DIGITAL ENCODING OF BLACK AND WHITE FACSIMILE SIGNALS

by

M. G. B. ISMAIL

A Doctoral Thesis

Submitted in partial fulfilment of the requirements for the

award of

Doctor of Philosophy

of the Loughborough University of Technology

SYNOPSIS

As the costsof digital signal processing and memory hardware are decreasing each year compared to those of transmission, it is increasingly economical to apply sophisticated source encoding techniques to reduce the transmission time for facsimile documents. With this intent, information lossy encoding schemes have been investigated in which the encoder is divided into two stages. Firstly, preprocessing, which removes redundant information from the original documents, and secondly, actual encoding of the preprocessed documents.

Initially, a psychovisual coding scheme called Adaptive Pel Location Coding (APLC) is described where preprocessing is incorporated within the algorithm itself. Two further preprocessing schemes are later proposed. The first, a simple but effective scheme, employs a set of masks to remove notches, pinholes and isolated picture elements in the original image. At the receiver, the application of another set of masks restores the image as accurately as possible. The second, a more involved method, uses subsampling and interpolation for pre- and post-processing respectively. Here, a 4:1 sample rate reduction is achieved resulting in very high overall compression efficiency. At the receiver, 'cosmetic' restoration is applied to the interpolated image to give a visually pleasing result.

The above schemes are preludes to two different source encoding techniques - Classified Adaptive Block/Run-Length Coding (CABC), and Adaptive Block/Location Coding (ABLC). CABC is a line-by-line sequential predictive coding scheme which divides each scan-line adaptively into smaller blocks using the probability of prediction errors as the criterion. ABLC, on the other hand, is an area coding algorithm which takes m x m elements at a time and uses a numerical 'complexity' measure as a coding criterion. Theoretical aspects of ABLC are discussed and expressions showing the dependence of bit-rate on image complexity are established. For all schemes described, substantial values of compression are achieved, the highest reducing the transmission time of a typical A4 document to about 20 seconds (transmitted at 4800 bits per second).

As a further development, coding of speech using one of the described techniques is considered. The aim is to transmit both speech and text using the same encoder. Delta Modulation (DM) is used as the preliminary stage since its binary output is very similar to a facsimile signal and is therefore amenable to further coding. Results obtained show that a remarkable reduction in bit-rate can be achieved.

ACKNOWLEDGEMENTS

I wish to express my most sincere gratitude to Mr. R.J. Clarke for his keen enthusiasm, constant guidance and help through the course of this work. His invaluable suggestions and comments helped considerably in the development of the work presented here. I am extremely grateful to Dr. R. Steele for initiating this work without whom this thesis would not be realised in its present form. His constant encouragement and advice have been the source of my inspiration throughout this thesis.

My thanks are also due to Dr. C. Xydeas for many fruitful discussions on speech coding and to all my colleagues, in particular Mr. K.N. Ngan, for some important discussions on transform coding. I also wish to express my special thanks to Mr. Phil Atkinson for his assistance in developing the photographs in this thesis.

A special gratitude to my parents for their constant encouragement and continuous moral support throughout my education.

I gratefully acknowledge my sponsors, the Malaysian Government and the University of Science, Malaysia, for their financial support which made this work possible.

Finally, my utmost gratitude to my wife, Yasmin, for her constant encouragement, understanding and patience which enabled me to see this thesis through to completion.

LIST OF ABBREVIATIONS AND SYMBOLS

ABC	:	Auto-adaptive block coding
ABLC	:	Adaptive Block/Location Coding
ADM	:	Adaptive Delta Modulation
ADPCM	:	Adaptive Differential Pulse Code Modulation
AM	:	Amplitude Modulation
ARQ	:	Automatic Repeat Request
APLC	:	Adaptive Pel Location Coding
BFICC	:	British Facsimile Industry Compatible Committee
BER	:	Bit error rate
BPB	:	Basic picture block
CABC	:	Classified Adaptive Block/Run-Length Coding
CCITT	:	International Telegraph and Telephone Consultative Committee
CCD	:	charged coupleddevice
CLAP	:	Classified Pel Pattern Method
CR	:	compression ratio
CFDM	:	Constant Factor Delta Modulation
CVSD	:	Continuous Variable Slope Delta Modulation
CSM	:	Combined Symbol Matching
DM	:	Delta Modulation
DSB	:	double sideband
DF	:	Depth First
DCT	:	Discrete Cosine Transform

