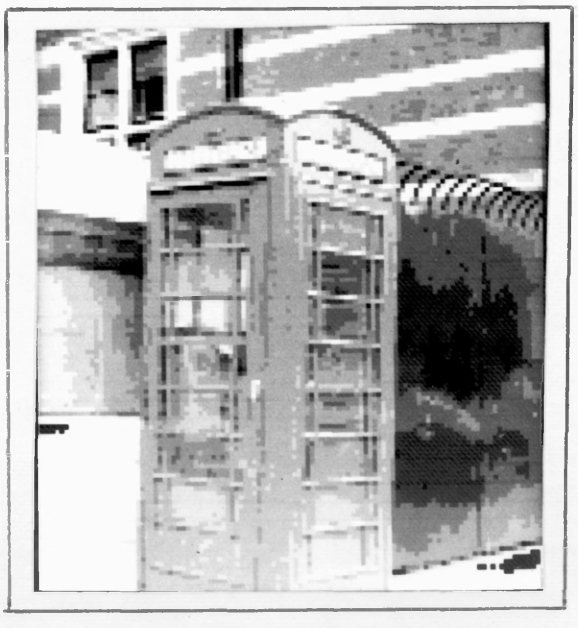


Fig. 6.3.3: Digital test image of a telephone box of size (128 x 128) with amplitude resolution of 8 bits, shown with only 16 levels.

transformed. The forward and inverse cosine transforms are given by expression (6.3.1):

$$F_i(u,v) = \sum_{i=0}^{N_1-1} \sum_{j=0}^{N_2-1} f(i,j)\, A_R(i,j)$$

$$\text{for } u = 0,1,\ldots, N_1-1 \qquad v = 0,1,\ldots, N_2-1 \qquad (6.3.1)$$

$$f_i(i,j) = \sum_{u=0}^{N_1-1} \sum_{v=0}^{N_2-1} F(u,v)\, B_R(u,v)$$

$$\text{for } i = 0,1,\ldots, N_1-1 \qquad j = 0,1,\ldots, N_2-2$$

where $A_R(i,j)$ and $B_R(u,v)$ are the forward and inverse transform kernels.

A typical cosine transformed block is shown in Fig. 6.3.4. The result shows that the magnitude of the cosine transformed coefficients decrease with increase in frequency, in a zig-zag manner. Next, the transform coefficients are operated on by a coefficient selector that decides which coefficients are to be quantized and then transmitted.

There are two methods for selecting coefficients;

(a) Threshold coding system [16B] where each coefficient whose magnitude is greater than a given threshold level is quantized with a fixed number of levels and is then coded. In this system the overhead information which gives the position of the selected coefficients is run-length coded.

4	4	4	4	4	4	4	0
4	5	5	5	5	5	4	0
4	5	6	6	6	5	4	0
4	5	6	7	6	5	4	0
4	5	6	6	6	5	4	0
4	5	5	5	5	5	4	0
4	4	4	4	4	4	4	0
4	4	4	4	4	4	4	0

(a)

33.5	-5.9	-1.1	2.8	-2.3	3.5	-1.1	-5.6
-4.5	0.8	0.3	-0.3	0.3	-0.4	0.1	0.7
-4.0	0.8	1.7	0.3	0.0	-0.1	0.1	0.3
6.9	-1.2	0.9	1.1	-0.7	0.9	-0.3	-1.4
-7.2	1.2	-0.5	-1.0	0.7	-0.8	0.3	1.4
8.2	-1.4	0.7	0.8	-0.4	1.8	-0.3	-1.7
-5.3	0.9	-0.3	-0.6	0.4	-0.7	0.2	0.9
-2.4	0.4	-0.1	-0.3	0.2	-0.4	0.0	0.8

(b)

Fig. 6.3.4: (a) The original test block
(b) The cosine transform of the test block

(b) Zonal-coding system, where a bit assignment matrix $N_B(u,v)$ is formed which gives the quantization levels with which the transformed coefficients are quantized. The coefficients with larger variances generally contribute significantly more to the reconstructed image than the coefficients with the smaller variances; the total distortion due to quantizing coefficients may be lessened by allocating more quantization levels or bits to the coefficients with the larger variances and proportionally fewer to the coefficients with the smaller variances.

For a source with Gaussian probability distribution and mean-square distortion criterion, Davisson [17B] has shown that the bit assignment matrix $N_B(u,v)$, based upon a rate-distortion theory, is given by

$$N_B(u,v) = \frac{1}{2} \log_2 [\sigma^2(u,v)] - \log_2 [D] \qquad (6.3.2)$$

where $\sigma^2(u,v)$ is the expected variance of the transformed coefficients given by the equation below

$$\sigma^2(u,v) = \frac{1}{QP} \sum_{i=0}^{Q} \sum_{j=0}^{P} [F_{i,j}(u,v)]^2 - [m(u,v)]^2 \qquad (6.3.3)$$

for $u,v, \varepsilon (0,1,2,\ldots, N)$

where P and Q give the number of the blocks in each direction, and m(u,v) is the mean of the transformed coefficients, N is the number of rows and columns in each block.

D (distortion) is a parameter which controls the tradeoff between the rate of the encoder and the quality of the reconstructed image. For example, decreasing D increases the bit rates with a corresponding decrease in mean-square error (MSE).

Figs. 6.3.5(a) and 6.3.5(b) denote the expected variance and the bit assignment matrix for a given rate R and distortion D, of the test image in Fig. 6.3.2. The corresponding matrices for test image in Fig. 6.3.3 are given in Figs. 6.3.6(a) and 6.3.6(b). Finally several bit assignment matrices are given in Fig. 6.3.6(c), with their corresponding bit rate R and distortion D.

Fig. 6.3.7 shows several zonal-coded images using the bit assignment matrices given in Fig. 6.3.6(c) , a MAX quantizer [12B].

In our tests normalized mean-square error (NMSE) was used as a criteria for image quality, given by

$$\text{NMSE} = \frac{1}{N_1 N_2} \sum_{i=0}^{N_1-1} \sum_{j=0}^{N_2-1} \left[\frac{f(i,j) - \hat{f}(i,j)}{f(i,j)} \right]^2 \quad (6.3.4)$$

where $f(i,j)$ and $\hat{f}(i,j)$ are the original and the processed images of size $(N_1 x N_2)$ respectively.

Hall [15B] reported a better image quality measure by calculating NMSE in Human Visual Domain (HVDM) rather than calculating it in spatial domain.

From the results in Fig. 6.3.7 it is seen that the NMSE end-subjective quality of the processed images improve as the bit rate is increased.

221	40	20	13	6	4	5	4
22	8	4	3	2	1	1	1
9	4	3	2	2	1	1	1
5	3	2	2	1	1	1	1
4	2	2	2	1	1	1	1
3	1	1	1	1	1	1	1
3	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1

Fig. 6.3.5a: The expected variance for blocks of 8x8 of the image in Fig. 6.3.2

8	5	4	4	3	2	3	2
5	3	2	2	1	1	0	0
3	2	2	1	1	1	0	0
2	2	1	1	1	0	0	0
2	1	1	1	1	0	0	0
2	1	1	1	1	0	0	0
2	1	1	1	1	0	0	0
2	1	1	0	0	0	0	0

Fig. 6.3.5b: The bit assignment matrix for block of 8x8 at bit rate of R = 1.3 with distortion D = 0.5

126	42	25	13	8	6	8	6
19	6	3	2	1	1	1	1
12	4	2	1	1	1	1	1
9	3	2	1	2	1	1	1
6	3	2	2	1	1	1	1
5	3	2	1	1	1	1	1
4	3	2	1	1	1	1	1
4	3	2	1	1	1	1	1

Fig. 6.3.6a: The expected variance for blocks of 8x8 of the image in Fig. 6.3.3

7	5	5	4	3	3	3	3
4	3	2	1	1	1	1	0
4	2	1	1	1	1	0	0
3	2	1	1	1	0	0	0
3	2	1	1	0	0	0	0
3	2	1	1	1	0	0	0
2	2	1	1	1	0	0	0
2	2	1	1	0	0	0	0

Fig. 6.3.6b: The bit assignment matrix for block of 8x8 at bit rate of R = 1.42 with distortion D = 0.5

5	4	3	2	2	2	2	2
3	2	1	1	1	1	1	1
2	1	1	1	1	1	0	0
1	1	1	1	1	0	0	0
1	1	1	0	0	0	0	0
1	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

(a) R = 0.94, D = 0.9

8	5	4	4	3	2	3	2
5	3	2	2	1	1	0	0
3	2	2	1	1	1	0	0
2	2	1	1	1	0	1	0
2	1	1	1	0	1	0	0
2	1	1	1	0	0	0	0
2	1	1	1	1	0	0	0
2	1	1	0	0	0	0	0

(b) R = 1.5, D = 0.5

8	6	5	4	3	3	3	3
5	4	3	2	2	2	1	1
4	3	2	2	2	1	1	1
3	2	2	2	1	1	1	1
3	2	2	2	1	1	1	1
3	2	1	1	1	1	1	1
2	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1

(c) R = 1.97, D = 0.2

9	6	5	5	4	3	4	3
6	4	3	3	2	2	2	1
4	3	3	3	2	2	1	1
4	3	3	3	2	2	2	1
3	2	2	2	2	2	2	1
3	2	2	2	2	1	1	1
3	2	2	2	2	1	1	1
3	2	2	2	1	1	1	1

(d) R = 2.47, D = 0.1

Fig. 6.3.6c(1): Bit assignment matrices for the picture in Fig. 6.3.2 at bit rate R with distortion D

7	5	4	3	3	2	3	2
4	2	1	1	1	0	0	0
3	2	1	1	0	0	0	0
3	1	1	0	1	0	0	0
2	1	1	1	0	0	0	0
2	1	1	0	0	0	0	0
2	1	1	0	0	0	0	0
2	1	1	0	0	0	0	0

(a) R = 1.1, D = 0.9

7	5	5	4	3	3	3	3
4	3	2	1	1	1	1	0
4	2	1	1	1	1	0	0
3	2	1	1	1	0	0	0
3	2	1	1	0	0	0	0
3	2	1	1	1	0	0	0
2	2	1	1	0	0	0	0
2	2	1	1	0	0	0	0

(b) R = 1.4, D = 0.5

8	6	5	4	4	3	4	3
5	3	2	2	2	2	1	1
4	3	2	2	1	1	1	1
4	3	2	2	2	1	1	0
3	2	2	2	1	1	0	1
3	2	2	1	1	1	1	1
3	2	2	1	1	0	0	1
3	3	2	2	1	0	0	1

(c) R = 2, D = 0.2

8	7	6	5	4	4	4	4
5	4	3	3	2	2	2	2
5	3	3	2	2	2	1	1
4	3	2	2	2	1	1	1
4	3	2	2	1	1	1	1
4	3	2	2	2	1	1	1
3	3	2	2	2	1	1	1
3	3	3	2	1	1	1	1

(d) R = 2.5, D = 0.1

Fig. 6.3.6c(2): Bit assignment matrices for the picture in Fig. 6.3.3 at bit rate R with distortion D

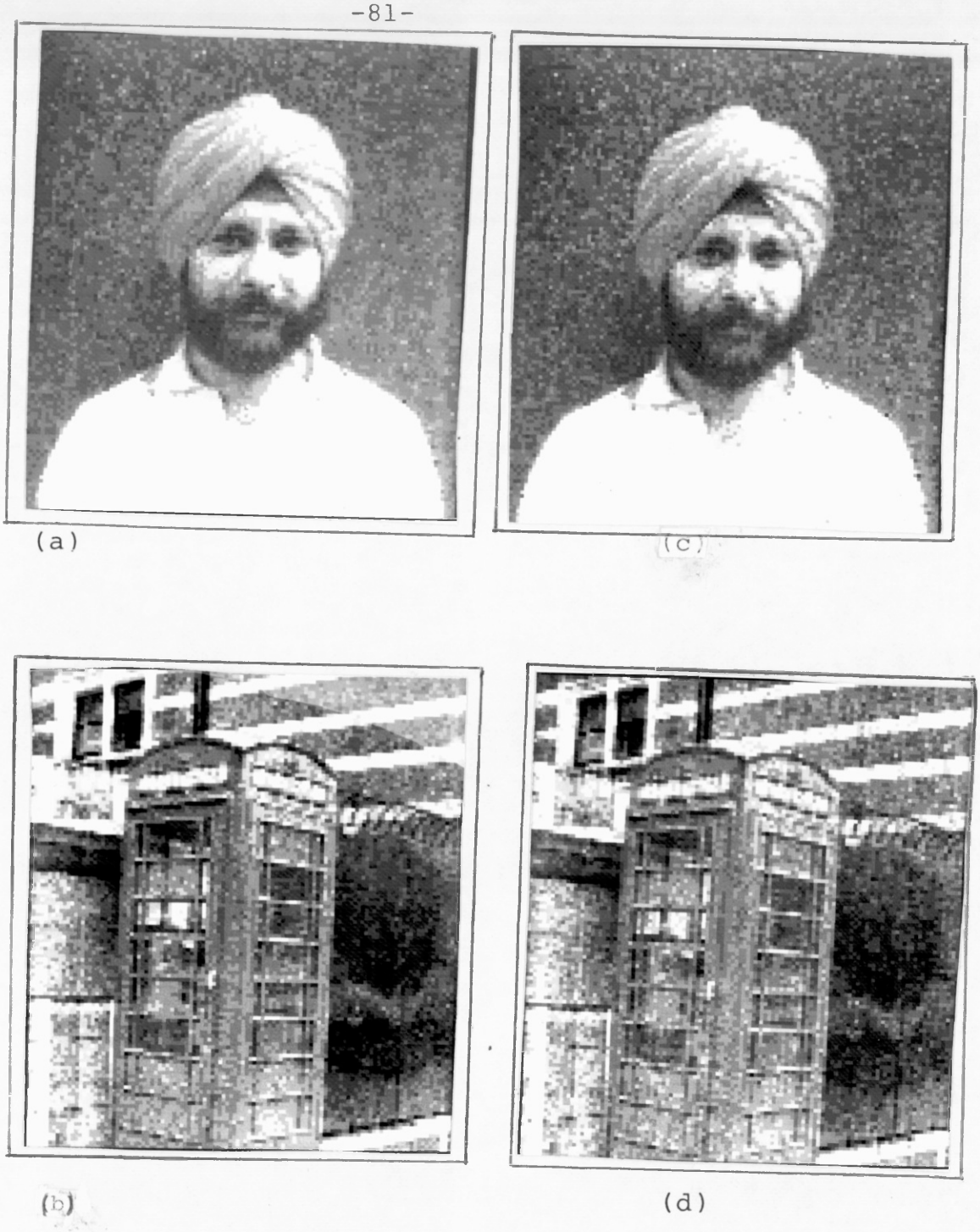


Fig. 6.3.7: (a) The coded image of Fig. 6.3.2 at bit rate of R = 2.47 bits/pixel, NMSE = 3.4 x 10^{-3} with distortion parameter DX = 0.1.

(b) The coded image of Fig. 6.3.3, R = 2.52, NMSE = 1.13 x 10^{-2} with DX = 0.1.

(c) The coded image of Fig. 6.3.2, R = 1.9, NMSE = 5.0 x 10^{-3} with DX = 0.2.

(d) The coded image of Fig. 6.3.3, R = 2.04, NMSE = 1.33 x 10^{-2} with DX = 0.2.

6.3.2 An Adaptive Coding Technique which takes into account the Inter-correlation between the Blocks

In the previous section an image of size NxN was divided into blocks of size MxM. Each block was assumed to be independent of the neighbouring blocks and coded independently. This assumption is not strictly valid because the pixels on the border of a block are correlated with respect to the pixels on the borders of the adjacent blocks. In this section the inter-block correlation is investigated and a method is developed to decorrelate the adjacent blocks. First the effect of overlapping the blocks is considered and its computational complexity is compared with that of the non-overlapping method. Then an adaptive overlapping technique is developed.

6.3.2.1 Experimental Results for Overlapped-block Technique

First the effect of overlapping blocks without quantization was considered for coding a picture of hand-written English; the letters 'shu' of size 96x96 digitized to 5 bits as shown in Fig. 6.3.8. The image was then divided into blocks of size 16x16 and each block was cosine transformed as given by expression (6.3.1). The effect of discarding a number of low magnitude coefficients and replacing them with zeroes and reconstructing was then considered.

Fig. 6.3.9 shows several images reconstructed, only retaining the first 16, 64, 144 of the transformed coefficients. As seen from the images, discarding the high frequency coefficients has the effect of low-pass filtering.

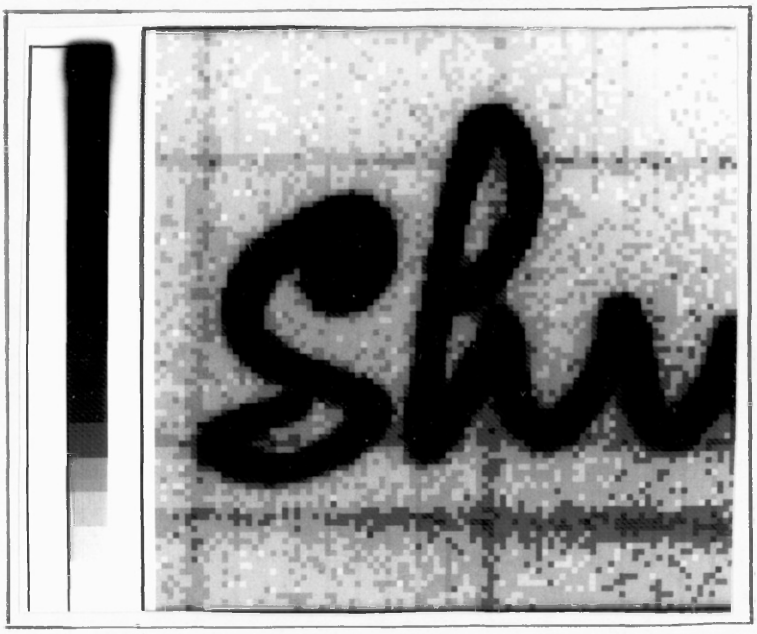


Fig. 6.3.8: The original Shu picture of size 96x96 with amplitude resolution of 5 bits/pixel

(a)

(b)

(c)

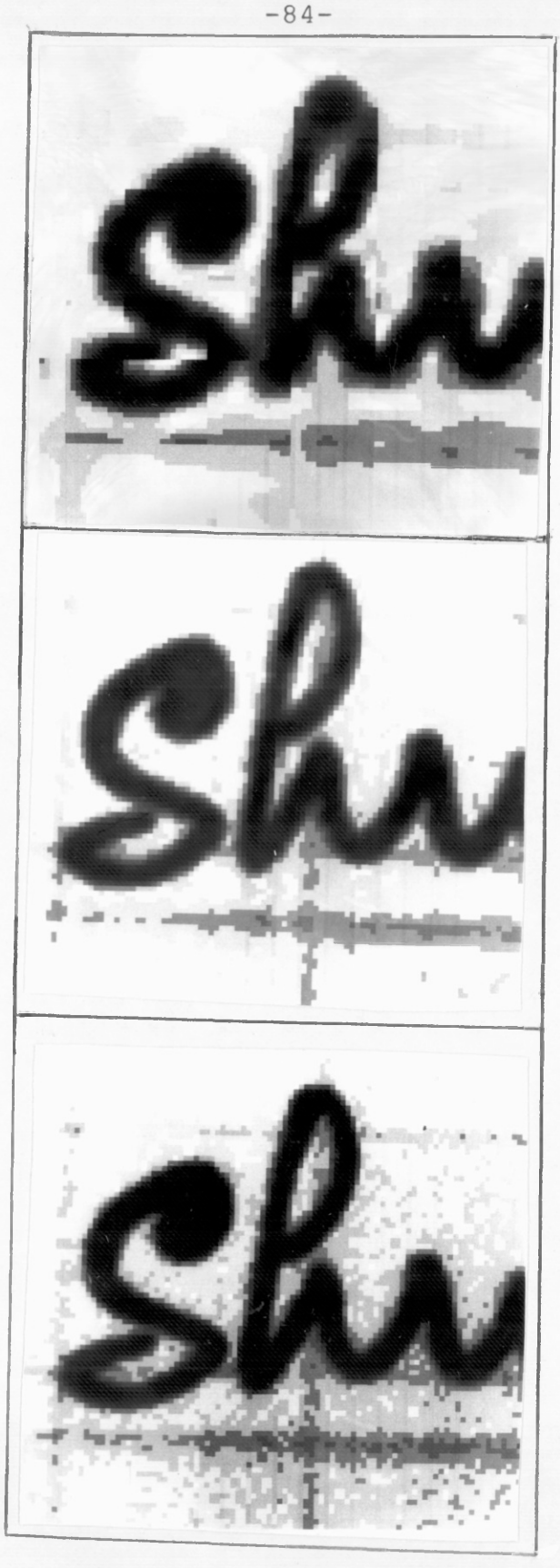


Fig. 6.3.9: (a) Reconstructed image by retaining only the first 16 transform coefficients.
(b) Reconstructed image by retaining only the first 64 transform coefficients.
(c) Reconstructed image by retaining only the first 144 transform coefficients.

In order to investigate the effect of overlapping, the test image was first extended to (112x112) by padding zeroes to evergy direction of the image; each block of 16x16 was then enlarged to an overlapped-block size of 32x32. This was achieved by overlapping each block with its eight neighbouring blocks by 8 pixels.

Fig. 6.3.10 represents the processed images obtained by overlapping the blocks eight pixels in each direction and retaining only the first 16, 64, 144 coefficients.

Comparing the results it is seen that by overlapping the blocks by a certain number of pixels, subjectively better coded images could be obtained.

Next the effect of quantizing the overlapped-blocks was considered. The expected variance and bit assignment matrices for overlapped-blocks are shown in Fig. 6.3.11. Fig. 6.3.12 shows the coded images for overlapped-blocks at several transmission rates R with their corresponding NMSE.

Comparison of the results in Figs. 6.3.7 and 6.3.12 show that by overlapping the blocks we can transmit the image at a lower bit rate than for that of the non-overlapped blocks, with the same distortion. The reason is that by overlapping the inter-block correlation is taken into account and each block is made more independent of the others than before. Also the larger the block size the more stationary will the data be because the larger the block, the more likely will areas of little variation be included with areas of great variation. The number of multiplications needed for cosine transforming a non-overlapped block and an overlapped block of size $M_1 x M_2$ and $2M_1 x 2M_2$ by the proposed technique in Chapter 4 are given by expressions

(a)

(b)

(c)

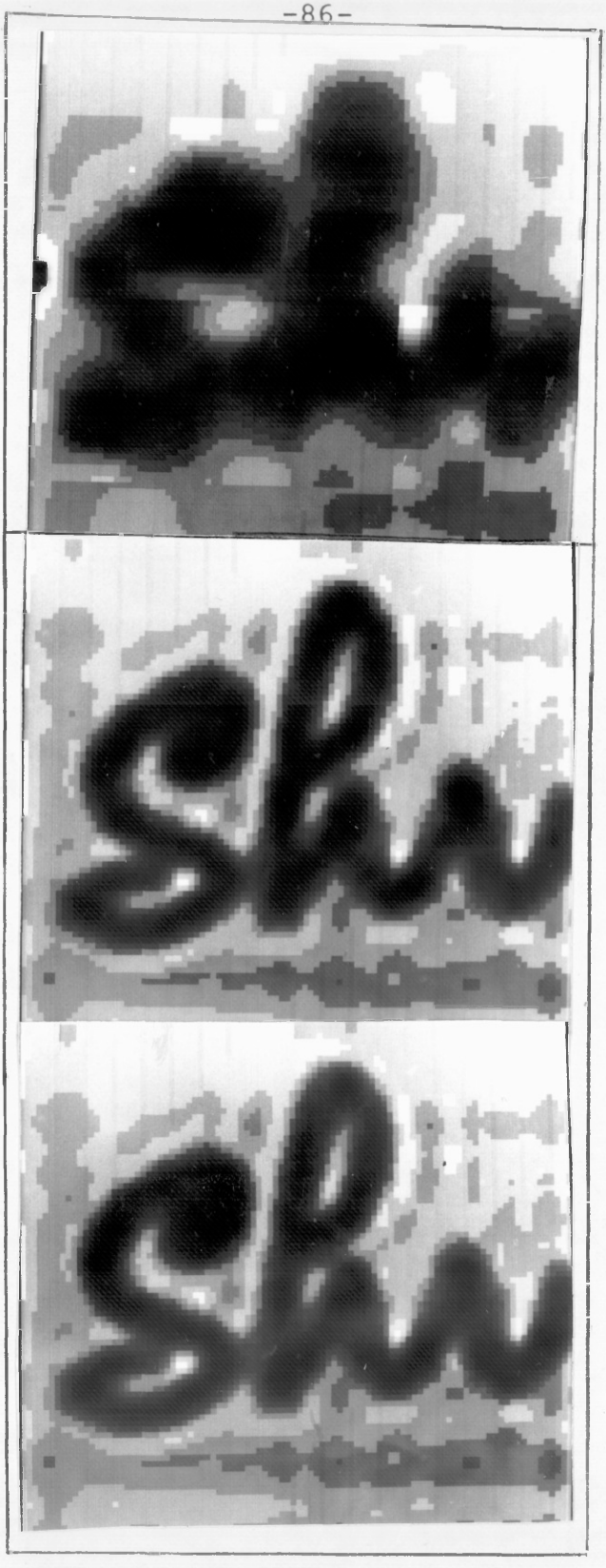


Fig. 6.3.10: (a) Reconstructed image by retaining only the first 16 transform coefficients of the overlapped block.
(b) Reconstructed image by retaining only the first 64 transformed coefficients.
(c) Reconstructed image by retaining only the first 144 transform coefficients.