EDC	:	Easily Decodable Run-length code
EDIC	:	Edge Difference Coding
EOL	:	End of line
EIA	:	Electronic Industries Association
ESF	:	Error Sensitivity Factor
EPB	:	Elementary picture block
FM	:	Frequency Modulation
FEC	:	Forward acting error correction code
GES	:	Greater Error State
IEE	:	Institution of Electrical Engineers
IEEE	:	Institution of Electronic and Electrical Engineers
ITC	:	Intermediate Ternary Code
IPB	:	Initial picture block
LES	:	Lesser Error State
LDM	:	Linear Delta Modulation
MHC	:	Modified Huffman Code
MRC	:	Modified READ Code
MSLT	:	Minimum scan line time
MUC	:	Make-up codewords
OCR	:	Optical Character Recognition
PDQ	:	Predictive Differential Quantizing
PSTN	:	Public Switched Telephone Network
PCM	:	Pulse Code Modulation
RHC	:	Reduced Huffman Code
READ	:	Relative Element Address Designate
RAC	:	Relative Address Coding

RTC	Return to control
SPB	Subpicture block
SQNR	Signal to quantization noise ratio
THC	Truncated Huffman Code
TC	Terminating codewords
VISI	very large scale integration
VSB	Vestigial sideband
VADS	Visual Acquisition and Display System
V27 ter.	CCITT Recommendation - 4800/2400 bits per second modem standardised for use in the PSTN
V29	CCITT Recommendation - 9600 bits per second modem standardised for use on leased circuits
WBS	White Block Skipping

B_n	B-spline basis function
C_k	coefficients used in B-spline interpolation
C_p	Complexity
$C_{p\blacksquare}$	Complexity due to black primitives
$C_{p\square}$	Complexity due to white primitives
$\hat{f}(\theta)$	Interpolated continuous function in one-dimension
$\hat{f}(\theta,\eta)$	Interpolated continuous function in two-dimensions
f_s	sampling frequency
H	entropy
H_w	entropy of white runs
H_b	entropy of black runs
H_{pel}	entropy per pel

ℓ_+	:	number of quartic divisions
L_r	:	output of delta modulator at rth sampling instant
$m_{\square}$	:	number of white primitives
$n_{\blacksquare}$	:	number of black primitives
$N_c^{<4>}$	:	connected numbers (4-connectivity)
$N_w^{<8>}$	:	weighted connected numbers (8-connectivity)
$P_w(r)$	:	probability of white runs of length r
$P_b(r)$	:	probability of black runs of length r
$P_{m,j}$	:	prefix bits (used in APLC)
Q_{max}	:	maximum theoretical compression factor
$\bar{r}_b$	:	average black run
$\bar{r}_w$	:	average white run
$S_k(\theta)$	:	basis function used for interpolation
x_i	:	sequence of pels or speech samples
$\Delta = \theta_k - \theta_{k-1}$	:	uniform spaced data points (knots)
$\ast$	:	convolution
$\oplus$	:	modulo-2 addition
γ	:	step size for delta modulation
α, β	:	leakage factors for CVSD
τ	:	time constant
dB	:	decibel

CONTENTS

Page

CHAPTER I

THESIS INTRODUCTION

1.1. INTRODUCTION

Processing and communication of graphics which are predominantly black and white has attracted considerable interest in the last decade or so. It is not difficult to find such images in practice. Graphics processing and communication is both useful in the business world as well as in the home environment, and examples include business letters, documents, weather maps, geographic maps, fingerprint cards and newspapers. Teleconferencing (audio-graphics), document transmission (conventional facsimile), electronic mail (word processors), graphics for computer-aided design and picture data bases are further examples of its use in the business world. In the home, the efficient use of the public switched telephone network and the television set has given birth to systems such as videotex (eg. British PRESTEL, French ANTIOPE, Canadian TELIDON and Japanese CAPTAIN) and other home information systems.

Although the input signal, i.e., the light intensity reflected from a graphical document, is an analogue quantity, there are advantages in digitising it to allow the best use of integrated circuit technology. Integrated circuits for processing and memory are becoming less expensive each year, whilst very large-scale integration (VLSI) promises to provide denser digital circuits which are easier to test and maintain compared with equivalent analogue circuits. Also, source encoding of the signals becomes easier in the digital domain. Furthermore, digital transmission has advantages in that signals of several types representing different sources can be multiplexed easily and periodic regeneration is possible when the signals become weakened and distorted. Digital signals also lend themselves to the technology of encryption and, of course, in-service performance monitoring, protection switching and off-line fault isolation are easily implemented.

Most graphics are black and white (two-level), although with time, there will be an increasing tendency to use graphics with gray-scale (multilevel) and colour. In this thesis, greater emphasis is put on two-level documents than on multilevel images. A typical A4 size document (210mm x 297mm) sampled at a density of 8 elements per millimetre generates about 4 million bits. The easiest and probably the most reliable way of conveying this information would be to transmit these 4 million bits unchanged through the transmission channel to the receiver, since in this case no special coding is required and disturbances on the telephone lines will perhaps only affect a few samples. However, the transmission of such documents through the telephone network at a data rate of 4800 bits per second would take nearly a quarter of an hour; too long under normal circumstances. Clearly, such a high volume of data per document is often uneconomical to store or transmit directly so techniques of data compression are of great interest. This thesis is concerned mainly with efficient representation of graphics and documents by employing source encoding techniques, either information lossless or information lossy, so that the transmission time and memory requirement for storage is as low as possible.

1.2. BACKGROUND TO THE RESEARCH

Although digital image processing, especially in areas of digital television and Transform Coding for viewphone and viewdata applications, is a well established field of research, it has only been pursued for about six years in the Department of Electronic and Electrical Engineering, University of Technology, Loughborough. Research in facsimile coding has been recently introduced with the aim of expanding into another area of image processing. Its application is somewhat different from the above two areas of research, and is focussed mainly on the efficient transmission of documents over telephone lines with the receiver producing a replica of the original document in hard copy.

The research work initially used the facilities provided by the University's Computer Centre, on an International Computers Limited, ICL 1904A mainframe computer, and the departmental Hewlett Packard HP2100A mini-computer. Research began in autumn 1978 and work was made possible, thanks to the British Post Office (now British Telecom), for providing the standard CCITT documents (See Chapter III). During early 1979, a new PRIME400 mainframe computer was added to the existing facilities of the Computer Centre, offering greater computer power for research. However, that computer was mainly used for development of programs due to the limited storage space on discs allocated to each user, compared with that required to store the CCITT documents. Anyway, being interactive, its fast response time was an advantage for development work. At the same time, the Visual Acquisition and Display System (VADS)[+] had been completed and interfaced with the HP2100A, hence providing a video display capability to the existing research facilities. Basically, VADS offers visual display of processed data stored on magnetic tapes transferred from the ICL1904A computer or the PRIME400 computer and it also has the capability of digitising analogue inputs generated by a camera. Although the latter function is inadequate for use with two-level graphics, due to the poor resolution of the camera, the use of VADS as a display unit for visual assessment of processed data was found to be satisfactory for the present research. Ideally, a facsimile printer or recorder interfaced with either the HP2100A or the recently acquired PDP 11/34 mini-computer is required which would provide a true representation and enable a better visual assessment of the processed data on A4 size paper. The basic research facilities with which much of the work was carried out are shown diagrammatically in Fig. 1.1.

[+] Built by W.C. Wong, now at Bell Laboratories, Crawford Hill. Holmdel, New Jersey, U.S.A.