207	55	17	11	11	6	5	6	3	3	3	2	3	2	2	2
59	11	6	4	3	2	2	1	1	1	1	1	1	1	1	1
37	7	4	3	2	2	1	1	1	1	1	1	1	1	1	1
18	4	3	3	2	2	1	1	1	1	1	1	1	1	1	1
3	2	2	2	1	1	1	1	1	1	1	1	1	1	1	1
10	2	2	1	1	1	1	1	1	1	1	1	1	1	1	1
12	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1
8	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
6	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
8	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
5	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
6	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0
5	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0

Fig. 6.3.11a: The vqriance matrix for overlapped block of size (16,16) of the image in Fig. 6.3.2

7	5	4	3	3	2	2	2	1	2	2	1	1	1	1	1
5	6	2	2	1	1	1	0	0	0	0	0	0	0	0	0
5	2	2	1	1	1	0	0	0	0	0	0	0	0	0	0
4	2	1	1	1	1	0	0	0	0	0	0	0	0	0	0
4	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
3	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
3	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fig. 6.3.11b: Bit assignment matrix for overlapped block of size (16,16) of the image in Fig. 6.3.2 at bit rate of R = 0.4 with distortion D = 0.9

118	52	20	12	13	7	9	5	5	5	3	4	3	3	3
39	7	4	3	2	2	1	1	1	1	1	1	1	1	1
30	5	2	2	1	1	1	1	1	1	1	1	1	1	1
16	4	2	1	1	1	1	1	1	1	1	1	1	1	1
6	3	2	1	1	1	1	1	1	1	1	0	0	1	1
10	3	2	1	1	1	1	1	1	1	1	0	0	1	1
10	3	2	1	1	1	1	1	1	1	1	0	0	1	1
7	2	1	1	1	1	1	1	1	1	0	0	0	1	0
3	2	2	1	1	1	1	1	1	1	0	0	0	0	1
5	2	1	1	1	1	1	1	1	1	0	0	0	0	1
6	2	1	1	1	1	1	1	1	1	0	0	0	0	1
4	1	1	1	1	1	1	1	0	0	0	0	0	0	1
2	1	1	1	1	1	1	1	0	0	0	0	0	0	0
4	1	1	1	1	1	1	1	0	0	0	0	0	0	1
5	1	1	1	1	1	1	1	0	0	0	0	0	0	0
4	1	1	1	1	1	1	0	0	0	0	0	0	0	0

Fig. 6.3.11c: The variance matrix for overlapped block of size (16x16) of the image in Fig. 6.3.3

6	5	4	3	3	2	3	3	2	2	2	1	2	1	1	1
5	3	2	1	1	1	1	0	0	0	0	0	0	0	0	0
4	2	1	1	0	0	0	0	0	0	0	0	0	0	0	0
4	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0
2	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
3	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0
3	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
2	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fig. 6.3.11d: The bit assignment matrix for overlapped block of size (16x16) of the image in Fig. 6.3.3 at bit rate for R = 0.5 with distortion D = 0.9

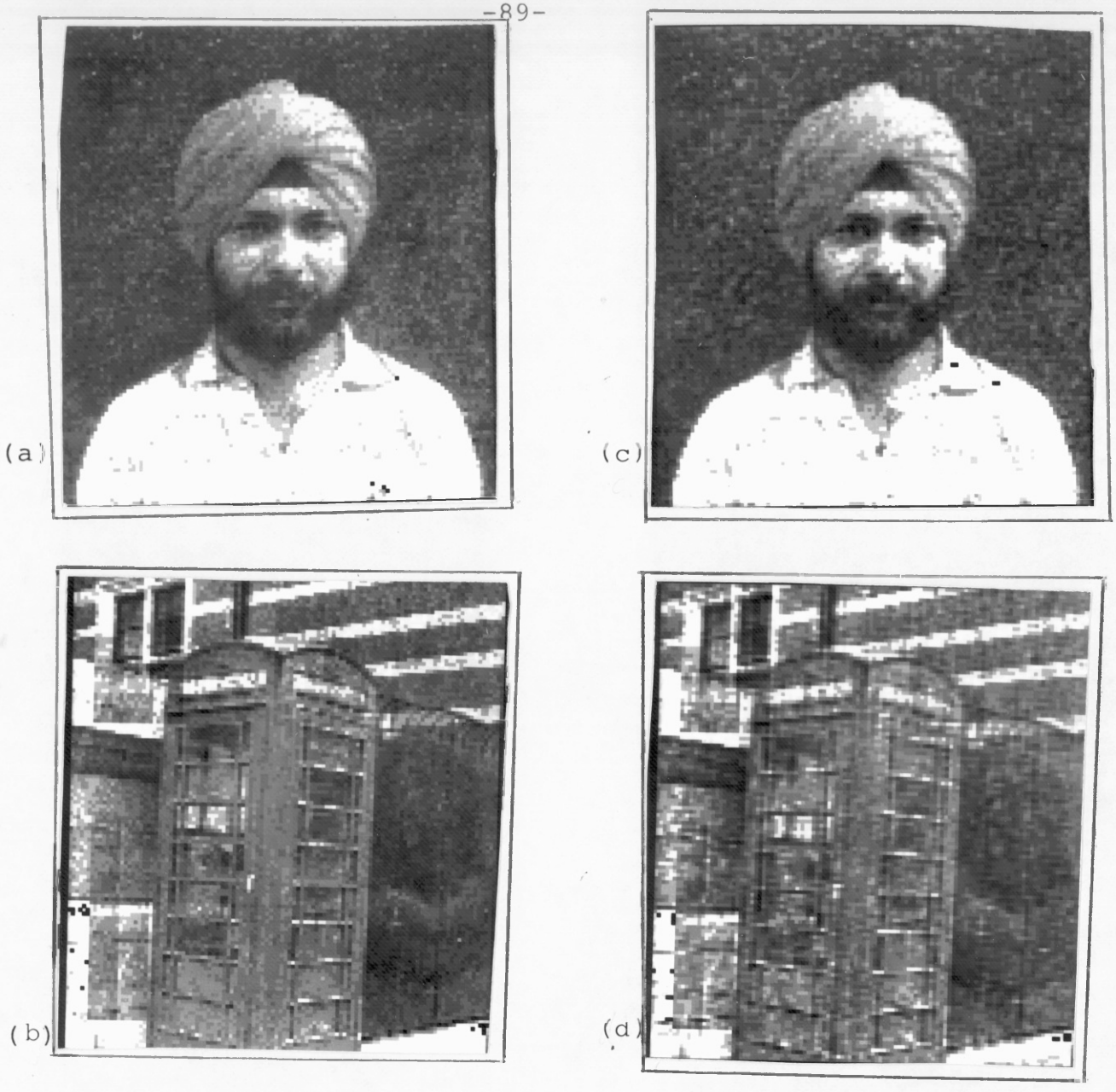


Fig. 6.3.12: (a) The coded image of Fig. 6.3.2 at bit rate of R = 1.9 bits/pixel, NMSE = 5.1×10^{-3} with distortion parameter DX = 0.07.

(b) The coded image of Fig. 6.3.3, R = 2.25, NMSE = 1.02×10^{-2}, DX = 0.03.

(c) The coded image of Fig. 6.3.2, R = 1.37, NMSE = 6.09×10^{-3}, DX = 0.1.

(d) The coded image of Fig. 6.3.3, R = 1.0, NMSE = 1.54×10^{-2}, DX = 0.2.

$$MU(M_1M_2) = 2M_1M_2 \quad (2 + \log_2 M_1)$$

$$MU(4M_1M_2) = 8M_1M_2 \quad (3 + \log_2 M_1)$$

Thus by overlapping the computational complexity is increased four-fold.

6.3.2.2 A New Adaptive Overlapped-block Technique with Computer Simulations [64B]

In order to reduce the computational cost of overlapped-block coding technique, an adaptive overlapping method is implemented. The basic idea is to find the dependency of each block on its four neighbouring blocks, and if it is highly dependent on its neighbouring blocks it is enlarged by overlapping it with its neighbouring blocks. In this way an overhead information matrix is formed with elements of 1 and 0, where one and zero denote whether the block is overlapped or not, respectively.

The correlation or the dependency between the blocks is based upon the conditional entropy H(X/Y) or the mutual entropy I(X,Y) between two neighbouring blocks X and Y.

Consider a block X with M possible levels S_i each with probability of occurrence P_i, then the entropy or the average information conveyed by block X,if successive pixels are independent, is given by [17B]:

$$H(X) = - \sum_{i=1}^{M} P_i \log_2 P_i$$

where $\log_2 P_i$ is the amount of self-information contained in each level. The maximum possible entropy occurs when the levels are equally likely

$$H_{max} = - \sum_{i=1}^{M} \frac{1}{M} \log_2 \frac{1}{M} = \log_2 M$$

If we consider two neighbouring blocks X with gray levels $\{x_i\}$ and Y with gray levels $\{y_i\}$, then the average information that may be obtained from viewing one block element, given that we have observed another element, is given by average conditional entropy of the element X with respect to Y as defined by

$$Y(X/Y) = - \sum_{i=1}^{n} \sum_{j=1}^{n} P(y_j) \, P(\frac{x_i}{y_j}) \log_2 P(\frac{x_i}{y_j})$$

and the average joint information or joint entropy between two blocks X and Y is given by

$$H(X,Y) = - \sum_{i=1}^{n} \sum_{j=1}^{m} P(x_i,y_j) \log_2 P(x_i,y_j) \qquad (6.3.5)$$

This joint information is always less than or equal to the sum of the entropies of the two block elements and equal only when the two gray levels are statistically independent as given by the expression below

$$\begin{aligned} H(X,Y) &= H(X) + H(Y) - I(X,Y) \\ &= H(X) + H(Y/X) \end{aligned} \qquad (6.3.6)$$

where I(X,Y) is known as the mutual information given by $I(X,Y) = H(X) - H(X/Y) = H(Y) - H(Y/X)$ the mutual information I(X,Y) between two neighbouring blocks X,Y may be interpreted as the information transmitted over a communication channel, since H(X) is the information at the input of the channel and equivocation H(X/Y) is the information about the input X given that the transmitted block Y is known. Now if block Y is totally correlated with block X, then H(X/Y) = 0 and I(X,Y) = H(X). However if block Y is independent of X, then H(X/Y) = H(X) and I(X/Y) = 0. Thus in order to find the dependency between the blocks the conditional entropies between the neighbouring blocks have to be calculated.

The proposed algorithm will find the first-order conditional entropies between each block and its four neighbouring blocks, as shown in Fig. 6.3.13. Then the

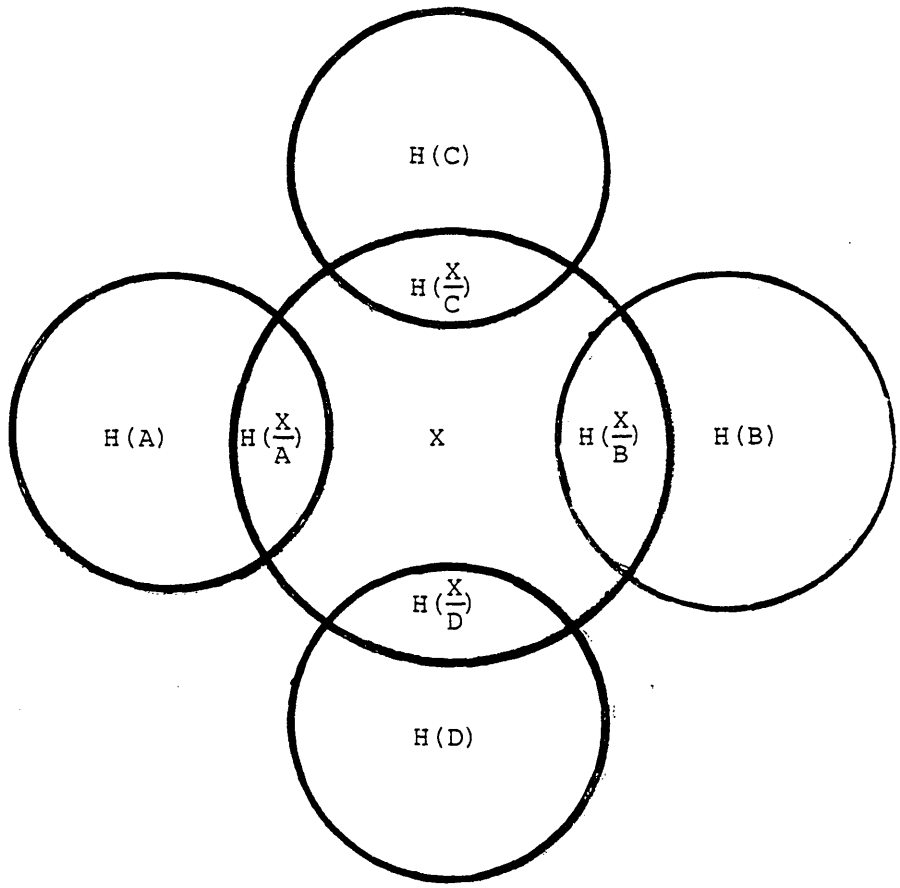


Fig. 6.3.13: Conditional entropies between neighbouring blocks

summation K of the four conditional entropies is obtained

$$K = H(X/A) + H(X/B) + H(X/C) + H(X/D)$$

This value K will give a measure of correlation between block X and its four neighbouring blocks. Only the blocks with a K value below a threshold are overlapped.

Fig. 6.3.14 shows the coded image by the above technique, where overlapping is chosen to be only four pixels in every direction. Since not all the blocks are overlapped the computational complexity is reduced compared with that of non-adaptive overlapped-block technique, with marginally the same subjective quality. Also since the entropy of each block is a measure of activity or variability in that block, the above technique could be made adaptive by classifying the blocks into disjoint classes with respect to their activity and then coding at different bit rates.

6.4 A New Zonal-coding Technique using a Vector Quantization Algorithm [64B]

To achieve a low bit rate in standard Zonal transform coding techniques, a large number of the high frequency coefficients are usually discarded. But this could result in a distorted image especially at very low bit rates. Here these high frequency coefficients are coded by a vector quantization algorithm (discussed in detail in chapters eight and nine), which exploits the dependency between these coefficients. Here the vector quantization algorithm will arrange the high frequency coefficients into vector patterns of dimension four, and each vector pattern is compared with

a code-book of standard vector templates of the same dimension, and is represented by its nearest (in a mean square sense) matching vector-template and transmitted. At the receiver these high frequency coefficients are reconstructed using the corresponding vector-templates in place of the original vectors. In practice this reconstruction can be done very rapidly by using the code-words to address a Read only Memory (ROM), in which the standard vector-templates are stored. The code-book, consisting of the most probable vector-templates, is formed in an ad hoc manner, by employing a training set of several transformed images, using the algorithm in [18B]. Fig. 6.4.1 shows the coded image by the above technique at a bit rate of R = 1.1 bits per pixel and NMSE = 5.9×10^{-3}. The transform used was cosine transform and blocks of size 8 x 8 pixels were used. The low frequency coefficients are coded by using a bit assignment matrix as shown in Fig. 6.4.2.

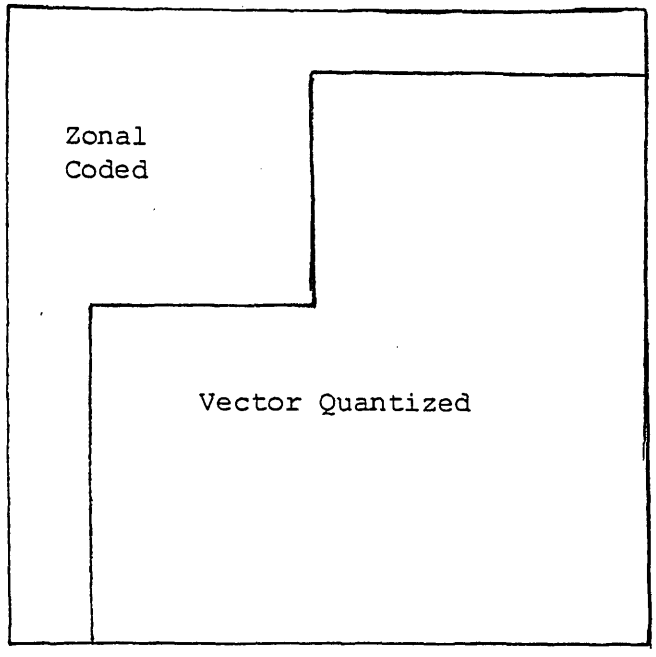


Fig. 6.4.2: The new bit assignment matrix

Fig. 6.3.14: Coded image by the adaptive overlapping technique at bit rate of R = 1.1 bits/pixel NMSE = 7.0×10^{-2}.

Fig. 6.4.1: Coded image by the proposed Zonal-coding technique at bit rate of R = 1.1 bits/pixel NMSE = 5.9×10^{-3}.

6.5 Conclusions

In this chapter image coding is introduced. In section 6.2 a simple digital transmission system is introduced and each of of its components explained. The statistical and psychovisual redundancies in the image data are also discussed. In section 6.3 a simple transform coder is introduced which consists of a two-dimensional transform, quantizer and a coder. The purpose of each component is explained. In section 6.3.1 block transform coding systems are discussed. Several techniques are discussed. Adaptive and non-adaptive block transform coders are explained. In section 6.3.1.1 conventional block transform coders are reviewed. Derivation of the bit assignment matrix and the expected variance matrix are also given in section 6.3.1.1. Finally, simulation results are given for several coded images. In section 6.3.2 the effect of overlapping blocks are considered. In 6.3.2.1 several coded images are given by overlapping blocks and compared with that of non-overlapping blocks. In section 6.3.2.2 a new adaptive overlapping block transform coder is explained. In this coder only highly correlated blocks are overlapped. The correlation measure between the blocks is based upon the conditional entropies between the blocks. In section 6.4 a new zonal coding strategy is developed where the bit assignment matrix is a combination of zonal and vector coding techniques.

The adaptive techniques which are based upon the directionality of the bit assignment matrix [19B], [20B] are not discussed in detail or simulated in this chapter. The expected variance of the transformed coefficients are evaluated for each image and transmitted. It is possible to use a recursive technique to estimate the variance matrix from the transformed coefficients.

CHAPTER SEVEN

CHAPTER SEVEN

PREDICTIVE CODING TECHNIQUES

7.1 Introduction

In this chapter, predictive coding techniques that take advantage of the picture signal statistics, such as differential pulse code modulation (DPCM) are introduced. Extended differential pulse-code modulation systems with adaptive predictors and adaptive quantizers are explained in Section 7.2. In general, the sample values of spatially neighbouring picture elements are correlated. Correlation or linear statistical dependency indicates that a linear prediction of the sample values based on sample values of neigbouring picture elements will result in prediction errors that have a smaller variance than the original sample values. One-dimensional prediction algorithms make use of the correlation of adjacent picture elements within the scan line [21B], other more complex schemes also exploit line-to-line [22B] and frame-to-frame [23B] correlation and are denoted as two-dimensional and three-dimensional predictions respectively. In Section 7.3 a new adaptive DPCM algorithm is introduced which employs Fuzzy Concepts [24B]. Only intraframe predictive coding techniques are introduced. In Section 7.4 we will give a comparison of the adaptive DPCM algorithms with transform coding techniques. The predictors used are linear since a non-linear predictor involves conditional expectation values which are difficult to implement.

7.2 Differential Pulse Code Modulation

A block diagram of a digital DPCM system is shown in Fig. 7.2.1. For every picture element x_N, the linear predictor generates a prediction value $\hat{x}_N$ which is calculated from N-1 preceding samples according to the relation

$$\hat{x}_N = \sum_{i=1}^{N-1} d_i \, x_{N-i}$$

Only preceding transmitted samples are used for prediction, so that the receiver is also able to calculate $\hat{x}_N$. The coefficients a_i are optimized to yield a prediction error $d_N = x_N - \hat{x}_N$ with minimum variance. Usually the difference signal is quantized to eight levels and coded with a 3-bit code, since the probability of occurrence of the quantized difference signal is not uniform (Fig. 7.2.2). It is possible to employ a variable-length statistical code, such as a Huffman code [14B], rather than a 3-bit constant length code, and achieve a greater coding compression.

In order to take the full advantage of DPCM systems, several authors [25B], [26B] have investigated adaptive predictors and quantizers. The aim of adaptive prediction algorithms is to reduce the prediction error and then the variance and thus to decrease the quantization error at picture contours, where invariant predictors generally produce the largest prediction error. There are several ways to make the predictor adaptive, but the main two types of adaptive predictors are, one that uses an adaptive coefficient in which the predictor coefficient is changing with respect to the incoming signal, and a more complicated so-called contour

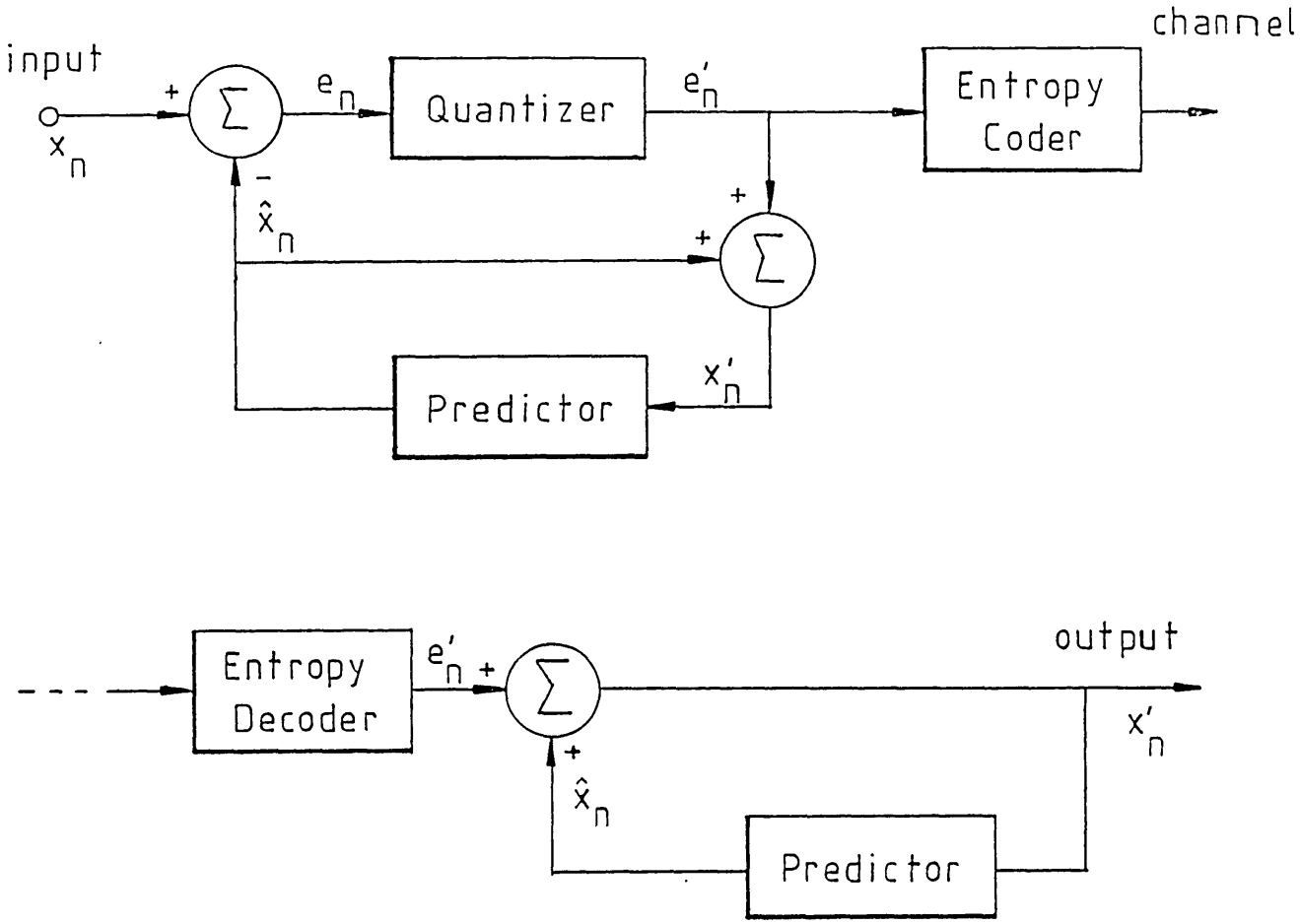


Fig. 7.2.1: A conventional DPCM system

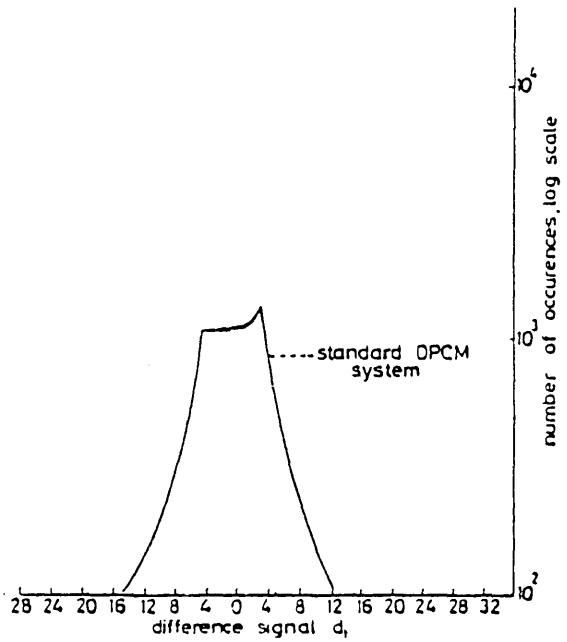


Fig. 7.2.2: Histogram of the difference signal

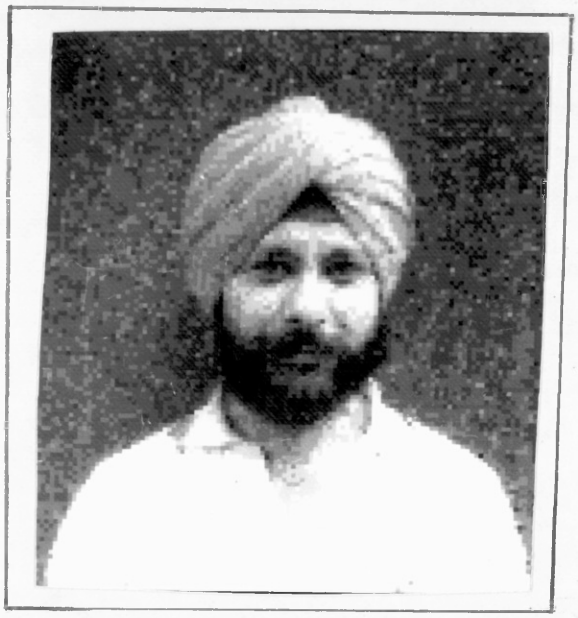
predictor which aims at selecting the neighbouring picture element that is most similar in amplitude to the actual picture element to be coded [22B].

Adaptive quantizers are used in order to eliminate the impairments (such as slope overhead, granularity noise, contouring patterns and edge busyness) caused by finite-level quantizers. Adaptivity could be based on using a non-uniform quantizer which could be designed to take into account the human visual system [27B].

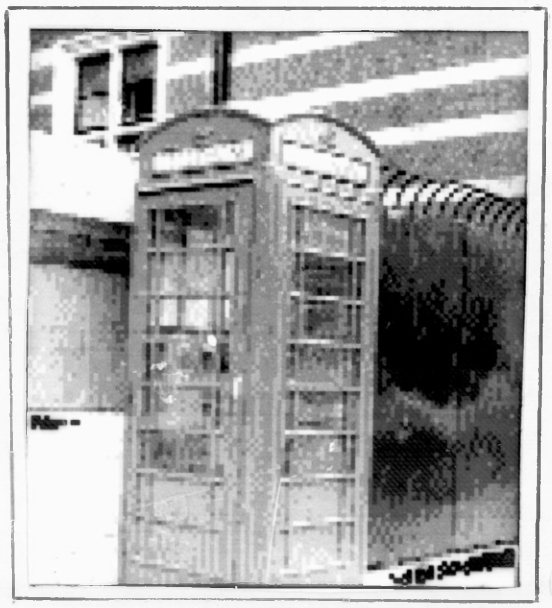
The conventional DPCM system in Fig. 7.2.1 was simulated for uniform 32 level Gaussian, Laplacian and Gamma quantizers. The DPCM signal was Huffman coded. The predictor was a fixed previous picture element predictor. Fig. 7.2.3 shows the coded images with a 32-level Laplacian quantizer. From extensive simulation it was found that the error signal probability distribution is very close to a Laplacian distribution as reported by other authors [21B].

7.3 A New Adaptive DPCM Coding [31B], [32B]

In the previous section the redundancies, such as correlation in the picture data or non-uniform probability density of the data were removed by DPCM and entropy coding techniques respectively. In this section the DPCM system of Fig. 7.2.1 is made adaptive in order to take full advantage of these two redundancies. The proposed hybrid DPCM system (HDPCM) is shown in Fig. 7.3.1. In the system we use transformation $T(d_i)$ on the difference signal d_i. This transformation will make the probability density distribution of the difference signal which is highly peaked into a much more highly peaked one, as shown in Fig. 7.3.2. Such a highly peaked density



(a)



(b)

Fig. 7.2.3: (a) DPCM coded image at bit rate of R = 3.1 bits/pixel with NMSE = 1.7×10^{-3}.
(b) DPCM coded image at bit rate of R = 3.1 bits/pixel with NMSE = 3.6×10^{-3}.

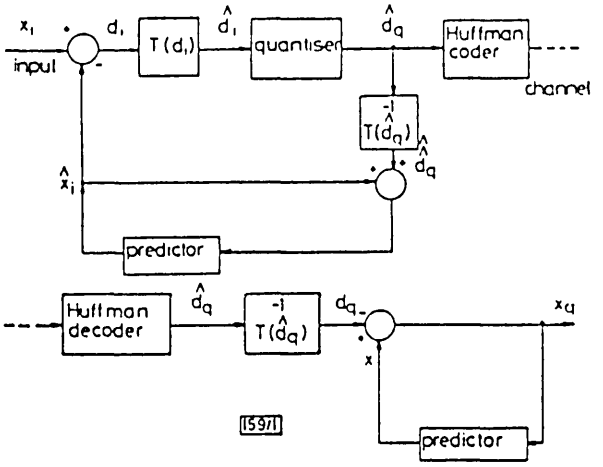


Fig. 7.3.1: Proposed hybrid system (HDPCM)

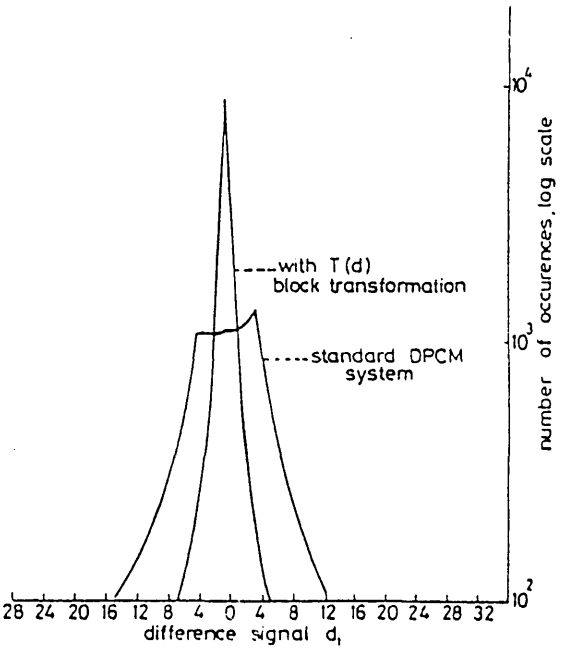


Fig. 7.3.2: Histogram of difference signal with and without transformation

distribution is very suitable for entropy coding, thus the purpose of the transformation $T(d_i)$ is to make full use of redundancy of non-uniform probability distribution of the signal. In the system simulated, a Huffman entropy coder is used, which is very suitable for such a probability distribution.

The transform $T(d_i)$ consists of three-operations [28B]:

(a) transformation from spatial d-domain to fuzzy p-domain ($0 \leq p \leq 1$) using the standard S function shown in Fig. 7.3.3

$$P_i = G(d_i) = 2(|d_i|/d_{max})^2, \quad d_i \leq d_{max/2}$$
$$= 1-2((d_{max} - d_i)/d_{max})^2, \quad d_i > d_{max/2}$$

where $d_{max/2}$ is the crossover point at which $P_i = 0.5$;

(b) enhancing the contrast in the p-domain using the fuzzy intensification operator [29B] given by,

$$\hat{P}_i = A(P_i) = 2(P_i)^2, \quad 0 \leq P_i \leq 0.5$$
$$= P_i, \quad 0.5 \quad P_i \leq 1$$

which decreases the values of P_i that are below 0.5 and leaves the rest unchanged. The degree of enhancement can further be increased by making successive use of this A-operator;

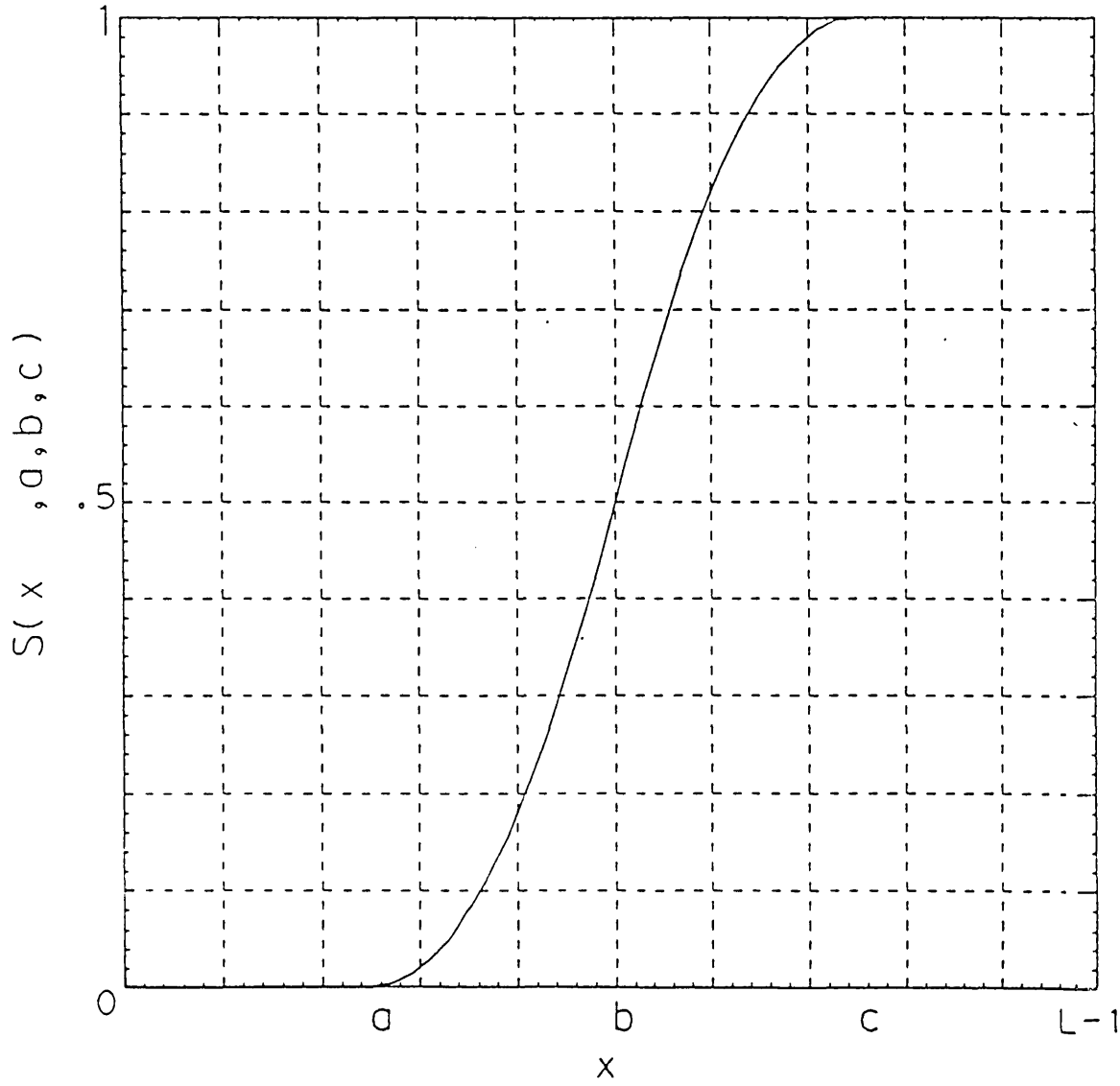


Fig. 7.3.3: Standard S-function

(c) inverse transformation $\hat{d}_i = G^{-1}(\hat{P}_i)$ is applied in order to obtain the enhanced spatial domain d_i from the intensified fuzzy p-values. The transformation $T^{-1}(\hat{d}_i)$ consists of the above operation in reverse order.

7.3.1 Implementation and Results

The designed system was simulated on a digital computer CDS 6900 for two digitized pictures of 128 x 128 quantized to 8 bits. Figs. 7.3.1.1a and b show the processed pictures of a man without and with the transformation block $T(d_i)$ coded at 2.5 bits per pixel and 3.1 bits per pixel, respectively. Fig. 7.3.1.2a and b demonstrate the coded 'telephone box' under the same processing condition. The corresponding normalized mean-square errors of the processed images are also given. From the above result it appears that the use of a $T(d_i)$ block can easily save about 0.5 bit with an insignificant degradation over the standard DPCM system.

7.4 Nth-order DPCM versus Block Transform Coding [30B]

The DPCM system that was simulated in section 7.2 used a first order linear predictor. If the order of the predictor is increased, that is a larger number of previous pixels are considered for prediction, the difference signal distribution is more highly peaked compared with that of a first order predictor. Bit rates of 1-2 bits per pixel are believed to be achieved with highly adaptive DPCM systems. This bit rate is comparable with that of transform coders discussed in chapter six. The DPCM systems are very sensitive to errors so they are

(a)

(b)

Fig. 7.3.1.1: (a) DPCM coded image R = 3.1 and NMSE = 1.7×10^{-3}.

(b) Coded image by the proposed hybrid system at bit rate of R = 2.5 bits/pixel with NMSE = 1.9×10^{-3}.

(a)

(b)

Fig. 7.3.1.2: (a) DPCM coded image R = 3.1 and NMSE = 3.6×10^{-3}.

(b) Coded image by the proposed hybrid system at bit rate of R = 2.5 bits/pixel with NMSE = 3.9×10^{-3}.

not suitable for noisy transmission mediums. However, transform coders are not very sensitive to the noise introduced during transmission. The transform coders require a larger storage memory compared to that of DPCM since transformation is performed on a large sequence of pixels.

7.5 Conclusions

Predictive coders are introduced in this chapter. Linear predictors are discussed in section 7.2. The linear predictors were only simulated, although non-linear predictors where the edge variation is incorporated are believed to be much more efficient than linear predictors. The complexity of the predictor will increase the cost of the system as well as the time required to do the prediction. Most of the DPCM systems use an adaptive quantizer. Several adaptive quantizers have been mentioned in section 7.2. Simulation results with a Gaussian, Laplacian and Gamma quantizer are given in section 7.2. It is found that the error signal probability distribution is very close to that of Laplacian probability distribution. In section 7.3 a hybrid DPCM is introduced. A non-linear function is used to transform the error signal into a signal with a highly peaked distribution. Entropy coders are employed to exploit the non-uniformity of the error signal. A Huffman coder was employed in our simulations. 0.5 bit per pixel can be saved by the proposed technique compared to conventional DPCM systems.

CHAPTER EIGHT

CHAPTER EIGHT

INTRAFRAME HYBRID IMAGE CODING

8.1 Introduction

In the previous two chapters two-dimensional transform and predictive image coding DPCM systems were introduced. These systems had comparative advantages and disadvantages. With transform image coding, higher image fidelity reconstruction can be achieved at low bit rates. Also, the transform techniques are not so sensitive to variations in image statistics and are less vulnerable to channel error effects than DPCM systems. DPCM systems have superior coding performances at high bit rates with less complex hardware implementation, and greatly reduced storage requirements.

The hybrid coding systems considered in this chapter are: hybrid transform/DPCM system [33B] and hybrid transform/vector coding [34B]. These two systems combine the attractive features of both transform and DPCM or vector quantization techniques respectively.

In sections 8.2 and 8.3 one- and two-dimensional hybrid transform/DPCM systems are discussed. Sections 8.4 and 8.5 illustrate two proposed hybrid transform/vector quantization coding systems.

8.2 One-dimensional Hybrid Transform/DPCM System

A one-dimensional hybrid system exploits the correlation of the image data in the horizontal direction by taking a one-dimensional transform of each line of the picture data $x(n_1,n_2)$, and operating on each column of the transformed data $P(n_1,m_2)$ using a bank of DPCM systems. The DPCM systems

quantize the signal in the transform domain taking advantage of the vertical correlation of the transformed data to reduce the coding error. In practice, a single adaptive DPCM code could be time-shared between columns. A simplified block diagram of the intraframe hybrid coder is presented in Fig. 8.2.1.

The general forms of the one-dimensional forward and inverse transforms along the image rows are:

$$P(n_1,m_2) = \sum_{n_2=0}^{N_2-1} x(n_1,n_2)\, A_R(n_2,m_2) \tag{8.1}$$

$$x(n_1,n_2) \sum_{m_2=0}^{N_2-1} P(n_1,m_2)\, B_R(n_2,m_2) \tag{8.2}$$

where $A_R(n_2,m_2)$ and $B_R(n_2,m_2)$ are the 1-D transform forward and inverse kernels respectively with m_2 as transform domain rows coordinate. Many different types of unitary transforms, for use in hybrid coding, including the Fourier, Sine, Cosine, Hadamard, slant and Karhumen-Loeve have been investigated. [1B,16B].

Each DPCM coder forms the difference signal

$$D(n_1,m_2) = P(n_1m_2) - \hat{P}(n_1,m_2) \tag{8.3}$$

with the transformed coefficient estimate $\hat{P}(n_1,m_2)$ being formed by a weighting of the coefficient from the previous line according to the equation

$$\hat{P}(n_1,m_2) = a_1(m_2)\, P(n_1-1,m_2) \tag{8.4}$$

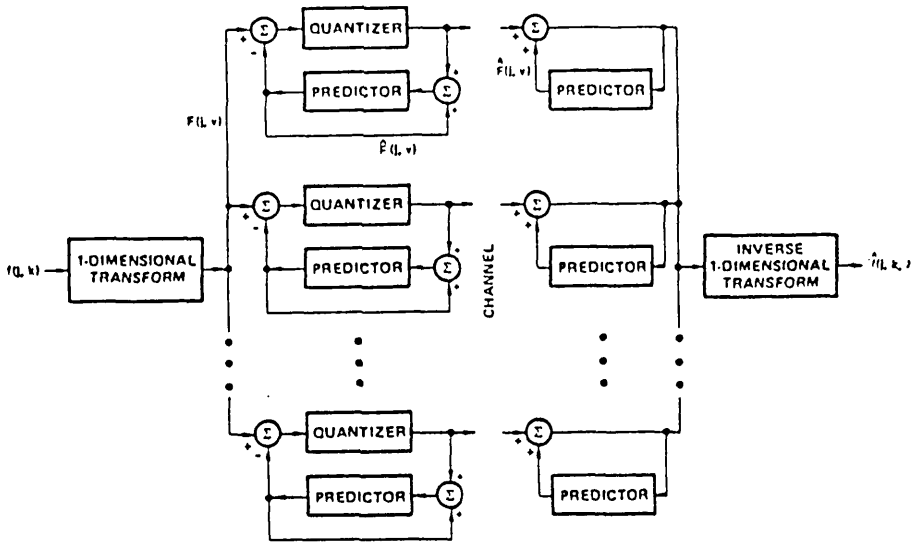


Fig. 8.2.1: One-dimensional hybrid transform/DPCM coder

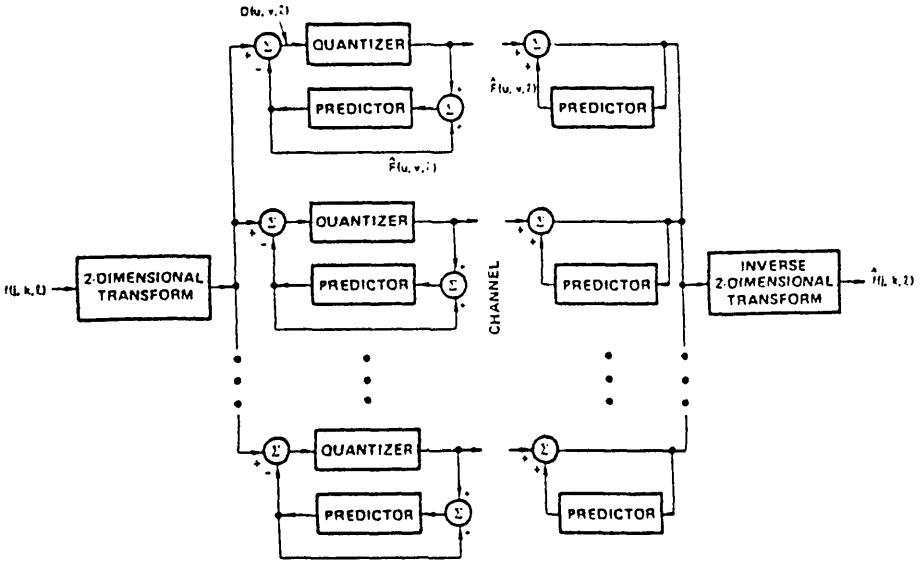


Fig. 8.2.2: Two-dimensional hybrid transform/DPCM coder

The weighting coefficient $a_1(m_2)$ is chosen to minimise the mean-square prediction difference $E\{D^2(n_1,m_2)\}$.

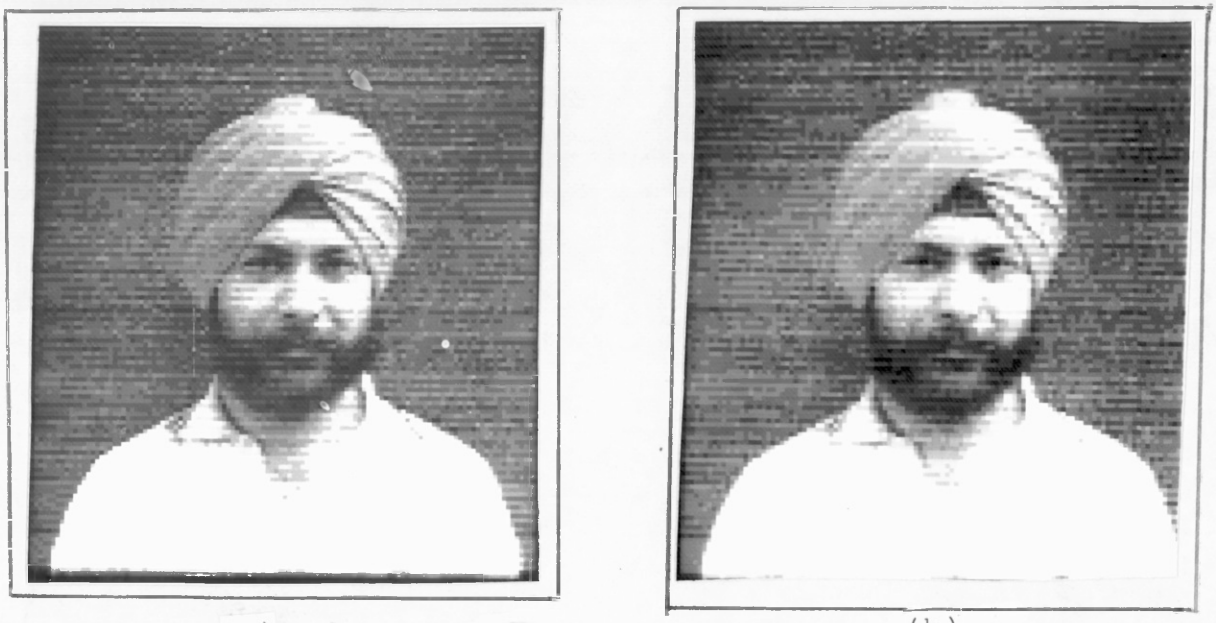
The difference signal is quantized and coded for transmission to the receiver. A zonal bit allocation strategy is used in which the number of code bits assigned to each difference signal is set in proportion to the logarithm of its variance [35B].

8.2.1 Experimental Performance of 1-D Transform/DPCM System

Computer simulations have been performed on two test images, as shown in Figs. 6.3.2 and 6.3.3 of Chapter Six to evaluate the performance of the one-dimensional hybrid coder. Because of its implementation simplicity a one-dimensional Hadamard-transform (HT) was employed on each line of the image data, followed by a first order predictive coder.

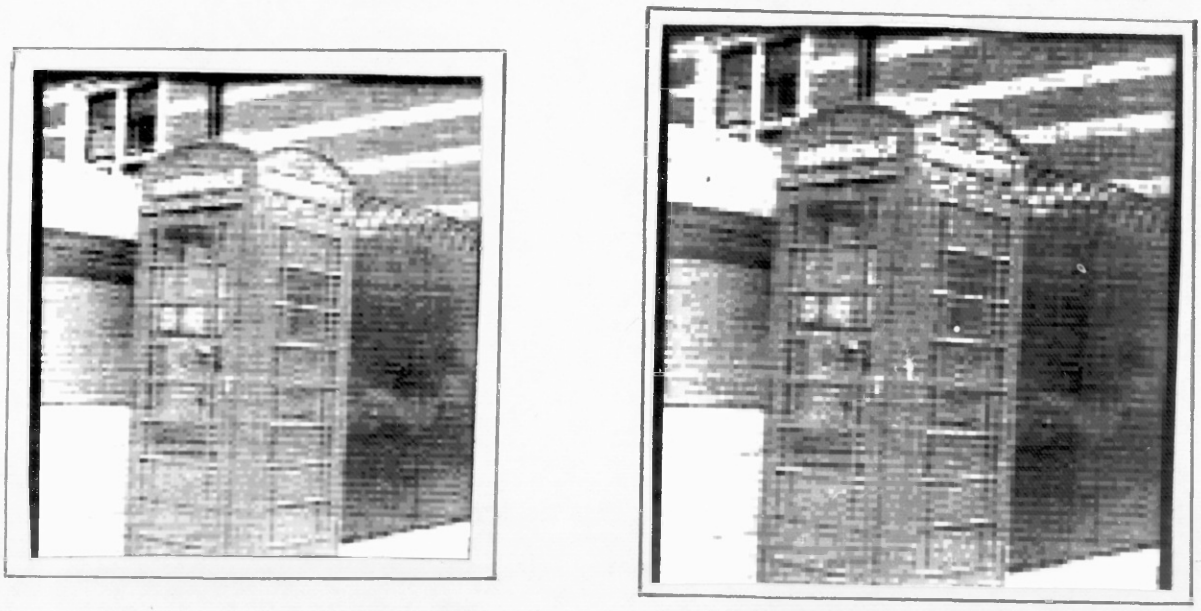
Fig. 8.2.3 illustrates hybrid one-dimensional HT/DPCM coder reconstructions of the Fig. 6.3.2 at average bit rates of 1.5-3.0 bits per pixel. Application of the same system to the telephone box data gives subjectively poorer coding results owing, in part, to the higher spatial frequency content of the source image as shown in Fig. 8.2.4. The corresponding NMSE values, as a measure of image fidelity, are also given for each processed image. Better quality coded-images could have been obtained if adaptive DPCM coders were used.

Since the unitary transformation involved is a one-dimensional HT of individual lines of the pictorial data, the equipment complexity and the number of computational operations is considerably less than that which is involved in a two-dimensional HT.



(a) (b)

Fig. 8.2.3: Coded images by hybrid transform/DPCM:
(a) At bit rate of R = 2.6 bits/pixel and NMSE = 1.89×10^{-2}
(b) At bit rate of R = 1.9 bits/pixel and NMSE = 2.0×10^{-2}.



(a) (b)

Fig. 8.2.4: Coded images by hybrid transform/DPCM:
(a) At bit rate of R = 1.9 bits/pixel and NMSE = 1.19×10^{-3}.
(b) At bit rate of R = 1.5 bits/pixel and NMSE = 1.21×10^{-3}.

8.3 Two-dimensional Transform/DPCM Coding

In block-transform coding,illustrated in Chapter Five, an image was divided into smaller blocks in order to reduce computational complexity. Since the elements of various blocks remain correlated in the transform domain the efficiency of the system will be reduced.

Habibi [33B] introduced the two-dimensional hybrid coding system that utilizes two-dimensional transformation of each block followed by bank of DPCM systems that would exploit the inter-block correlation, thus improving the coding efficiency of the system.

The block diagram of the system is shown in Fig.8.2.2.

8.4 A New One-dimensional Hybrid Coding Technique using Vector-quantization

In this section a new hybrid coding technique has been proposed [34B]. where one-dimensional transforms are taken along image rows and a vector-quantization process [18B, 36B] is applied on the columns of the transformed image data, exploiting the inter-row correlation.

Fig. 8.4.1 shows the block diagram of the proposed hybrid transform-vector quantization coder. In operation, a one-dimensional transform is taken along each image line of the $N_1 \times N_2$ image block, $x(n_1,n_2)$, yielding a sequence of transform coefficients,

$$P(n_1,m_2) = \sum_{n_2=0}^{N_2-1} x(n_1,n_2)\, A_R(n_2,m_2) \qquad (8.5)$$

where $A_R(n_2,m_2)$ is the one-dimensional transform kernel.

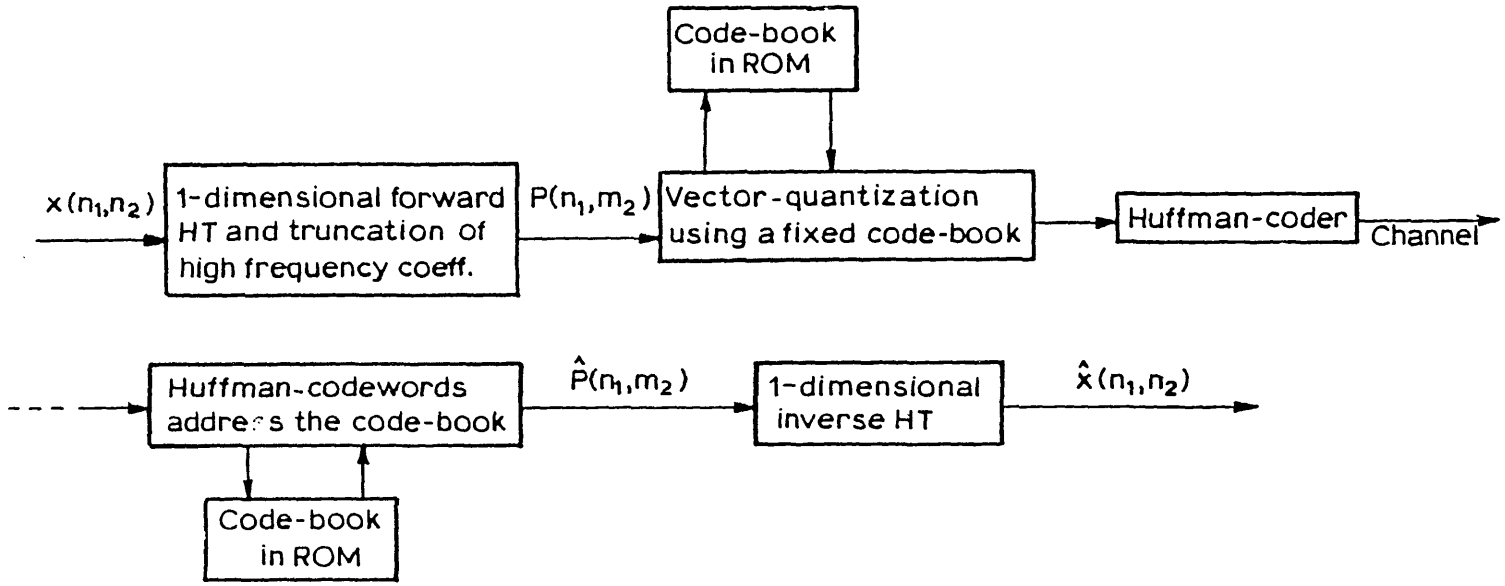


Fig. 8.4.1: The proposed one-dimensional hybrid/vector quantization system.

Each transformed row $P(n_1,m_2)$ is normalized by the square root of the expected variance given by the expression

$$\sigma^2(m_2) = \frac{1}{N_1} \sum_{n_1=0}^{N_1-1} [P(n_1,m_2)]^2 - [\mu(m_2)]^2 \quad \text{for } m_2=0,1,\ldots N_2-1 \tag{8.6}$$

where $\mu(m_2)$ is the mean variance.
Then a large number of the high frequency coefficients are discarded as in the zonal coding process of Chapter six, section 6.3.1.1, where the high frequency coefficients with small expected variance are discarded.

The neighbouring rows of the transformed image are highly correlated. This inter-row correlation can be exploited by constructing a set of K-dimensional vectors, where each vector consists of the K "column-wise" corresponding samples taken from K neighbouring rows. Each vector is subsequently compared with a code-book of standard vector-templates, and is represented by its nearest matching vector-template in a Euclidean sense as given by the expression

$$d_E = \sum_{i}^{K} (x_i-u)^2 \tag{8.7}$$

where (x_i) and (u) are the image and vector template points respectively. Then the set of permissible vector templates is Huffman-coded and transmitted. At the receiver the transformed image is reconstructed using the corresponding vector-templates in place of the original vectors. In practice this reconstruction can be done very rapidly by using the code-words to address a Read-Only-Memory (ROM) in which the standard vector-templates are stored. The above vector

quantization process can be done in parallel by several code-words addressing a bank of parallel code-books at the same time. This would increase the throughput of the system, but copies of the same code-book have to be stored in several ROM's.

The vector quantization process is a vector (or block) quantizer consisting of a code-book of possible reproduction vectors and a minimum distortion encoding rule. An N-level K-dimensional quantizer is a mapping, q, that assigns to each input vector, $x = (x_o, \ldots x_{q-1})$ a reproduction vector, $x = q(n)$, drawn from a finite reproduction alphabet, $\hat{A} = y_i$, $i = 1, \ldots N$. The quantizer q is completely described by the reproduction alphabet (or code-book) $\hat{A}$ together with the partition, $S = \{S_i;\ i = 1, \ldots N\}$, of the input vector space into the sets $S_i = \{x;\ q(n) = y_i\}$ of input vectors mapping into the ith reproduction vector.

The problem of generating an optimal code-book with respect to a distortion measure has been considered recently by Linde, et al [18B]. The approach is to define a reasonable distortion measure [37B], [38B] and attempt to create an optimal (code-book) given an initial estimate of the code-book. This is done by an iterative algorithm suggested by Lloyd's Method I [37B] that employs a long training sequence to optimize the code-book.

The code-book used in the proposed system is formed by the algorithm in [18B], using the low sequency samples of the transform domain with Mean Square Error as a distortion measure. After complete decoding, denormalisation of the coefficients is effected and an inverse transform is performed on the rows to retrieve the picture.

8.4.1 Some Experimental Results for 1-Dimensional Transform-Vector Coding

The proposed one-dimensional system in Fig. 8.4.1 was simulated for the test image in Fig. 6.3.2; the unitary transforms used for this system were the one-dimensional Hadamard transform (HT) because of its computational simplicity.

Fig. 8.4.2 shows coded images at bit rates of R = 0.8 - 1.1, using the designed code-books of size (N,K) = (42-230,4).

Processed images were compared with those using one-dimensional hybrid coding of Habibi [33B] which are shown in Fig. 8.2.3 of section 8.2. The results showed that the coded images by our technique had much lower NMSE than the Habibi technique; in addition their visual qualities were marginally better. We believe that this was due to code-book design using NMSE as a distortion measure. Better results could be obtained if the code-book design is based upon a more complex distortion measure which is subjectively better. However, this proposed hybrid technique is much simpler and faster than that of Habibi, since at the receiver the code-words will address a code-book, or several code-books in parallel, stored in ROM, where a large block of date is reconstructed. The simplicity of the proposed hybrid technique is very attractive for interframe coding of TV images for real time processing.

The code-books for the above simulations consisted of the most probable vector-templates in the low sequency region of the HT domain, which were formed in an ad hoc manner

(a)

(b)

(c)

Fig. 8.4.2: Coded images by the proposed one-dimensional hybrid system:

(a) At bit rate of R = 1.1 bits/pixel, NMSE = 4.9×10^{-3} with codebook size of CB = 230.

(b) At bit rate of R = 0.95 bits/pixel, NMSE = 7.37×10^{-3} with codebook size of CB = 193.

(c) At bit rate of R = 0.847 bits/pixel, NMSE = 7.56×10^{-3} with codebook size of CB = 42.

by employing a training set of several transformed images.

8.5 A New 2-Dimensional Hybrid Coding Technique using a Vector-Quantization Technique [39B]

In standard two-dimensional zonal coding (Chapter 6, section 6.3.1.1), as a consequence of the computational complexity involved, an image array $x(n_1,n_2)$ is divided into small blocks $f_i(n_1,n_2)$ for i = 1,... B(the number of the blocks), where each block i is coded as a unit, independent of all other blocks. This, unfortunately, reduces the efficiency of the coder since the elements of the various blocks remain correlated in the transform domain. Here we introduce a new two-dimensional hybrid coder, where the vector-quantization scheme in the previous section is used to exploit the correlation between the blocks.

Fig. 8.5.1 shows the proposed system, where a two-dimensional unitary transformation of each block i is obtained, given by the expression (8.8)

$$F_i(m_1,m_2) = \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} f_i(n_1,n_2)\, A_R(n_1,n_2)$$

$$\text{for } m_1 = 0,1,2, \ldots N_2-1$$
$$m_2 = 0,1,2, \ldots N_2-1 \qquad (8.8)$$

where $A_R(n_1,n_2)$ is the forward transform kernel. Then a zonal sample selection process is used where the coefficients with the largest variance, given by expression (8.9) are selected [19B]

$$\sigma_k^2(m_1,m_2) = \sum_{i=1}^{K} [F_i(m_1,m_2)]^2 \qquad (8.9)$$

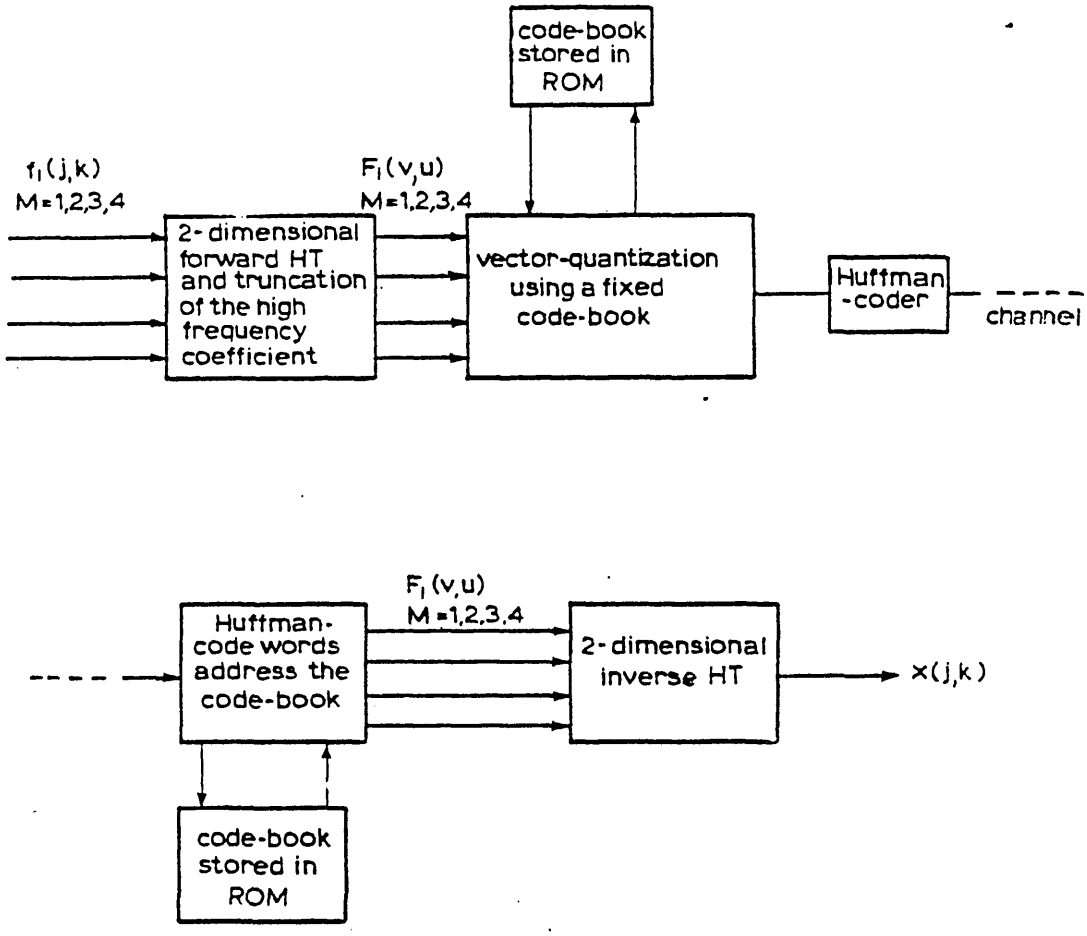


Fig. 8.5.1: Block diagram of the proposed two-dimensional hybrid system.

where K is the number of blocks. Since the elements of various neighbouring blocks are highly correlated, this inter-block correlation can be exploited by clustering the corresponding transformed samples of each K neighbouring block into vectors of dimension K. The vector quantization process, illustrated in the previous section, is applied on the vectors and the corresponding code-words transmitted. At the receiver these code-words will address the code-book stored in ROM to reconstruct the transformed blocks by substituting the corresponding vector-templates in place of the original vectors.

An inverse two-dimensional unitary transform is applied on each block as given by eqn.(8.10).

$$f_i(n_1,n_2) = \sum_{m_1=0}^{N_1-1} \sum_{m_2=0}^{N_2-1} F_i(m_1,m_2)\, B_R(m_1,m_2) \qquad (8.10)$$

$$\text{for } n_1 = 0,1,2,\ldots N_1-1$$

$$n_2 = 0,1,2,\ldots N_2-1$$

where $B_R(m_1,m_2)$ is the inverse transform kernel.

8.5.1 Experimental Results for the Proposed 2-dimensional Hybrid Coder

The proposed hybrid coder was simulated on a digital computer. The experimental data were the two pictures displayed in Chapter Six, Figs. 6.3.2 and 6.3.3. A block size of 8x8 with two-dimensional Hadamard transform as our unitary transform were used.

Fig. 8.5.2 shows the processed images at bit rates of R = 0.51 - 1.89 bits/pixels, using the designed code-books of size (N,K) = (70-255,4).

Comparing the processed images with that of standard zonal coding in Fig.6.3.7, Chapter Six, Section 6.3.1 the proposed system performs better.

The code-book used in the simulation was formed by the algorithm in section 8.4 using mean-square-error as the distortion error.

8.6 Conclusions

This chapter is devoted to hybrid image coders. A short comparison between DPCM and Transform coders is given in section 8.1. Also the concept of the hybrid coders is explained. In section 8.2 one-dimensional hybrid transform DPCM systems are reviewed. A one-dimensional Hadamard transform was combined with a first order predictor. One-dimensional transform is performed on each line of the image and a DPCM on the transformed coefficients, no adaptivity was introduced in the simulated system, which is discussed in section 8.2.1. Two-dimensional transform/DPCM coders are introduced in section 8.3. Full details of the applications of these coders are given in chapter nine.

In section 8.4, a new one-dimensional hybrid coder is introduced. This coder consists of a one-dimensional Hadamard transform and a vector quantization scheme. The vector quantization scheme is a look-up table procedure, where each vector of transformed coefficients is compared with a set of vectors (codebook). A distance measure, such as Euclidean

(a)

(b)

(c)

Fig. 8.5.2: Coded images by the proposed two-dimensional hybrid system:

(a) At bit rate of R = 1.7 bits/pixel, NMSE = 2.05×10^{-3} with codebook size of CB = 255.

(b) At bit rate of R = 1.52 bits/pixel, NMSE = 2.05×10^{-3} with codebook size of CB = 193.

(c) At bit rate of R = 1.33 bits/pixel, NMSE = 4.3×10^{-3} with codebook size of CB = 103.

distance, is used to find the best match. Other more complicated distance measures can be employed.

Codebook design is investigated in section 3.4. The codebook is formed by employing a long training sequence. Fixed codebook sizes were used throughout the simulation, although adaptive codebooks could be used. One adaptive scheme could be to start with a codebook and modify it as image coding is performed. So some vectors have to be transmitted. The effect of noise is not investigated in this chapter, but, it is believed that it will perform much better than hybrid transform/DPCM coder, since channel noise only affects each vector; no more noise is introduced in the neighbouring vectors. However, in DPCM coders any noise is carried to the next sample because of the predictor. No adaptivities are introduced in the proposed system. In chapter nine codebook adaptation is explained. Experimental simulations are given in section 8.4.1. In section 8.5 the two-dimensional Transform/Vector quantization coding system is introduced. Simulations are given in section 8.5.1. Applications of the proposed coders to moving images are considered in chapter nine.

A large number of the high frequency coefficients are zonally discarded in the proposed coders. An adaptive coder would discard a number of these coefficients with respect to the energy content of each block of rows. Also the transform coefficients are normalised by the square root of expected variance, which was calculated from the transformed rows. It is also possible to estimate the variance recursively from the transformed coefficients.

CHAPTER NINE

CHAPTER NINE

INTERFRAME CODING OF MOVING IMAGE SEQUENCE

9.1 Introduction

Transmission of moving images is an exciting application for pictorial data compression. An analog TV the signal usually has a bandwidth of 4.3 MHz. The digital TV signal is formed by sampling at about 10.76 MHz and digitizing into an 8-bit PCM signal, for which, in turn, a bitrate of 86 M bits/sec. is required. Thus, the essential problem in digital TV coding is to reduce the bandwidth at the expense of bit error rate and an acceptable picture quality. In this chapter we shall review some digital image source interframe coding techniques for several applications and then propose two new techniques.

Section 9.2 will introduce some specific applications where coding is required to achieve the bit rate. In Section 9.3 the response of the human visual system to moving images is discussed. A review of the interframe coding techniques is given in Section 9.4. In Section 9.5 the new techniques are introduced and simulated and the conclusion is given in Section 9.6.

9.2 Application of Digital Moving Images

(1) One of the major applications of digital TV transmission is in Cable-TV network. Coaxial cable (or optical) networks seem to be suitable for TV transmission because of their large bandwidth. The Cable-TV network systems all use principles that have been developed for local-area networks

(LAN's) whereby a number of users share a data highway. LAN's are conceived as networks connecting together a number of nodes normally using a bus or ring topology. They must cater for the fact that the independently initiated transfers will occasionally demand more bandwidth than the network has available and also that they must continue to operate when one or more nodes fail. The data transfer is done by circuit, message or packet switching. For example in packet switching, data is assembled into packets with a header and a tail, and flow of packets is controlled by contention or collision-control strategies in the case of buses, and by daisy-chain, token or register strategies in the case of rings. Other services that TV network could offer are data services, video-tax services, individual video services, such as customer access to a video library.

(2) Another application is in Picturephone or Teleconferencing where cameras are stationary and scenes consist mainly of small areas moving in front of a relatively large stationary background. A considerable saving in transmission rate can be achieved by coding techniques such as Frame replenishment, motion compensated predictive and adaptive hybrid Transform/Vector Quantization coding techniques. In the above applications, the image resolution is very small and it is nearly a face to face communication. Picturephone is very valuable for distance communication between deaf people since the movements of lips or hand will convey the necessary information.

(3) Satellite TV transmission whereby European TV broadcasts can be received is another application for image coding. Here users receive transmission from satellites either with their own small disks or through optical-fibre cables from central receivers. Recently digital TV sets have been developed; such TV sets are able to receive digital transmissions from other countries.

The bandwidth and the coding techniques required for transmission of moving images are application-dependent. For example in commercial TV images the usual quality as well as the image resolution should be very high, thus a large bandwidth is required. However in video conferencing or picturephone the movement is low to moderate and a large degradation is tolerable, thus a smaller bandwidth can be achieved.

9.3 Response of the Human Visual System to Moving Images

In order to exploit the redundancies in moving images by interframe coding techniques a knowledge of the psychophysics of vision is required. Studies have been made to obtain a linear system model for the human visual system. It is found that the visual system can be represented by a three-dimensional bandpass filter [4OB], [41B]. Other features of the human visual system are that at high temporal frequencies the spatial contrast sensitivity is reduced, and similarly at high spatial frequencies there is an overall decrease in flicker sensitivity. The behaviour of the human visual system has also been investigated in tracking and non-tracking modes. For example, if an image can be tracked the retinal integration

is minimized and the spatial resolution requirement in the moving area is high. However, if an object cannot be easily tracked, then the human visual system can tolerate a loss of spatial resolution [42B].

Seyler and Budrikis [43B] have found that the human observer does not perceive a temporary reduction of spatial detail after scene changes. In the following sections several coding techniques are introduced,some of which will employ psychophysical redundancy of human visual behaviour.

9.4 Interframe Coding Techniques Review

In the following sub-section some interframe coding techniques are introduced and a new technique is proposed.

9.4.1 Subsampling Techniques in Spatial and Temporal Domain and Frame Repeating

The types of subsampling generally employed are horizontal, vertical, field (which is vertical-temporal subsampling) and frame subsampling (temporal only) [44B]. Pease and Limb [45B] investigated the effect of spatial and temporal subsampling on television signals by dividing the picture into stationary and moving areas which are appropriately subsampled. They found, due to psychophysical integration of moving objects that it is possible to reduce the spatial sampling rate in moving areas (non-stationary areas) of a television picture without degrading the picture quality. Also temporal subsampling can be employed in stationary areas. At the receiver the unsampled picture elements were replaced by interpolating between the sample elements.

In slow moving pictures such as teleconferencing temporal resolution can be greatly reduced without impairing picture quality. A simple method of achieving this is by frame repeating, in which one new frame of information out of n is transmitted and for the remaining (n-1) frames this one frame is just repeated. As would be expected if the object moves fast enough, the perceived motion is very jerky. If repeating is done on sub-image blocks of the picture, the object will appear broken at fast motion. The subsampling technique can be done also in the transform domain.

9.4.2 3-dimensional Differential Pulse Code Modulation and its Adaptivities

Predictive coding techniques of Chapter seven may also be used to decorrelate the samples for interframe coding. Early work with interframe predictors was based on using the frame difference for each picture element [46B], [47B], but frame difference prediction will be effective only for stationary sections of a picture. In moving areas it is advantageous to use a predictor combining elements from both the actual and previous frame [48B]. Simplest predictors are first order linear predictors which assume images are samples of a stationary random process. Since this is in general not the case, adaptive linear predictors and non-linear predictors can lead to better results. The number of previous pels used in the predictor is called the order of the predictor and if prediction is based on the previous horizontal, vertical and field elements it is known as a 3-D predictor.

Haskell [23B] studied the performance of a number of such fixed predictors on scenes with varying amount of motion. It was found that frame difference performs well in areas with little motion or stationary areas, while intrafield predictors such as element difference do better when there is greater motion. The reason is that as the speed of motion in a television picture increases, the spatial correlation between moving-area picture element will also increase owing to the integrating effect of the television camera. Moreover, the temporal correlation among such picture elements will decrease. The combination intra- inter-field schemes were found to be better than either of these and tended to be less sensitive to the amount of motion.

The fact that images are not stationary and, in fact, contain many edges and contours has led to the use of non-linear predictors. These predictors attempt to determine the direction of contours in the image and choose the prediction accordingly.

Most interframe coders have used a fixed quantizer; greater improvement is possible over fixed quantizers by the use of adaptive quantizers. The adaptation can be based on statistical and on psychovisual criteria, and can involve switching between a number of fixed quantizers, or changing the basic step size of a given quantizer. For example, in areas of the picture where there is very little activity, a fine quantizer can be used, and in areas with more activity the quantization can be coarser [49B].

Entropy coding is used to reduce the bit rate from the DPCM coder since the output levels of the quantizer are not

equiprobable, but this will in return produce a variable data rate, and hence the system will require a buffer memory.

9.4.3 Movement-Componsated Predictive Coding Techniques

A time-varying image to be transmitted consists in general of a number of objects with different motions superimposed on a background. If the camera is fixed, the background is stationary while in the case of panning it moves with approximately uniform velocity. In either case, with the exception of newly exposed background and foreground caused by motion, each pel is present in the previous field, displaced by an amount dependent on the motion of the given object. Hence, if the displacement of the object is known, a picture element on the same position of the moving object in the previous frame becomes a better sample for prediction in predictive interframe coding. This forms the basis for the schemes known as movement-compensated predictive coding [50B], [51B].

The technique will first estimate the displacement vector for the image; a simplified approach is to segment the image into fixed background and objects and then estimate the displacement. The displacement estimates are then used to generate the movement-compensated prediction, and finally the prediction errors are quantized.

Motion displacement estimation usually involves considerable calculations. There are two main classes of methods for estimating translational displacements; correlation or matching techniques and differential methods. Recently Netravali and Robbins [51B] developed simple and practical algorithms for estimating the motion displacement, utilizing

the relation between the spatial differential and temporal differential signals. Their algorithm is called the "pel recursive" method, which minimizes recursively the motion estimation error on the basis of a steepest descent algorithm. The other method of motion estimation is an extension of pattern matching used in scene analysis. The displacement estimate is done on a block-by-block basis. Constant translation is assumed within a block of picture elements. A present frame is divided into small sub-blocks and each sub-block is compared with its previous frame picture by changing the relative spatial position. The displacement vector is determined by searching for the best-matched picture in the previous frame [52B].

In teleconferencing use, pictures to be transmitted are, typically, scenes with a few people sitting behind a table or an overview picture of all the attendees in the conference room. Motions are usually translational and little rotation or scaling happens. Thus the above algorithms can be used. But for other movements such as rotation and zooming and panning more complicated techniques are required [51B]. Further work in this field will be to estimate other motions such as rotation.

9.4.4 Conditional Frame Replenishment Coding Technique

In digital application such as video-telephone or teleconferencing where typical cameras are stationary and scenes consists mainly of small areas moving in front of a relatively large stationary background, a considerable saving can be achieved with PCM by sending the parts of the picture that are changed between frames [53B]. The location

as well as the intensity level of each pixel has to be transmitted. Thus for a high moving image sequence a lot of overhead information is required. Since the moving-area picture elements are highly correlated the bit rate can be reduced if these elements are predictively coded. The differential signal is then transmitted and location of pixels are transmitted by a clustering method [54B]. The above technique is a variable bit rate coding technique since the bit rate is dependent upon the change in movements between the frames. A buffer is required to smooth the data bit rate for transmission through a fixed bit rate channel. A great deal of study has been carried out on segmentation of the picture into moving areas that have changed significantly since the previous frame and into stationary background areas that have not changed significantly [55B]. Study on the adaptivity of the linear predictor, subsampling of the moving areas when buffer overflow occurs and buffer and channel sharing by several frame replenishment coders, have also been investigated.

9.4.5 Hybrid Transform/DPCM Coders

The concept of hybrid transform/DPCM coding was introduced in the previous chapter for still pictures. This concept can easily be extended to a three-dimensional moving image sequence [56B]. In the interframe hybrid coder, a two-dimensional transform is performed on each block, and predictive coding (DPCM) is applied in the temporal direction. The two-dimensional transform on the image array $u(i_1,i_2,i_3)$ is given by

$$U(K_1,K_2,i_3) = \sum_{i_1=0}^{N_1-1} \sum_{i_2=0}^{N_2-1} U(i_1,i_2,i_3)\, A(K_1,i_1,K_2,i_2)$$

$$\text{for } K_1 = 0, \ldots N_1-1$$

$$\text{for } K_2 = 0, \ldots N_2-2$$

where $A(K_1,i_1,K_2,i_2)$ represents the two-dimensional transform kernel. At each spatial frequency (K_1,K_2) a prediction $\hat{U}(K_1,K_2,i_3)$ is performed. The prediction error is given by

$$P(K_1,K_2,i_3) = U(K_1,K_2,i_3) - \hat{U}(K_1,K_2,i_3) \qquad (9.1)$$

This prediction error is quantized by zonal coding strategy where each error prediction is quantized using a number of bits dependent on the energy of the prediction error at frequency (K_1,K_2). In general a portion of the high frequency coefficients need not be coded by DPCM.

In the transform domain the correponding low frequency coefficients of each block are more correlated than those of the high frequency coefficients, thus adaptive predictors can be designed. Roese [56B] showed that an improvement in coding performance can be obtained by spatially adapting the hybrid coder to use the temporal statistical measures of the transform coefficients temporal difference signal in each block. Local adaptation to the measured statistics of each sub-block will normally produce improved coding results when compared with non-adaptive implementations. However, adaptation does result in increased coder complexity.

Another adaptation is to use as prediction the transform coefficients for a block in the previous frame shifted by the estimate of the local velocity [51B]. This is similar to the movement-compensated prediction technique. Here the displacement estimation technique recursively estimates the displacements from the previously transmitted transform coefficients; alternatively the algorithm, due to Limb and Murphy, estimates displacements by taking ratios of

accumulated frame difference and spatial difference signals in a block.

9.4.6 3-dimensional Transform Coding

In order to incorporate inter-frame redundancy reduction, three-dimensional transform coders are employed. A sequence of image frames $U(i_1,i_2,i_3)$ undergoes a three-dimensional transform in blocks of size $N_1 \times N_2 \times N_3$ pixels according to the general formula

$$U(K_1,K_2,K_3) = \sum_{i_1=0}^{N_1-1} \sum_{i_2=0}^{N_2-1} \sum_{i_3=0}^{N_3-1} U(i_1,i_2,i_3)\, A(K_1,i_1,K_2,i_2,K_3,i_3) \tag{9.2}$$

$$\text{for } K_1 = 0,\ldots N_1-1$$
$$K_2 = 0,\ldots N_2-1$$
$$K_3 = 0,\ldots N_3-1$$

where $U(K_1,K_2,K_3)$ is the transformed block and $A(K_1,i_1,K_2,i_2,K_3,i_3)$ the forward unitary transform kernel. The transform coefficients are then quantized and coded for transmission. Zonal sampling or zonal coding quantization and coding strategies are usually employed, Roese [56B]. At the receiver the coefficients are decoded and an inverse transform process is performed to reconstruct the frame sequence.

Adaptive three-dimensional transform coding has also been investigated with adaptation based on the bit assignment matrix and local unitary transforms which are currently used as Fourier, Cosine, Hadamard, Harr, etc.

Recently study has shown that the performance of three-dimensional transform interframe coders greatly exceeds that of two-dimensional transform coders. However, the main disadvantage of such coders are requirement of excessive storage and computation. Although cheap memory and array processors have solved the problem to some extent.

9.4.7 Other Techniques

(a) Two-dimensional vector quantization technique [58B,59B]:

Here each frame is divided into blocks of size 2x2 and block vector quantized. The above technique can be extended to three-dimensional vector quantization easily. But the size of the code-book will increase as the number of frames is increased.

(b) Interpolative Coding [16B]:

Here the image intensity variation is approximated by a polynomial.

(c) Contour or Feature Coding [16B]:

The image is first segmented and classified and contours are represented (polygon approximation).

(d) Two-component Image Coding [16B]:

The statistics of the image signal are not well defined but by converting the signal into components of different classes a better result can be obtained.

(e) Block Truncation Image Coding [60B,61B]:

This method divides an image into small pixel blocks, each of whose pixels are individually quantized to two levels, Y_o and Y_1, such that the block sample mean η and variant σ^2

are preserved. Thus the block is represented by a two-tone bitplane.

9.5 A New Interframe Coding Technique using Vector Quantization

In the previous section it was shown that coding techniques that exploit spatial as well as temporal correlation have higher performance than interframe coders. Thus in this section we introduce a new hybrid interframe coder [62B], where the spatial redundancy is exploited by performing a two-dimensional transform on each frame (block) and a vector quantization scheme on the transformed coefficients to exploit the temporal redundancy. The vector quantization scheme is similar to that used for the proposed intraframe coder in the previous chapter.

The block diagram of the proposed coding system is shown in Fig. 9.5.1.

In the proposed coding technique:

(1) Image sequence is sub-divided into three-dimensional block arrays each of size J x K x L pixels.

(2) A two-dimensional transform is performed on each spatial (horizontal and vertical directions) block of size J x K pixels, given by the expression

$$F(v,u,\ell) = \sum_{j=0}^{J-1} \sum_{k=0}^{K-1} f(j,k,\ell)\, A(v,j,u,k) \qquad (9.3)$$

$$\text{for } u = 0, 1, \ldots, J-1$$
$$v = 0, 1, \ldots, K-1$$

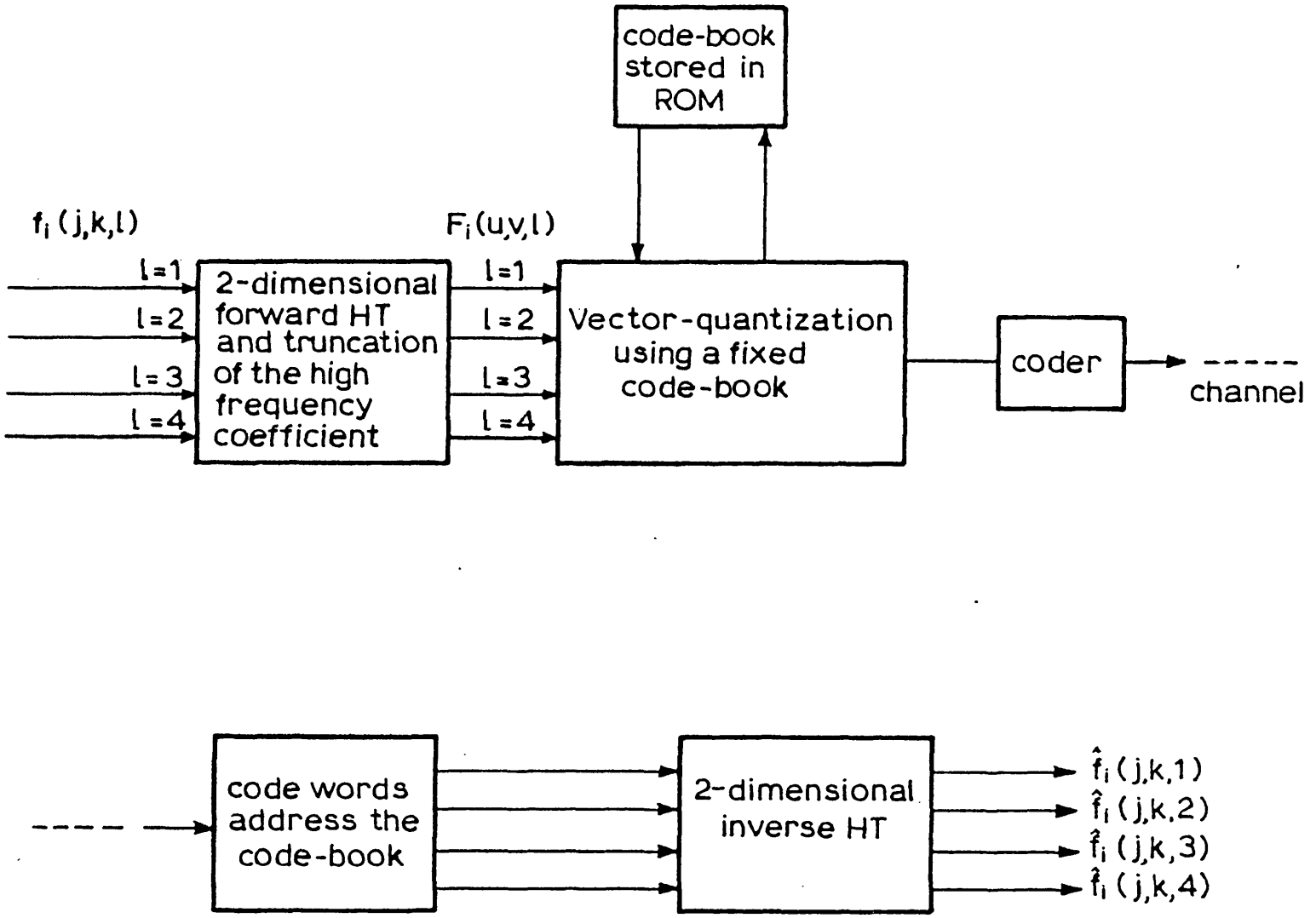


Fig. 9.5.1: The proposed interframe coding system

where f(j,k,ℓ) denotes a three-dimensional block array of amplitude values for a digital image sequence, and A(v,j,u,k) represents the forward two-dimensional transform kernel.

(3) Since most of the energy is clustered at low frequency coefficients a large number of high frequency coefficients are zonally discarded, and the remaining coefficients are normalized by their corresponding expected variance.

(4) Each spatial frequency (v,u) of L temporally adjacent blocks are clustered into vectors of dimension L. Each vector is then compared with a code-book of standard vector templates, and is represented by its nearest (in a mean-square error sense) matching vector-template. A binary code-word (Huffman code-words) is then assigned to each permissible vector template and transmitted.

(5) At the receiver the transmitted code-words will address a look-up table code-book to reconstruct the three-dimensional block array as shown in Fig. 9.5.2.

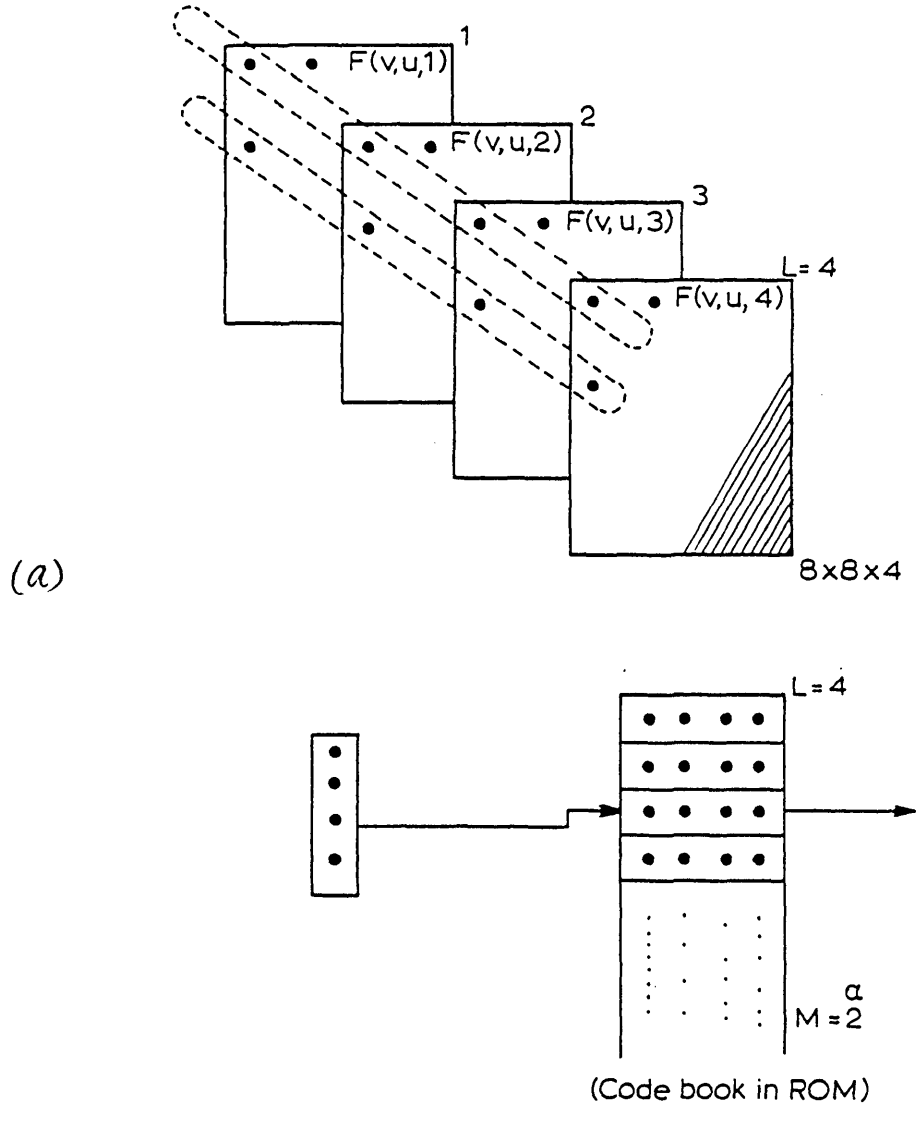


Fig. 9.5.2: (a) Shows the clustering of four frames

(b) Look-up table at the receiver

(6) A two-dimensional inverse transform is then applied on each block, as given by

$$f(j,k,\ell) = \sum_{u=0}^{J-1} \sum_{v=0}^{K-1} F(v,u,\ell)\, B(v,j,u,k) \qquad (9.4)$$

$$\text{for } j = 0,1,\ldots, J-1$$

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

where $B(v,j,u,k)$ is a two-dimensional inverse transform kernel

9.5.1 Simulation and Results

The proposed system was simulated on a digital computer for an image sequence shown in Fig. 9.5.1.1 where each frame has a low spatial resolution of 128 x 128 and is quantized to 8 bits per pixel, but displayed with only 16 quantization levels.

Fig. 9.5.1.2 shows the coded frames at bit rate of R = 0.37 bits/pixel/frame with code-book of size and dimension (N,M) = (31,4) respectively. The block array size used was 8x8x4 and the transformation applied was two-dimensional Hadamard Transform. About 62.5% of the high frequency coefficients, having a variance above a specified threshold, were discarded. The processed images were distorted at this bit rate especially at the edges since a large number of high frequency coefficients were discarded.

Fig. 9.5.1.3 shows the processed image sequence at R = 0.48 bits/pixel/frame where only 25% of the high frequency coefficients were discarded. Edges appear smoother and the

image sequence has a better visual quality. NMSE for all the processed frames was evaluated as a criterion for image quality. Since the energy of the transformed images is conveyed by the low frequency coefficients, the code-book was formed of the most probable vector-templates in the low frequency region of the Hadamard transform domain by the algorithm reported in the previous chapter. The processed images in Figs. 9.5.1.2 and 9.5.1.3 were reconstructed with only 31 vector templates, these being the most probable low frequency vector templates. The effect of the size of the code-book was investigated; Fig. 9.5.1.4 shows the processed images with code-book size of (134,4) and (285,4) at bit rate of R = 0.69 and R = 0.8 bit/pixel/frame respectively. Thus increasing the size of the code-book will allow the inclusion of a number of high frequency vector-templates, which may be used to reconstruct the edges. The code-book was formed with the method explained in the previous chapter.

The number of frames clustered together was fixed at four. A better technique would be to make the vector-quantizer adaptive to movement of the objects within the image sequence, such that in image frames with little movement a large number of frames are clustered; this is a consequence of the high temporal correlation of frames, the vectors being quantized with the appropriate code-book. However, with a high density of moving objects within the image sequence a smaller number of frames have to be clustered, since the scene changes considerably from one frame to another. The image sequence is then coded by switching to the appropriate code-book. In this adaptive technique several code-books of

Fig. 9.5.1.1: The input image sequence of resolution (128 x 128)

Fig. 9.5.1.2: Interframe coded image sequence at bit rates of R = 0.3 bits/pixel/frame with codebook size of CB = (31,4) with NMSE = $3.2...3.6 \times 10^{-3}$.

Fig. 9.5.1.3: Processed image sequence at R = 0.48 bits/pixel/frame with codebook size CB = (31,4), and NMSE = 2.8×10^{-3}.

(a)

(b)

Fig. 9.5.1.4: (a) Processed image sequence at R = 0.69 bits/pixel/frame with codebook size CB = (134,4), and NMSE = 2.3×10^{-3}.

(b) Processed image sequence at R = 0.8 bits/pixel/frame with codebook size CB(285,4) and NMSE = 1.9×10^{-3}.

different dimensions can be employed to adapt to the motion of the moving object. However, in normal commercial TV images there are several objects in the image sequence each moving with a different velocity. Thus in order to exploit the local movements within the image sequence the above adaption technique will be employed on the basis of three-dimensional sub-block arrays; this is proposed in the next section.

9.5.2 An Adaptive Vector-Quantization Coding Technique [65B]

The proposed system is shown in Fig. 9.5.2.1 (a), (b).

(1) In this system N_3 frames of an image sequence are stored in buffer memory and sub-divided into three-dimensional block arrays $f(j,k,\ell)$ each of size J x K x L pels.

(2) Two-dimensional unitary transform is performed on the spatial blocks as in the previous section.

(3) A large number of the high frequency coefficients are discarded and the remaining coefficients are normalized by their expected variance.

(4) An adaptive vector quantization scheme is then applied in the temporal direction by using two code-books (vector quantizer) of different dimensions, say L/2, L, in a manner whereby each block array is first classified as belonging to an area of slow (C_S) or rapid (C_R) motion. The classification is based on a temporal activity index which is a measure given by the difference between the sum of the A.C. coefficients of each sub-image sequence. The current block array

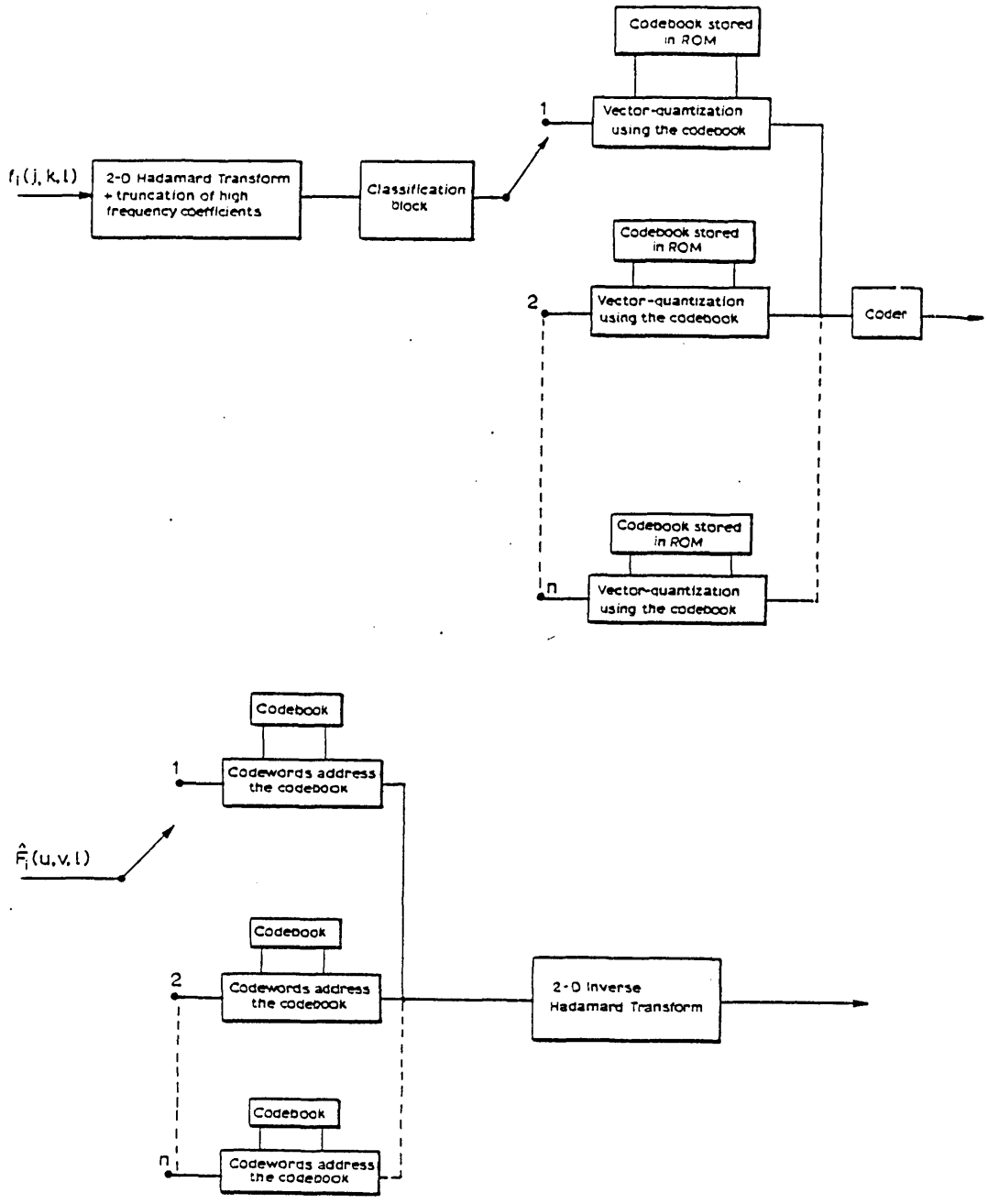


Fig. 9.5.1.2(a): The proposed adaptive interframe coder

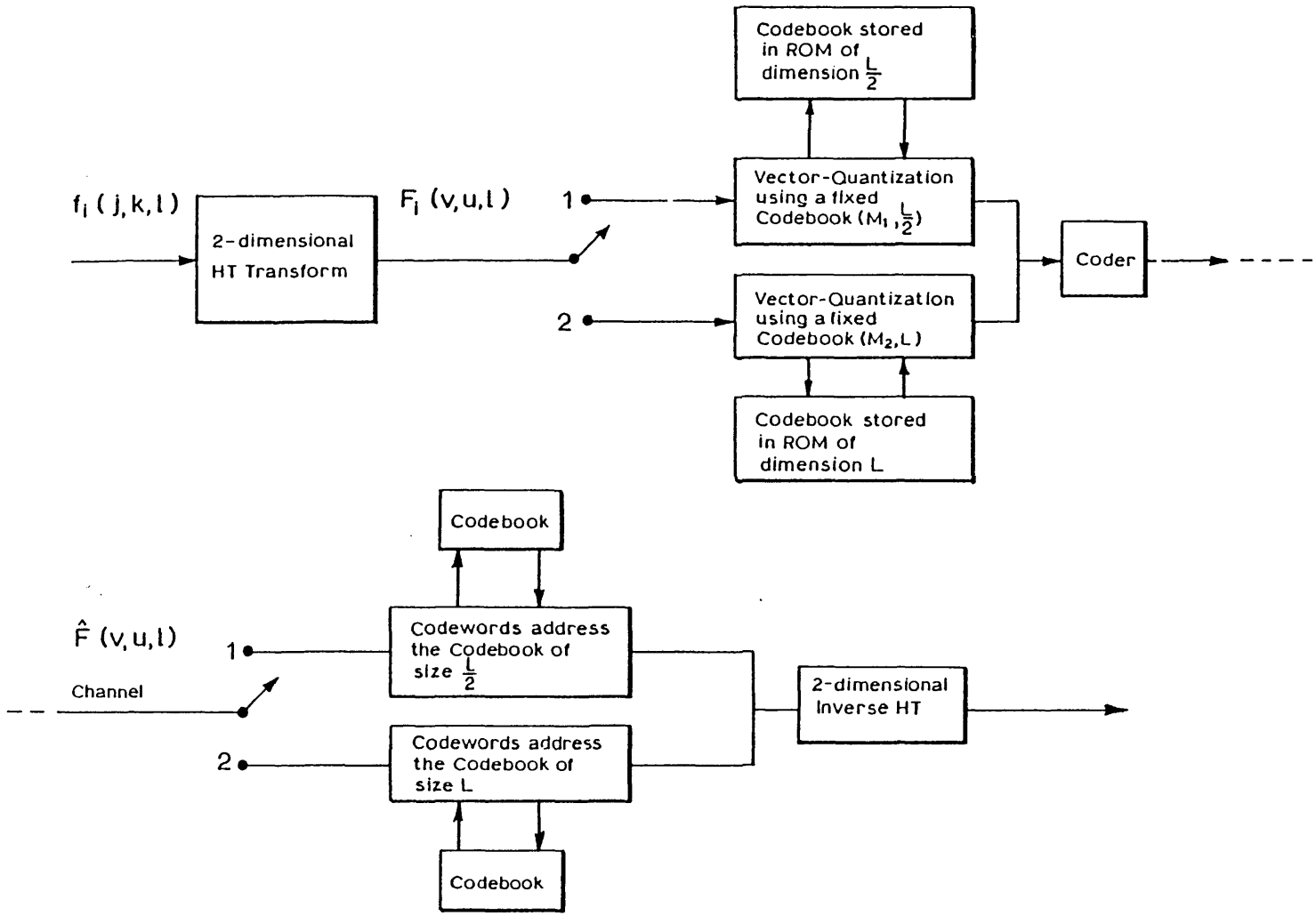


Fig. 9.5.1.2(b): The simulated adaptive interframe coder with only two codebooks

$f(j,k,\ell)$ is classified by specifying suitable thresholds for the two classes.

(5) The transformed block array $F(v,u,\ell)$ is coded by switching to the corresponding vector-quantizer. Each vector-quantizer will cluster each spatial frequency (v,u) of L temporally adjacent spatial blocks into vectors of dimension L or L/2 corresponding to each class. Then code-words representing their nearest matching vector-templates from the corresponding code-books are transmitted.

(6) At the receiver each transformed block $U(v,u,n_3)$ is reconstructed using the corresponding vector templates in place of the original vectors. The reconstruction is done very rapidly by code-words addressing the Read Only Memory's (ROM's) (look-up tables) as shown in Fig. 9.5.2.2.

(7) The coded image sequence is then formed by applying the inverse two-dimensional transform.

The overhead information is minimized by transmitting the encoded block as a packet where each packet will have a header which specifies which vector-quantizer is used, as well as any adaptivity and addition of error-correcting redundancy. The code-books or the vector quantizers were formed by the algorithm discussed in the previous chapter.

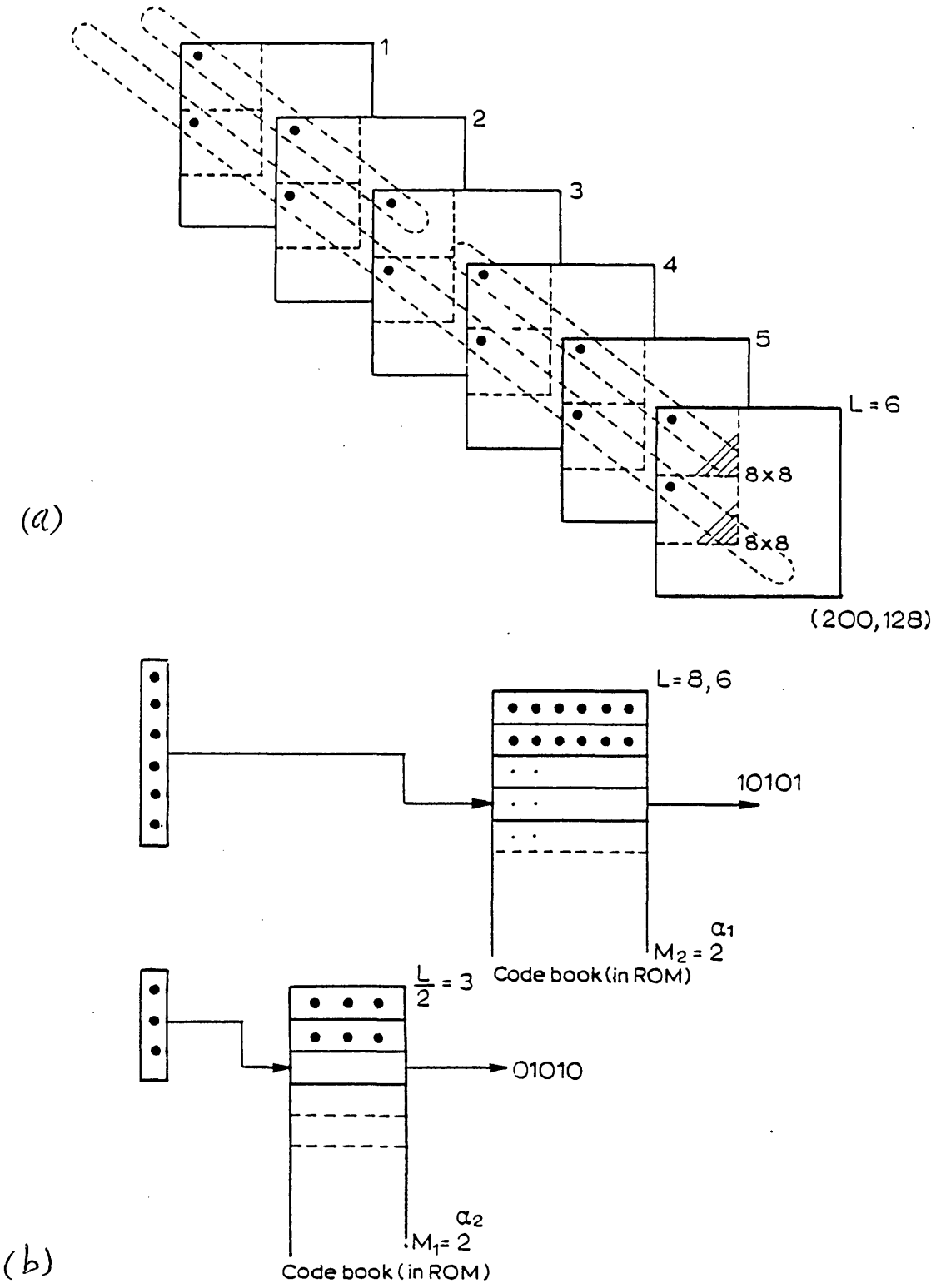


Fig. 9.5.2.2 (a) Shows the clustering of six frames and three frames.

(b) Look-up tables at the receiver.

Since the proposed technique resultsin a variable bit rate, a buffer is required for smoothing the data transmission over a fixed bit rate channel. Further improvement on the bit rate can be achieved by adaptive bit sharing through plural channels similar to that of the TASI technique [63B].

9.5.3 Simulation and Results

The adaptive system was simulated for a typical image sequence of resolution (136,200) quantized to 8 bits per pixel as shown in Fig. 9.5.3.1. The coded frames at average bit rate of R = 1.6 bits/pixel/frame are shown in Fig. 9.5.3.2 with block array of size (8x8x6) and code-books, of size and dimension(M_1 , L/2) = (285,3) and (M_2,L) = (111,6), and the number of blocks coded by each code-book was T_1 = 294, T_2 = 131 respectively. Fig. 9.5.3.3 shows the coded image sequence at R = 0.84 bit/pixel/frame with the same code-books but discarding a larger number of high frequency coefficients than in Fig. 9.5.3.2.

Fig. 9.5.3.4 shows the coded images at R = 0.6 bits/pixel/frame with code-books of size (M_1, L/2) = (313,4) and (M_2,L) = (203,8), and the number of blocks coded by each code-book was T_1 = 282, T_2 = 143 respectively.

The visual quality of the processed image sequence is much better than that of the non-adaptive technique in section 9.5.1. Using more code-books of different dimension could improve the quality of the image sequence and reduce the bit rate significantly. The only disadvantage of this technique is the requirement for storage of several frames. Adaptivity is based upon the movements within the image

Fig. 9.5.3.1: Original input image sequence of resolution (136,200) with amplitude resolution of 8 bits.

Fig. 9.5.3.2: Processed image sequence at bit rate of R = 1.6 bits/pixel/frame with codebooks of size CB1 = (285,3) and CB2 = (111,6) with NMSE = 9.2... 9.5×10^{-3}.

Fig. 9.5.3.3: Processed image sequence at bit rate of R = 0.84 bits/pixel/frame with codebooks of size CB1 = (285,3), CB2 (111,6), with NMSE = 1.0 ... 1.3×10^{-3}.

Fig. 9.5.3.4: Processed image sequence at bit rate of R = 0.6 bits/pixel/frame with codebooks of size CB1 = (313,4) and CB2 = (203,8) with NMSE = 1.8 ... 2.0 x 10^{-3}.

sequence, where the local movements are exploited by the proposed vector quantization scheme.

9.6 Conclusion

In this chapter the applications and coding techniques for image sequence are discussed. In section 9.2 some recent applications of digital moving images are mentioned. The Human Visual System is introduced in section 9.3, where the features of the human visual system are investigated. A three-dimensional bandpass filter is believed to be representative of the visual system. Section 9.4 reviews some of the interframe coding techniques. Three-dimensional subsampling techniques are treated in subsection 9.4.1. These techniques are very simple to implement; however when there is a large movement between the frames, subsampling is not a desirable technique. The unsampled pixels at the receiver are reconstructed by interpolating between the sample elements. The bit rate achieved by these coding techniques is not significant, although a large number of adaptations can be employed. In section 9.4.2 three-dimensional DPCM were explained. The effect of adaptive predictors and quantizers were discussed. These coders are believed to be very simple to implement with very little memory requirement. The bit rate achieved with DPCM coders is about 1-2 bits/pixel/frame. More complicated versions of DPCM systems are discussed in section 9.4.3 and 9.4.4. The effect of motion is considered in these coders in order to improve the predictor. Better results are obtained, but to estimate the motion as a number of pixels is computationally very inefficient. Adaptive DPCM coders are believed to be

very effective where the temporal variation is small.

If the movement in the image sequence is very large the DPCM coders do not reduce the bit rate enough. Thus more complicated techniques were considered in sections 9.4.5, 9.4.6 and 9.4.7. First hybrid transform/DPCM coders are considered for a moving image sequence; adaptivity is also considered. Three-dimensional coders are considered in section 9.4.6, these coders are believed to produce very low bit rates. However, the computational cost for implementation is very high. Thus real time image coding cannot be done with these coders. Also in order for the three-dimensional transformation to be effective several frame stores are required. Thus a large memory storage is required although it is not that important because the price of memory is coming down; large memory chips are now available. In section 9.4.7 a number of techniques are just mentioned.

In section 9.5 a new hybrid coder was developed which incorporates a two-dimensional transform and a vector quantization scheme. The vector quantization scheme is similar to that introduced in chapter eight, but it was performed in the temporal direction. The simulation results show that coding bit rates of R = 0.37 bits/pixel/frame can be achieved. In section 9.5.2 the adaptive coding system is considered where several codebooks of different dimension are employed. The simulation coded images show that better results are obtained with the adaptive coder. Coding bit rates of R = 0.6 bits/pixel/frame are achieved. A more adaptive version of this technique would be to vector quantize the high frequency coefficients with a codebook that is optimised only

for high frequency coefficients. The proposed technique can also be applied to multi-spectral images where the vector-quantization is performed in the spectral domain. Coding of colour images is another application of the proposed technique. Here the vector quantization is performed along the three components of the coded image.

The codebooks used were des igned by the algorithm explained in section 8.4. The codebook could be designed with more complicated algorithms such as those that will take into account the human visual system. Hadamard transform was used because of the computational simplicity. Other transforms, such as cosine transform, can be used which is believed to have a better compression ability than Hadamard transform. The resulting bit rats, R, for the coded images for a codebook of size (N,M) and a discarding ratio C' is given as

$$R = \frac{\log_2 N}{MC'}$$

The above hybrid coding algorithm is believed to be more efficient than the conventional hybrid coders. The proposed algorithm can be implemented in parallel since the algorithm operates on a block of the image sequence at a time.

CHAPTER TEN

CHAPTER TEN

CONCLUSIONS AND COMMENTS

The implementation of digital signal processing algorithms in the transform domain has been investigated for many years. Conventionally the discrete Fourier transform is employed to transform the image data into the frequency domain, where the signal processing algorithms are performed on the transformed samples. Implementation of digital circular convolution is usually performed in the transform domain. The input signal is transformed and then a point by point multiplication of the transformed data with the impulse response of the filter is carried out. One orthogonal transform which has the circular convolution property is the discrete Fourier transform (DFT) which is implemented by fast algorithms such as fast Fourier transform (FFT). The problem with DFT is that complex arithmetic has to be performed for real input data which is usually encountered in digital image processing. Also round off error and the quantization error due to twiddle factors (W_N^k) have prevented implementation of DFT on short length processors.

Recently, a number of new transforms have been introduced in the integer domain. These transforms are defined in a ring or a field of integers are are known as Number theoretic transforms. They are orthogonal and have circular convolution properties. Since arithmetic is performed in a finite field there is no round off error. The only disadvantage of these integer transforms are that the transform length is limited by the processor word length, also a large dynamic range

is required since only integer arithmetic is performed. These limitations have led us to define new orthogonal transforms in other algebraic fields.

In chapter three P-adic transforms are introduced. They are found to have circular convolution properties. Thus they can be used to implement digital convolution. They have a much larger dynamic range compared with those of NTT. Other advantages of P-adic transforms are that rational numbers can be implemented exactly. The arithmetic is performed in a field of integers. No simulation was performed to implement digital convolution by P-adic transform. However, it is believed to be interesting to implement digital convolution by P-adic transform and compare the result with that of DFT, in order to investigate the round off error that is introduced by DFT. Fast transform algorithms, such as polynomial transform, Winograd's algorithm and prime factor algorithm can be used to implement P-adic transforms.

Hardware implementation of P-adic transforms on microprocessors is still to be investigated. The hardware implemented NTT and P-adic transforms are believed to be very efficient compared to discrete Fourier transform. A number of algorithms have recently been developed for hardware implementation of NTT in a field of integers, where arithmetic is performed residue a prime integer such as Fermat or Mersenne primes. The only disadvantage of P-adic transforms is that the input data has to be converted into P-Adic form. However, if data is in P-adic form and the processor is a P-adic processor,that is,it performs hardwired P-adic arithmetic, the transforms are believed to be very efficient.

In some applications complex data has to be transformed. Complex P-adic transform is defined in chapter four, where complex data are transformed to implement complex convolution. Comparison is made between the complex P-adic transform and complex Number theoretic transform (CNTT) as is discussed in chapter four. Finally, P-adic transform in extension fields of P-adic field is defined. Only quadratic extension fields were considered. It is attractive to consider higher extension fields. In chapters three and four only Fermat and Mersenne primes were chosen for the P-adic field. However, other primes with some attractive properties can also be used. P-adic arithmetic is performed over a segmented P-adic field since P-adic representations of numbers is infinite. This segmentation only introduces a limitation over the dynamic range. Finally a P-adic transform over a g-adic ring is introduced, since a g-adic ring consists of the algebriac summation of several P-adic fields, the P-adic transform has to satisfy every P-adic field component of the g-adic ring. Further work is needed to investigate simulation of these extension field transforms for implementation of digital convolution. Only one-dimensional P-adic transforms were treated here. Further study is needed to define a two-dimensional P-adic transform and its implementations.

In chapter five several orthogonal transforms are introduced. Some of these transforms do not have circular convolution properties, thus they cannot be used to implement digital convolution. However, they can be used in some other signal processing applications. One major application of orthogonal transform is in digital image coding. Discrete cosine transform (DCT), Hadamard transform (HT), and finally

polynomial transform are discussed in chapter five. Applications of DCT and HT in image coding are studied in chapter six.

Polynomial transform algorithms are discussed in detail, since any multi-dimensional transform can be implemented by polynomial transform algorithms. Several polynomial transform algorithms are discussed in chapter five. Some of these transforms can be implemented by radix-2 fast Fourier transform. Further work is needed to investigate polynomial transforms and their implementation on general computers. Hardware implementation of polynomial transforms is still to be investigated. In chapter five a new algorithm was introduced to implement discrete cosine transform by using a polynomial transform algorithm.

One major property of some of these orthogonal transforms is their ability to concentrate the energy around some low frequency coefficients. In chapter six this property of the discrete cosine transform is used in image coding systems. Block Transform coding is introduced in chapter six and its adaptive versions are also considered. Conventional block transform coding such as zonal and threshold coding are discussed. Only zonal coding technique was simulated since it is believed to be more efficient than threshold coding because the location of the discarded coefficients are not transmitted as in that of threshold coding. In block transform coding each block is assumed to be independent of the neighbouring blocks and coded independently. However this assumption is not valid because the pixels on the border of adjacent blocks are correlated. In chapter six, section 6.3.2, overlapping blocks are considered

for zonal coding. The simulation results are compared with that of non-overlapping zonal block coding. It is found that on average a lot of the adjacent blocks are highly correlated. Due to a large increase in the computational time of the overlapping block coding system, an adaptive overlapping block coding system is then introduced. In this system only those neighbouring blocks that are highly correlated are overlapped by a five number of pixels. The computation cost is decreased and the inter-block correlation is taken into account. Finally in chapter six, section 6.4, a new zonal-coding technique is introduced. In conventional zonal coding systems a large number of the high frequency coefficients are discarded. In the proposed zonal coding system these high frequency coefficients are vector quantized. That is the bit assignment matrix which consists of a zonal quantizer and a vector quantizer. The proposed system is simulated for only one vector quantizer and no adaptivity is considered. More study is suggested to be done in improving this zonal coding technique and comparing it with a coding system that is a combination of zonal and threshold coding.

In chapter seven predictive coders are introduced. Differential pulse code modulation is explained, where only a first order predictor is considered. In the simulations several quantizers are considered. It is found that a quantizer with Laplacian distribution performs better than other quantizers, such as Gaussian or Gamma distribution quantizers. An entropy coder, such as the Huffman coder, is employed to exploit the non-uniform distribution of the difference signal. Higher order predictors are also discussed. Adaptive quantizers where the

quantization level is dependent upon the amplitude variation in the difference signal are mentioned, but no simulation is given. In section 7.3 a new adaptive DPCM system is introduced where a non-linear function is employed to change the distribution of the difference signal to a highly peaked one, so the entropy coder will be more efficient. The proposed system is simulated and results show that a saving of 0.5 bits per pixel can be achieved compared with those of a conventional DPCM system. Hardware implementation of DPCM systems are still to be investigated since DPCM systems are very simple to implement. Finally, in section 7.4, a short comparison of DPCM systems and transform coders is given.

In chapter eight hybrid/DPCM systems are discussed. These systems combined the attractive features of both transform and DPCM coding systems. In section 8.2 one-dimensional hybrid transform/DPCM are reviewed. Hadamard transform is employed because of its computational simplicity. First order predictor is also employed for prediction of the transformed coefficients. Simulation for the one-dimensional hybrid system is performed in chapter eight, section 8.2.1. No adaptive hybrid systems are simulated although better results can be obtained by adaptive systems. In section 8.3 two-dimensional hybrid systems are discussed. These systems could be configured to exploit inter-block or interframe correlations. In section 8.4 a new hybrid coding system is introduced. This system uses the vector quantization scheme to exploit the inter-row correlation. The vector quantization scheme consists of a codebook where the most probable vectors are stored. The vector quantization algorithm extracts the most probable vector template in the codebook and transmits a codeword to represent the corresponding vector. The

proposed coder is simulated for the test images and it is found that a better result is obtained with the new coder than the one-dimensional hybrid coder. If the proposed coder is hardware implemented the computational time is believed to be much less, since at the receiver the codewords have only to access the codebook in memory to reconstruct the vector templates. In section 8.4.1 formation of the codebook is considered. To find the nearest vector template in the codebook during the coding process a criteria has to be chosen. In our simulation Euclidean distance was chosen. Better criteria measures could have been used which still have to be investigated. Use of several codebooks is also investigated but this is not reported in detail. This one-dimensional hybrid system is believed to be very efficient for applications such as remote sensing and teleconferencing, etc. In section 8.5 two-dimensional hybrid transform/vector quantization is discussed. In this system inter-block correlation is exploited by vector quantization. The proposed system was simulated for the test images at several bit rates. In the proposed systems a fixed number of high frequency coefficients are discarded before the vector quantization scheme is introduced. These coding systems could be made more adaptive by discarding a number of coefficients with respect to the activity in the image line or block.

In chapter nine interframe coding techniques are discussed. Applications are considered where interframe coding is required. Several interframe coding techniques are reviewed in section 9.4. In section 9.5 a new interframe hybrid coder is introduced. In this new hybrid system temporal correlation is exploited by the vector quantization scheme. This is very similar to the

proposed hybrid intrafield coder in chapter eight. The proposed system is simulated for an image sequence and results are compared with other techniques. In section 9.5.2 an adaptive version of the proposed coding technique is discussed. Here use of several codebooks is suggested to exploit the temporal changes due to movement of the objects within the image sequence,as well as panning. The simulation results are reported and it is found that the adaptive coder performs as well as the three-dimensional transform coders. Adaptivity is considered by using several codebooks. It is possible to base the adaptivity upon the design of the codebooks or zonal discarding of the high frequency coefficients.

A detailed comparison of the proposed technique with other interframe coders is still needed. Hardware implementation was only superficially discussed; detailed consideration is needed for its hardware implementation. Only Hadamard transform was considered, however other transforms can also be used. The number of codebooks can be increased such that the low frequency and high frequency coefficients are coded by different codebooks, where each codebook is optimised for a particular region.

The vector quantization techniques have not yet been applied to medical image processing. For example, one- or two-dimensional projections obtained in Computer-Aided Tomographic images can be coded by the proposed vector quantization algorithms. Finally, the vector quantization algorithms still remain to be applied to the colour TV images as well as the filtered images obtained in sub-band coding.

APPENDICES

"A"

APPENDIX A

INTRODUCTION TO ELEMENTARY NUMBER THEORY

In this appendix the basic concepts of number theory and modular arithmetic are presented.

1. DIVISIBILITY OF INTEGERS

Considering two integers a and b, with b positive, the division of a by b is defined by:

$$a = bq + r \qquad 0 \leqslant r \leqslant b \tag{A.1}$$

where q is called the quotient and r is called the remainder. If the latter is equal to zero, then b is said to be a divisor of a, and this operation is denoted by $b \mid a$. The integer a is prime if it has no other divisors than 1 and itself, otherwise a is composite.

The fundamental theorem of arithmetic states that any composite number, a, can be uniquely factorised as:

$$a = \prod_i P_i^{m_i} \tag{A.2}$$

where P_i is a prime number and m_i is a positive integer.

2. CONGRUENCES AND RESIDUES

If two integers, a_1 and a_2, give the same remainder when divided by an integer b (the modulus), i.e.

$$\begin{aligned} a_1 &= bq_1 + r \\ a_2 &= bq_2 + r \end{aligned} \tag{A.3}$$

then a_1 and a_2 are said to be "congruent modulo b", and this is denoted as:

$$a_1 \equiv a_2 \quad \text{modulo } b \tag{A.4}$$

Alternatively, this can be expressed as:

$$b|(a_1 - a_2) \tag{A.5}$$

The remainder, r, is called the residue and is our main interest. It can be expressed as:

$$r = \langle a \rangle_b \tag{A.6}$$

As can be seen from eqn. (A.3), addition, subtraction and multiplication can be performed directly on residues, or:

$$\langle a_1 + a_2 \rangle = \langle\langle a_1 \rangle + \langle a_2 \rangle\rangle \tag{A.7a}$$

$$\langle a_1 - a_2 \rangle = \langle\langle a_1 \rangle - \langle a_2 \rangle\rangle \tag{A.7b}$$

$$\langle a_1 \cdot a_2 \rangle = \langle\langle a_1 \rangle \cdot \langle a_2 \rangle\rangle \tag{A.7c}$$

3. GROUPS AND ABELIAN GROUPS

A group, G, is a non empty set of elements which, with a binary operation $\odot$, satisfies the following postulates:

(i) Closures: For every a and b in G, $a \odot b$ is also in G.

(ii) Associativity: For every a,b,c in G, $(a \odot b) \odot c = a \odot (b \odot c)$

(iii) Identity: There is a unique element e in G, called the identity element, such that $a \odot e = e \odot a = a$ for all a's in G.

(iv) Inverse: For every a in G there is a unique inverse b in G, such that $a \odot b = b \odot a = e$.

A group G is called an Abelian group if every pair of elements commute, i.e. $a \odot b = b \odot a$ for all a's and b's in G.

4. RINGS AND FIELDS

Rings and Fields are sets of integers whose elements obey certain properties.

A ring, Z, is defined if its elements adhere to the following conditions:

(i) a.b is a legitimate element of the set.

(ii) (Z,+) is an Abelian group.

(iii) Associativity.

(iv) Distributivity of (x,.1).

If multiplication is commutative, then Z is a commutative ring.

In a ring, $Z_M = \{0,1,2,\ldots, M-1\}$, all integers are congruent modulo M, to some integer in Z_M. If in Z_M, each integer has a unique multiplicative inverse, then the ring becomes a field. It can be shown [A16] that a ring is a field if and only if M is a prime number.

5. MODULAR ARITHMETIC

The following arithmetic operations can be performed with modular arithmetic:

(a) Addition: Arithmetic modulo some arbitrary modulus can be performed as ordinary arithmetic, followed by a division of the result by the modulus. The remainder is the answer.

Example: $7 + 14 = 21 = 4 \pmod{17}$

(b) Negation : To negate a number, a, it is subtracted from the modulus M as follows: $-a = -a + M \pmod{M}$.

Example: $-8 = -8 + 17 = 9 \pmod{17}$

(c) Subtraction: This consists of negating a number as in (b) and then adding it according to (a).

Example: $7 - 8 = 7 + (-8) = 7 + 9 = 16 \pmod{17}$

(d) Multiplicative Inverse: A multiplicative inverse of an integer p exists in Z_M if and only if p and M are relatively prime. The inverse, p^{-1}, is then given by:

$$p.p^{-1} = 1 \pmod{M}$$

Example: $4^{-1} = 13 \pmod{17}$

Since

$$4 \times 13 = 52 = 1 \pmod{17}$$

From (a), (b), (c) and (d) it can be deduced that due to the nature of modular arithmetic, numbers do not have sizes or magnitudes, unless the quotient used in calculating the number modulo M is known.

6. CHINESE REMAINDER THEOREM

One of the techniques often used is to map an M-point one-dimensional data sequence x_n into a K-dimensional data array. This is done by noting that if n is defined modulo M, with $n = 0,\ldots,M-1$, we can redefine n by the Chinese theorem as

$$n \equiv \sum_{i=1}^{K} \left(\frac{M}{m_i}\right) n_i T_i \qquad \text{modulo } M,$$

where the index n_i along dimension i takes the values $0,\ldots, m_i-1$. This mapping, which is possible only when M is the product of relatively prime factors m_i, is very important for the computation of discrete Fourier transforms and convolutions.

7. FERMAT EULER'S THEOREM AND EULER'S TOTIENT FUNCTION

Euler's totient function is defined as the number (M) that gives the number of positive integers less than or equal to M that are relatively prime to M. With M = P a prime integer then $\varphi(P) = P-1$. If $M = P^c$, c an integer, P a prime, then $\varphi(P^c) = P^{c-1}(P-1)$. Also $\varphi(a.b) = \varphi(a)\,\varphi(b)$, where a and b are just two integers.

Fermat-Euler's theorem is given by

$$a^{\varphi(M)} \equiv 1 \quad \text{modulo } M \tag{A.8}$$

where (a,M) = 1 which denotes a and M are relatively prime. a is said to belong to exponent (M) modulo m, a is also known as primitive roots in modulo M. The number of primitive roots is given by $\varphi(\varphi(M))$. If M = P is a prime then expression (A.8) becomes

$$a_{\varphi}^{P-1} \equiv 1 \quad \text{modulo } P \tag{A.9}$$

a_φ is the primitive roots which will generate all the elements in $I_P = (0,1,2,\ldots P-1)$. a_φ is said to be of order (P-1). It is possible to obtain roots of order r from primitive roots a_φ.

$$a_r^r \equiv 1 \quad \text{modulo } P \tag{A.10}$$

if $r|P-1$. Thus the order of a_r, r must divide the maximum order P-1. The roots a_r of order r can be evaluated from

primitive root given by:

$$a_r = a_Q^{\frac{r}{p-1}} \qquad (A.11)$$

8. QUADRATIC RESIDUES

Quadratic equations $x^2 \equiv a \pmod M$ is said to be a quadratic residue or a non-residue modulo M if it is soluble or non-soluble respectively. If M = P denotes an odd prime and (a,P) = 1, the Legendre symbol $(\frac{a}{P})$ is defined to be 1 if a is a quadratic residue -1, if a is a quadratic non-residue modulo P. Then

$$(\frac{a}{P}) = a^{(P-1)/2} \pmod P \qquad (A.12)$$

$(\frac{a}{P}) = +1$ if a is a quadratic residue of P

$(\frac{a}{P}) = -1$ if a is a quadratic non-residue of P

9. EXTENSION FIELDS OF INTEGER FIELDS I_P

Let I_p represent an integer field of P elements, $I_p = (0,1,2,\ldots P-1)$. Extension fields of I_p can be represented by $I_p(\sqrt{a})$, where a is a quadratic non-residue in I_p. So $x^2 \equiv a \pmod P$ has no solution. The elements of the extension fields are represented by

$$A = \alpha + \sqrt{a}\,\beta \qquad (A.13)$$

where

$$A, \sqrt{a} \in I_p(\sqrt{a}) \quad \text{and} \quad \alpha, \beta \in I_p$$

If a = -1 then complex extension fields are considered.
If a is an integer value then quadratic extension fields are considered.

10. RESIDUE POLYNOMIALS

The theory of residue polynomials is closely related to the theory of integer residue classes. A polynomial P(z) divides a polynomial H(z) if a polynomial D(z) can be found such that H(z) = P(z) D(z). H(z) is said to be irreducible if its only divisors are of degree equal to zero. If P(z) is not a divisor of H(z), the division of H(z) by P(z) will produce a residue R(z),

$$H(z) = P(z)\ D(z) + R(z)$$

where the degree of R(z) is less than the degree of P(z). All polynomials having the same residue when divided by P(z) are said to be congruent modulu P(z) and the relation is denoted by

$$R(z) \equiv H(z) \qquad \text{modulo } P(z)$$

All the concepts in integer fields and rings can be applied to polynomial residues.

APPENDICES

"B"

APPENDIX B

INTRODUCTION TO P-ADIC FIELD

In this appendix P-adic fields are introduced and their properties are investigated.

1. P-adic Sequence

Let P be a fixed prime. A sequence of integers $\langle x_n \rangle$ with the property that $x_{n-1} \equiv x_n \bmod P^n$, $n = 1,2,3,\ldots$, is called a P-adic sequence of α.

Example 1: Let $P = 5$, $\alpha = 221$

$\langle x_n \rangle = 1, 21, 96, 221, 221, 221, \ldots$

Example 2: Let $P = 10$, $\alpha = -222$

P-adic representation is

$\langle x_n \rangle = 8, 78, 778, 9778, 99778, \ldots$

2. Canonical Representation of P-adic Sequence

Consider a P-adic sequence $\langle x_n \rangle$ and let $\hat{x}_n$ be the unique rational integer sequence such that $\hat{x}_n \equiv \hat{x}_n \bmod P^{n+1}$ and $0 \leqslant \hat{x}_n \leqslant P^{n+1}$ for $n = 1,2,3,\ldots$, then $\langle x_n \rangle$ is the canonical sequence that represents α.

Example 1: Let $\gamma = \langle Z_n \rangle$, where $\langle Z_n \rangle = -1,-1,-1,\ldots$ the canonical sequence that determines γ with $P = 5$ is given by $\langle Z_n \rangle = 4, 24, 124, 624, \ldots$

3. P-adic Series

We can write the canonical sequence associated with α, an integer in series form given as

$$\alpha = \sum_{i=0}^{\infty} a_i P^i \quad \text{for } 0 \leqslant a_i \leqslant P \pmod{P} \qquad \text{(B. 1)}$$

To investigate eqn. (B.1) in more detail, consider a P-adic sequence representation of an integer α, $\hat{x}_n \equiv \hat{x}_{n-1} \bmod P^n$, $\hat{x}_n = \hat{x}_{n-1} + a_n P^n$, and because $0 \leq \hat{x}_n < P^{n+1}$, the integer a_n must satisfy the inequality $0 \leq a_n < P$.

So

$$\begin{aligned} \hat{x}_0 &= a_0 \\ \hat{x}_1 &= a_0 + a_1 P \\ \hat{x}_2 &= a_0 + a_1 P + a_2 P^2 \\ &\vdots \\ \hat{x}_n &= a_0 + a_1 P + a_2 P^2 + \ldots + a_n P^n \end{aligned} \tag{B.2}$$

Then a P-adic integer is represented as

$$\alpha = \sum_{i=0}^{\infty} a_i P^j \qquad 0 \leq a_i < P \tag{B.3}$$

where its sequence of partial sum is a canonical sequence for some P-adic integers.The P-adic integers have finite series representation.

The P-adic series can be evaluated from the canonical P-adic representation given by relation

$$a_0 = \hat{x}_0 \quad , \quad a_n = \frac{(\hat{x}_n - \hat{x}_{n-1})}{P^n} \tag{B.4}$$

for n = 1,2,3,...

where $\langle \hat{x}_n \rangle$ is the canonical representation of α and a_i its P-adic series coefficients.

Example 1: $\alpha = 221$, $P = 5$

$\langle \hat{x}_n \rangle = 1,21,96,221,\ldots$

$a_o = 1$, $a_1 = \frac{21-1}{5} = 4$, $a_2 = \frac{96-21}{25} = 3$

$a_3 = \frac{221-96}{125} = 1$, $a^4 = 0$, $a_5 = 0$, ...

$221 = 1 + 4.5 + 3.5^2 + 1.5^3$

4. P-Adic Series Representation of Rational Numbers

Let $\alpha = a/b$ be a non-zero rational number such that b#0. Then it is clear that α can be expressed in a unique way as

$$\alpha = P^{\pm n} \frac{c}{d}$$

where P is a given prime number and c, d are integers such that P does not divide c and d. Then the P-adic expansion of α is obtained by the following algorithm

(a) Find n such that $\beta = c/d \; P^n$.

(b) Solve the congruence $dx \equiv 1 \pmod{P}$. If x_n is a solution, then $a_n = c\, x_n \pmod{P}$.

(c) Set $\gamma = \beta - a_n P^n$. If $\gamma = 0$, set $a_i = 0$ for $i > n$ and stop, otherwise set $\beta = \gamma$ and continue the procedure from step (b).

Example:

1/2 and 1/5, respectively have the five-adic expansion

$$\frac{1}{3} = 0.231313 \ldots \text{ and } \frac{1}{15} = 2.313\ldots$$

5. Segmented P-adic Codes or Hensel Codes

The P-adic series representation of rational numbers is infinite. However, to be able to do P-adic arithmetic a segmented P-adic field has to be introduced.

Let α be a rational number with P-adic expansion, $a_n, a_{-n+1}, \ldots, a_{-1}. a_o \ldots a_K$, where $r = n + k + 1$. This is called Hensel code of α and is denoted by $H(p,r,\alpha)$. For convenience $H(p,r,\alpha)$ is denoted as an ordered pair in the mantissa-exponent form thus: (m_α, e_α). Since we keep the length of $H(p,r,\alpha)$ constant (r digits), e_α is permitted to be zero or to be only negative values. When $e_\alpha = -n$, the radix point is placed n digits to the right of the left most digit of m . Accordingly, the mantissa is an integer and we can always assume that m is of the form

$$m_\alpha = . a_o, a_1 m ,,, a_{r-1}$$

and

$$e_\alpha \leq 0$$

Example:

$$H(5,4, \frac{7}{15}) = (0.43.3, -1)$$

$$H(5,4, \frac{15}{7}) = (0.0402, 0)$$

$$H(5,4,15) = (0,0300, 0)$$

Conversion of Hensel codes into rational numbers is possible by several algorithms. One method is by using a look-up table where all the rational numbers are stored. Another method is, since every rational number (a/b) is represented in the form $(a.b^{-1})$ Mod p^r, it is possible to determine a as

well as b if some common multiple of all the denominators involved in a given algorithm is known. For instance, if we assume kb is known, then the rational number corresponding to $(a.b^{-1})$ Mod P^r is

$$[\frac{1}{kb} \cdot (ab^{-1}.kb)] \text{ Mod } P^r$$

Therefore, as long as $(ab^{-1}.kb)$ and kb are representable uniquely as H(p,r) codes, the conversion is straightforward.

6. Arithmetic in P-adic Field

(a) Addition and subtraction

Let A and B be P-adic numbers, and let their P-adic series be

$$A = \sum_{n=-m}^{\infty} a_n P^n \quad , \quad B = \sum_{n=-m}^{\infty} b_n P^n \qquad \text{(B.5)}$$

$$\text{for } 0 \leq a_n, b_n < P$$

It can be shown that

$$(A \pm B) = \sum_{n=-m}^{n} (a_n \pm b_n) P^n \qquad \text{(B.6)}$$

(b) Multiplication and division

Multiplication of two P-adic numbers

$$A = \sum_{n=-f}^{\infty} a_n P^n \quad \text{and} \quad B = \sum_{n=-\ell}^{\infty} a_n P^n$$

is given by

$$A.B = \sum_{n=-f-\ell}^{\infty} u_n P^n \qquad \text{(B.7)}$$

$$\text{for } 0 \leq u_n < P$$

where u_n is evaluated by multiplying the series term by term and rearranging the terms. Division is more detailed as given in [A42].

7. Units in P-adic Field

(a) Definition of unit

An element of a ring is called a unit if it has a multiplicative inverse. In the ring of integers, Z_m the only units are 1 and -1. The units in a field or ring of integers Z_m is given by $\varphi(Z_m)$ where φ is the Euler function.

(b) Units in P-adic Field

(i) A P-adic integer $\alpha = \langle \hat{x}_n \rangle$ is a unit if and only if $\hat{x}_o \neq 0 \text{ Mod } P$ and since α has a P-adic series representation $\alpha = \sum_{i=0}^{\infty} a_i P^i$, α is a unit if and only if $a_o \neq 0$.

(ii) A rational number of the form $\frac{r}{s}$, $(r,s) = 1$, is a unit if and only if $(S,P) = 1$, $(r,P) = 1$.

(iii) Every P-adic integer that is not zero has a unique representation in the form $\alpha = P^m C$, where C is a unit and m is a non-negative integer.

Proof for the above statements is given in [A42].

8. The Quadratic Extension Fields of Q_p

A quadratic extension field of Q_p is obtained by adjoining to Q_p a root of some quadratic equation with coefficients in Q_p. Without loss of generality, this equation has the form

$$x^2 - d = 0 \qquad (B.8)$$

where $d \neq 0$ is a P-adic number which is not the square of a P-adic number. As a consequence, the equation (B.8) cannot be

solved in Q_p itself. On denoting a formal solution of (B.8) by $\sqrt{d}$, the quadratic extension field, K_p say, is then derived from Q_p by adjoining $\sqrt{d}$,

$$K_p = Q_p(\sqrt{d}) \tag{B.9}$$

The elements of K_p can be written as

$$A = a + b\sqrt{d}$$

where

$$a,b \in Q_p .$$

9. Definition of g-adic Field

Any rational number can be represented as a g-adic series;

$$\alpha = a_{-f} g^{-f} + a_{-f+1} g^{-f+1} + \ldots + a_{-1} g^{-1} + a_o + a_1 g + a_2 g^2 + \ldots \tag{B.10}$$

where the coefficients a_n are digits $0,1,\ldots, g-1$.

Thus α has the canonic series

$$\alpha = \sum_{n=-f}^{\infty} a_n g^n \tag{B.11}$$

Any g-adic number can be decomposed into its distinct P-adic fields, so

$$Q_g = Q_{P_1} \oplus Q_{P_2} \oplus \ldots \oplus Q_{P_n} \tag{B.12}$$

where

$$g = P_1 P_2 \ldots P_n$$

10. Newton's Approximation Method in P-adic Fields

Newton's method for obtaining the real zeros of a real, values function f(x) is a well know iterative method which generates the approximations;

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)} \qquad \text{for } i \geq 0 \qquad (B.13)$$

where x_o is an initial approximation to a zero of f(x), and $f'(x) \neq 0$ is computable.

Consider $f(x) \in Z_p[x]$ to be a polynomial with coefficients in the ring of P-adic integers, if the first coefficient of a P-adic sequence is known A_o, then the P-adic sequence $\{A_n\}$ defined by

$$A_n = A_{n-1} - \frac{f(A_{n-1})}{f'(A_{n-1})} \qquad (B.14)$$

where $0 \leq A_n < P^{n+1}$

is a P-adic sequence of the form

$$A_n = A_{n-1} + a_n P^n \qquad (B.15)$$

where $a_n \in I_P$.

It is possible to get expression in (B.14) in terms of P-adic series. Suppose f(n) = 0 has a P-adic integral root $\alpha = a_o + a_1 P + a_2 P^2 + \ldots$ where $a_i \in I_P$.

Let $A_n = a_o + a_1 P + \ldots a_n P^n$, then clearly α is a solution of f(x) = 0 in Q_P is and only if

$$f(\alpha) = 0 \ (\text{Mod } P^n) \qquad (n=1,2,3,\ldots)$$

if and only if

$$f(A_n) = 0 \ (\text{Mod } P^{n+1}) \qquad (n=0,1,2,\ldots)$$

By substituting (B.15) in (B.14)

we get

$$A_{n-1} + a_n P^n = A_n = A_{n-1} - \frac{f(A_{n-1})}{f'(A_{n-1})}$$

or

$$a_n P^n = - \frac{f(A_{n-1})}{f'(A_{n-1})} \quad (\text{Mod } P^{n+1}) \qquad \text{for } (n \geq 1)$$

Since $f(A_{n-1})$ can be represented as

$$f(A_{n-1}) = h_{n-1} P^n \qquad (\text{Mod } P^{n+1}) \qquad \text{for } (n > 1)$$

$$a_n f'(A_{n-1}) + h_{n-1} \equiv 0 \quad (\text{Mod } P) \qquad \text{for } (n > 1)$$

Example:

Solve the equation $f(n) = x^2 + 1 = 0$ in Q_{13}. Since -1 is a quadratic residue modulo 13 with $x_0 = 5$ or $x_o = 8$ then choosing $A_o = a_o = 5$, we can calculate the other P-adic series coefficients from Newton's approximation method. Since $f(n) = x^2 + 1$ and $f'(n) = 2x$, $f(A_o) = 26$, expression $f(A_{n-1}) = h_{n-1} P^n$ (Mod P^{n+1}) becomes

$$26 \equiv 13\, h_o \quad (\text{Mod } 169)$$

whose solution in I_{13} is $h_o = 2$. Then expression $a_n f'(A_{n-1}) + h_{n-1} \equiv 0$ (Mod P) becomes

$$10\, a_1 + 2 \equiv 0 \quad (\text{Mod } 13)$$

So $a_1 = 5$, the solution can be shown to be $\alpha = .5510 \ldots$, if Newton's approximation is iterated.

REFERENCES

"A"

REFERENCES

[A1] G.T. Herman, Special Issue on Computerized Tomography, Proc. IEEE, March 1983, pp.291-446.

[A2] G.T Herman, Image Reconstruction from Projections, Topics in Appl. Phys., vol. 32, 1979, Springer-Verlag.

[A3] G.T. Herman, Image Reconstruction from Projections: The Foundations of Computerized Tomography, Academic Press, New York, 1981.

[A4] Special Issue on Acoustic Imaging, Proc. IEEE, April 1979, pp. 452-664.

[A5] A.K. Louis, and F. Natterer, Mathematical Problems of Computerized Tomography, Proc. IEEE, vol. 71, no. 3, March 1983.

[A6] O.A. Landgrebe, Analysis Technology for Land Remote Sensing, Proc. IEEE, vol. 69, no. 5, May, 1983, pp.628-642.

[A7] A. Rosenfeld, Digital Picture Analysis, Topics in Appl. Phys., vol. 11, 1976, Springer-Verlag, pp. 5-63.

[A8] Special Issue on Robotics, Proc. IEEE, vol. 17, no. 7, July 1983, pp.787-911.

[A9] R.H. Taylor and D.D. Grossman, An Integrated Robot System Arichitecture, Proc. IEEE, vol. 17, no. 7, pp.842-856.

[A10] D.H. Ballard and C.M. Brown, Computer Vision, 1982, Prentice-Hall, Inc. Englewood Cliffs, New Jersey.

[A11] H.J. Nussbaumer, Fast Fourier Transform and Convolution Algorithms, Springer-Verlag, 1982.

[A12] D.F. Elliott and K.R. Ras, Fast Transforms, Algorithms, Analyses Applications, Academic Press, 1982.

[A13] R.O. Harger, Synthetic Aperture Radar Systems Theory and Design, Academic Press, 1970.

[A14] W.W. Pratt, Digital Image Processing, John Wiley & Sons, 1978.

[A15] A.C. Agarwal and C.S. Burrus, Number Theoretic Transforms to Implement Fast Digital Convolution, Proc. IEEE, vol. 63, no. 4, April 1984.

[A16] S.H. McClellan and C.M. Rader, Number Theory in Digital Signal Processing, Prentice-Hall Signal Processing Series, 1979.

[A17] C.M. Rader, Discrete Convolutions via Mersenne Transforms, IEEE Transactions on Computers, vol.C-21, no. 12, December 1972.

[A18] R.C. Agarwal and C.S. Burrus, Fast Convolution using Fermat Number Transforms with Applications to Digital Filtering, IEEE Trans. ASSP-22, 87-97, 1974.

[A19] S. Winograd, On Computing the Discrete Fourier Transform, Math. Comput. 32, 175-199(1978).

[A20] D.P. Kolba and T.W. Parks, A Prime Factor FFT Algorithm using High-Speed Convolution, IEEE Trans. ASSP-25, 90-103, 1977.

[A21] I.S. Reed and T.K. Truong, The Use of Finite Fields to Compute Convolutions, IEEE Trans. on Information Theory, vol. 21, 1975, pp.208-213.

[A22] I.S. Reed and T.K. Truong, Convolutions over Residue Classes of Quadratic Integers, IEEE Trans. on Information Theory, vol. 22, no. 4, 1976, pp.468-475.

[A23] H.J. Nussbaumer, Complex Convolutions via Fermat Number Transform, IBM, J. Res. Dev. 20, 282-284, 1976.

[A24] I.S. Reed and T.K. Truong, Complex Integer Convolutions over a Direct Sum of Galois Fields, IEEE Trans. on Information Theory, vol. 21, no. 6, November 1976, pp.657-661.

[A25] E. Dubois and A.N. Venelsanopoulos, The Generalized Discrete Fourier Transform in Rings of Algebraic Integers, IEEE Trans. on ASSP, vol. 28, no. 2, April 1980, pp.169-175.

[A26] I.S. Reed et al., Image Processing by Transform over a Finite Field, IEEE Trans. on Computers, vol. C-26, no. 9, September 1977.

[A27] H.J. Nussbaumer, Digital Filtering using Complex Mersenne Transforms, IBM J. Res. Dev. 20, 498-504.

[A28] H.J. Nussbaumer, Digital Filtering using Pseudo Fermat Number Transforms, IEEE Trans. ASSP-26, 79-83, 1977.

[A29] K. Hensel, Theorie der algebraischen Zahlen, Texbner, Leipzig, 1908.

[A30] G. Bachman, Introduction to P-adic Numbers and Valuation Theory, Academic Press, New York, 1964.

[A31] E.V. Krishnamurthy, Matrix Processors using P-adic Arithmetic for Exact Linear Computations, IEEE Trans. Comput. C-26, 1977, pp. 633-639.

[A32] E.V. Krishnamurthy, T.M. Rao and K. Subramanian, Finite Segment P-adic Number Systems with Applications to Exact Computation, Proc. Indian Acad. Sci. Sect. A81, 1975, pp. 58-79.

[A33] E.V. Krishnamurthy et al., P-adic Arithmetic Procedures for Exact Matrix Computations, Proc. Indian Acad. Sci., Sect. A82, 1975, pp.165-175.

[A34] N.M. Nasrabadi and R.A. King, Fast Digital Convolution using P-adic Transforms, Electr. Lett. vol. 19, no. 7, 31 March 1983.

[A35] I. Niven and H.S. Zuckerman, An Introduction to the Theory of Numbers, John Wiley & Sons, Inc., 3rd ed., 1972.

[A36] I.S. Reed and T.K.Truong, Fast Mersenne-Prime Transforms for Digital Filtering, Proc. IEEE, vol. 125, no. 5, May 1978, pp.433-440.

[A37] K.Y. Liu, I.S. Reed and T.K. Truong, Fast Number-Theoretic Transforms for Digital Filtering, Electr. Lett., vol. 12, no. 24, November 1976.

[A38] C.M. Rader, Discrete Fourier Transofrms when the Number of Data Samples is Prime, Proc. IEEE 56, 1968, pp.1107-1108.

[A39] R.C. Agarwal and J.W. Cooley, New Algorithms for Digital Convolution, IEEE Trans. ASSP-25, 1977, pp.392-410.

[A40] N.M. Nasrabadi and R.A. King, Complex Number Theoretic Transform in P-adic Field, IEEE 1984, International Conference on Acoustics, Speech, and Signal Processing.

[A41] G.H. Hardy and E.M. Wright, An Introduction to the Theory of Numbers, Oxford University Press, 5th ed., 1979.

[A42] K. Mahler, P-adic Numbers and their Functions, 2nd ed., Cambridge University Press, 1980.

[A43] N. Ahmed, T. Natarjan and K.A. Rao, Discrete Cosine Transform, IEEE Trans. on Computers, January 1974, pp.90-93.

[A44] J. Makhoul, A Fast Cosine Transform in One and Two Dimensions, IEEE Trans, ASSP-28, 1980, pp.27-33.

[A45] N.M. Nasrabadi and R.A. King,Computationally Efficient Discrete Cosine Transform Algorithm, Electr. Lett., vol. 19, no. 1, 6 January 1983.

[A46] H.J. Nussbaumer and P. Quandalle, Fast Computation of Discrete Fourier Transforms using Polynomial Transforms, IEEE Trans., ASSP-27, 1979, pp.169-181.

[A47] H.J. Nussbaumer and P. Quandalle, Computation of Convolutions and Discrete Fourier Transforms by Polynomial Transform, IBM J. Res. Dev. 22, 1978, pp.134-144.

[A48] H.J. Nussbaumer, DFT Computation by Fast Polynomial Transform Algorithms, Electr. Lett. 15, 1979, pp.701-702.

[A49] B. Arambepola and P.J.W. Rayner, Discrete Transform over Polynomial Rings with Applications in Computing Multidimensional Convolutions, IEEE Trans. on ASSP-28, no. 4, August 1980.

[A50] H.J. Nussbaumer, Digital Filtering using Polynomial Transforms, Electr. Lett. 13, 1977, pp.386-387.

[A51] I.S. Reed, H.M. Shao and T.K. Truong, Fast Polynomial Transform and its Implementation by Computer, IEEE Proc., vol. 128, pt. E, no. 1, March 1981.

[A52] B. Arambepola and P.J.W. Rayner, Efficient Transforms for Multimensional Convolutions, Electr. Letts., vol. 15, no. 6, March 1979, pp.189-190.

[A53] B. Arambepola and P.J.W. Rayner, Multidimensional Fast Fourier Transform Algorithms, Electr. Lett., vol. 15, no. 13, June 1979, pp.382-383.

[A54] H.J. Nussbaumer, New Polynomial Transform Algorithms for Multidimensional DFT's and Convolutions, IEEE Trans., vol. ASSP-29, no. 1, February 1981, pp.761-783.

[A55] H.J. Nussbaumer, Inverse Polynomial Transform Algorithms for DFT's and Convolutions, IEEE 1981, Int. Conf. of ASSP.

REFERENCES

"B"

REFERENCES

[1B] P.A. Wintz, Transform Picture Coding, Proc. IEEE, vol. 60, no. 7, July 1972, pp. 809-820.

[2B] A. Habibi, P.A. Wintz, Image Coding by Linear Transformation and Block Quantization, IEEE Trans. on Communication Tech., vol. COM-19, no. 1, February 1971.

[3B] D.J. Granrath, The Role of Human Visual Models in Image Processing, Proc. IEEE, vol. 69, no. 5, May 1981, pp. 552-561.

[4B] C.F. Hall, E.L. Hall, A Nonlinear Model for the Spatial Characteristics of the Human Visual System, IEEE Trans. on Systems, Man.and Cybernetics, March 1977, pp. 161-170.

[5B] G.B. Anderson, T.S. Huang, Piecewise Fourier Transformation for Picture Bandwidth Compression, IEEE Trans on Communication Tech., vol. COM-19, no. 2, April, 1971, pp.133-140.

[6B] M. Ghanbari, D.E. Pearson, Fast Cosine Transform Implementation for Television Signals, Proc. IEEE, vol. 129, Pt. F, no. 1, February 1982, pp. 59-68.

[7B] M. Tasto, P.A. Wintz, Image Coding by Adaptive Block Quantization, IEEE Trans. on Communication Technology, vol. COM-19, no. 6, December 1971, pp. 957-971.

[8B] W.K. Pratt, H.C. Andrews, Hadamard Transform Image Coding, Proc. IEEE, 1969, 57, pp. 58-68.

[9B] W.K. Pratt, W.H. Chem, L.R. Welch, Slant Transform Image Coding, IEEE Trans. COM.-22, no. 8, pp.1075-1093.

[10B] N. Ahmed, T. Natarjan, K.R. Rao, Discrete Cosine Transform, IEEE Trans. on Computers, January 1974, pp. 90-93.

[11B] J.K. Wu, R.E. Burge, Adaptive Bit Allocation for Image Compression, Computer Graphics and Image processing, 19, 1982, pp. 392-400.

[12B] J. Max, Quantizing for Minimum Distortion, IRE Trans. on Information Theory, March 1960, pp. 7-12.

[13B] M.D. Paez, T.H. Glisson, Minimum Mean-Squared-Error Quantization in Speech PCM and DPCM Systems, IEEE Trans. on Communications, April 1972, pp. 225-230.

[14B] D.A. Huffman, A Method for the Construction of Minimum Redundancy Codes, Proc. IRE 40, no. 9, 1098-1101.

[15B] D.A. Hall, Computer Image Processing and Recognition, Academic Press, New York, 1979.

[16B] W.K. Pratt, Digital Image Processing, Wiley, New York, 1978.

[17B] N. Abrahamson, Information Theory and Coding, McGraw-Hill, New York, 1963.

[18B] Y. Linde, A. Buzo, R. Gray, An Algorithm for Vector Quantizer Design, IEEE Trans. on Communication, vol. COM-28, no. 1, January 1980.

[19B] W.H. Chen, C.H. Smith, Adaptive Coding of Monochrome and Colour Images, IEEE Trans. on Communications, vol. COM-25, no. 11, November 1977.

[20B] R.E. Burge, J.K. Wu, An Adaptive Transform Image Data Compression Scheme Incorporating Pattern Recognition Procedures, IEEE 1980, Int. Conf. on Pattern Recognition and Image Processing.

[21B] J.B. O'Neal, Jr., Predictive Quantizing Systems for the Transmission of Television Signals, The Bell System Technical Journal, May/June 1966, pp. 689-721.

[22B] W. Zschunke, DPCM Picture Coding with Adaptive Prediction, IEEE Trans. on Communications, vol. COM-25, no. 11, November 1977.

[23B] B.G. Haskell, Entropy Measurement for Non-adaptive and Adaptive Frame-to-Frame Linear Predictive Coding of Video Telephone Signals. Bell System Tech. J. 54, 1975, pp. 1155.

[24B] L.A. ZADEH, Outline of a New Approach to the Analysis of Complex Systems and Decision Processes, IEEE Trans. SMC-3, 1973, pp. 28-44.

[25B] D.J. Connor, R.C. Brainard, J.O. Limb, Intraframe Coding for Picture Transmission, Proc. IEEE, vol. 60, no. 7, July 1972, pp. 779-791.

[26B] A.N. Netravali, On Quantizers for DPCM Coding of Picture Signals, IEEE Trans. Inform. Theory, vol. IT-23, pp. 360-370.

[27B] P. Pirsch, L. Stenger, Acta Electronic, vol. 19, 1977, pp. 277-287.

[28B] L.A. Zadeh, Outline of a New Approach to the Analysis of Complex Systems and Decision Processes, IEEE Trans. Syst. Man. Cybern. vol. SMC-3, January 1973, pp. 28-44.

[29B] S.K. Pal, R.A. King, Image Enhancement using Smoothing with Fuzzy Sets, IEEE Trans. Syst. Man. Cybern., vol. SMC-11, no. 7, July 1981, pp. 494-501.

[30B] A. Habib, Comparison of nth Order DPCM Encoder with Linear Transformations and Block Quantization Techniques, IEEE Trans. on Communication Tech., vol. COM-19, no. 6, December 1971, pp. 948-956.

[31B] N.M. Nasrabadi, R.A. King, Entropy-Coded Hybrid Differential Pulse-Code Modulation, Electron. Lett., vol. 19, no. 2, 20 January 1983, p. 83.

[32B] S.K. Pal, R.A. King, Image Enhancement using Fuzzy set, Electr. Letts., 8 May 1980, vol. 16, no. 10, pp. 376-379

[33B] A. Habibi, Hybrid Coding of Pictorial Data, IEEE Trans. Comm., vol. COM-22, Sept. 1975, pp. 614-624.

[34B] N.M. Nasrabadi, R.A. King, Image Coding using Vector Quantization in the Transform Domain, Pattern Recognition Lett., no. 1, July 1983, pp. 323-329.

[35B] K.R. Rao, M.A. Narasimhan, W.J. Gorzinski, Processing Image Data by Hybrid Techniques, IEEE Trans. on Systems, Man. and Cyber., vol. SMC-7, no. 10, Oct. 1977, pp. 728-734.

[36B] A. Buzo, A.H. Gray, Jr., R.M. Gray, Speech Coding Based upon Vector Quantization, IEEE Trans. on Acoustics, Speech and Signal Processing, vol. ASSP-28, no. 5, Oct. 1980, pp. 562-574.

[37B] S.P. Lloyd, Least Squares Quantization in PCM's, Bell Telephone Laboratories Paper, Murray Hill, N.J., 1957.

[38B] F. Itakura, S. Saito, Analysis Synthesis Telephony Based upon Maximum Likelihood Method, Reports of the 6th International Cong. Acoust., Y. Kohari, ed. Tokyo, C-55, C-17-20, 1968.

[39B] J.O. Ibikunle, N.M. Nasrabadi, R.A. King, Design of Coder for Video Conferencing, First GREISI-CESTA Image Symposium, Biarritz, France, May 21-25, 1984.

[40B] J.G. Robson, Spatial and Temporal Contrast-Sensitivity Functions of the Visual System, J. Opt. Soc. Am., vol. 56, 1966, pp. 1141.

[41B] Z.L. Budrikis, Model Approximations to Visual Spatio-temporal Sine-wave Threshold Data, Bell Systems Tech. J., vol. 52, 1973, pp. 1643.

[42B] M. Miyahara, Analysis of Perception of Motion in TelevisionSignals and its Application to Bandwidth Compression, IEEE Trans., COM-23, 1975, p.761.

[43B] A.J. Seyler, Z.L. Budrikis, Detail Perception after Scene Changes in Television Image Presentations, IEEE Trans. Information Theory, IT-11, 1965, p. 31.

[44B] R.F.W. Pease, J.O. Limb, Exchange of Spatial and Temporal Resolution in Television Coding, Bell Syst. Tech. J., vol. 50, 1971, p. 191.

[45B] J.O. Limp, R.F.W. Pease, A Simple Interframe Coder for Video Telephony, Bell Syst. Tech. J., vol. 50, 1971, p. 1877.

[46B] R.E. Graham, Predictive Quantizing of Television Signals, IRE Wescon, Conv. Rec. 2, Pt. 4, 1958, p. 147.

[47B] D.J. Connor, R.F.W. Pease, W.A. Scholes, Television Coding using Two-dimensional Spatial Prediction. Bell Syst. Tech. J., vol. 50, 1971, p. 1049.

[48B] I.J. Dukhowich, J.B. O'Neal, A Three-dimensional Spatial Non-linear Predictor for Television, IEEE Trans. COM-26, 1978, p. 578.

[49B] P. Nall, R. Zelinski, Bounds on Quantizer Performance in the Low Bit-rate Region, IEEE Trans. COM-26, 1978, p. 300.

[50B] J.O.Limb, J.A. Murphy, Estimating the Velocity of Moving Images in Television Signals, Comp. Graph. Image Proc., vol. 4, 1975, p. 311.

[51B] A.N. Netravali, J.D. Robbins, Motion-Compensated Television Coding, Part I, Bell Syst. Tech. J, vol. 58, 1979, p. 631.

[52B] Y. Ninomiya, Y.Ohtsuka, A Motion-compensated Interframe Coding Scheme for Television Pictures, IEEE Trans. on Communications, vol. COM-30, no. 1, Jan. 1982, pp. 201-211.

[53B] F.W. Mounts, Video Encoding System with Conditional Picture Element Replenishment, Bell Syst. Tech. J., vol. 48, 1969, p. 2545.

[54B] B.G. Haskell, Differential Addressing of Clusters of Changed Picture Elements for Interframe Coding of Video Telephone Signals, IEEE Trans. COM-24, 1976, p. 140.

[55B] J.O. Limb, R.F.W. Pease, K.A. Walsh, Combining Intra-frame and Frame-to-Frame Coding for Television, Bell Syst. Tech. J., July 1974, pp. 1137-1173.

[56B] J.A. Roese, W.K. Pratt, G.S. Robinson, Interframe Cosine Transform Image Coding, IEEE Trans. COM-25, 1977, pp. 1329-1338.

[57B] T.R. Natarajan, N. Ahmed, On Interframe Transform Coding, IEEE Trans. COM-25, 1977, pp. 1323-1329.

[58B] A. Gersho, B. Ramamurthi, Image Coding using Vector Quantization, IEEE Conf. on Acoustic Signal and Speech Processing, 1982, pp. 428-431.

[59B] T. Murakami, K. Asai and E. Yamazaki, Vector Quantization of Video Signals, Electr. Letts., 11 Sept. - Nov. 1982, vol. 18, no. 23.

[60B] E.J. Delp, O.R. Mitchell, Image Compression using Block Truncation Coding, IEEE Trans. Commun., vol. COM-27, Sept. 1979.

[61B] D.J. Henly, O.R. Mitchell, Digital Video Bandwidth Compression using Block Truncation Coding, IEEE Trans. on Commun., vol. COM-29, no. 12, Dec. 1981, pp. 1809-1817.

[62B] N.M. Nasrabadi, R.A. King, Transform Coding using Vector Quantization, 1983 Conf. on Information Science and System, 23-25 March 1983, John Hopkins University, Baltimore, Maryland, USA.

[63B] J.M. Fraser et al., Overall Characteristics of a TASI, Bell Syst. Tech. J., vol. 41, July 1962.

[64B] N.M. Nasrabadi, R.A. King, Computationally Efficient Adaptive Block-Transform Coding, Proc. EUSIPCO-83, 2nd European Conf. on Signal Processing, Sept. 12-13, 1983.

[65B] N.M. Nasrabadi, R.A. King, A New Image Coding Technique using Transform Vector Quantization, 1984 IEEE Int. Conf. on Acoustic, Speech and Signal Processing, March 19-21, 1984, San Diego, California.