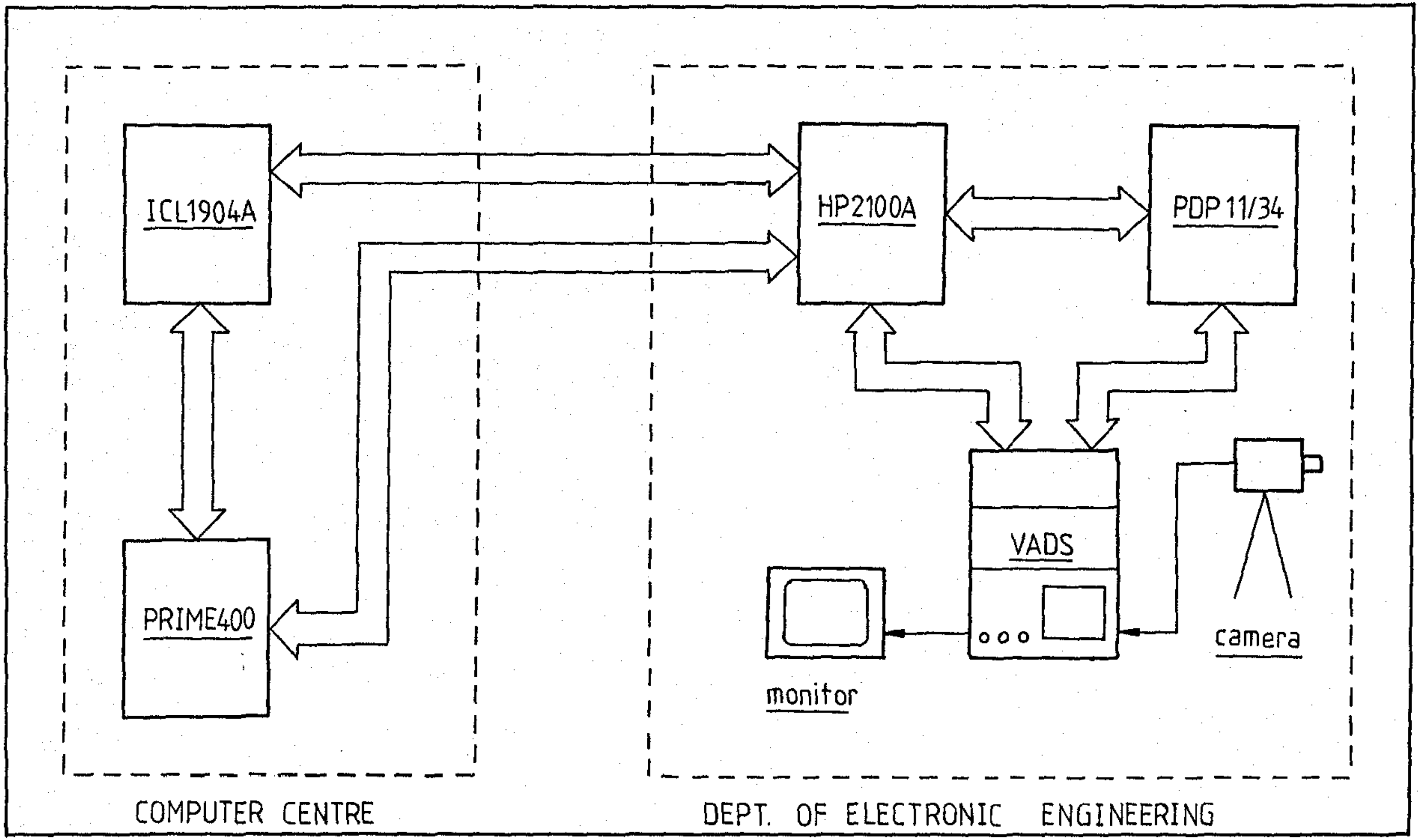


Fig. 1.1 - Configuration of basic research facilities

1.3. ORGANISATION OF THE THESIS

The research programme is purely software oriented with consideration given to hardware aspects when designing a particular system. The research can be categorised loosely into two main divisions, that is, investigating information preserving and non-information preserving schemes for facsimile coding, with both schemes coupled in the later chapters to form a basic system. As a slight digression, the application of facsimile techniques to speech, in which a simple, inexpensive coder (such as a delta modulator) is used as a preliminary stage, is briefly explored, with the intention of marrying the transmission of speech and documents using a single coder. The overall lay-out of the thesis is illustrated in Fig. 1.2.

Chapter II is of an introductory nature reviewing the 'state of the art' in facsimile coding. It begins with the historical background of facsimile, and gradually introduces some of the known techniques of digital facsimile coding. These are divided into two categories: information preserving and non-information preserving schemes. The effects of transmission errors and methods for combating them are also mentioned in this chapter.

The one- and two-dimensional coding standards, which have been ratified by the CCITT, are described in Chapter III, and some of the factors leading to their choice are discussed.

In Chapter IV, a relatively simple method of facsimile coding called Adaptive Pel Location Coding (APLC) is described. It

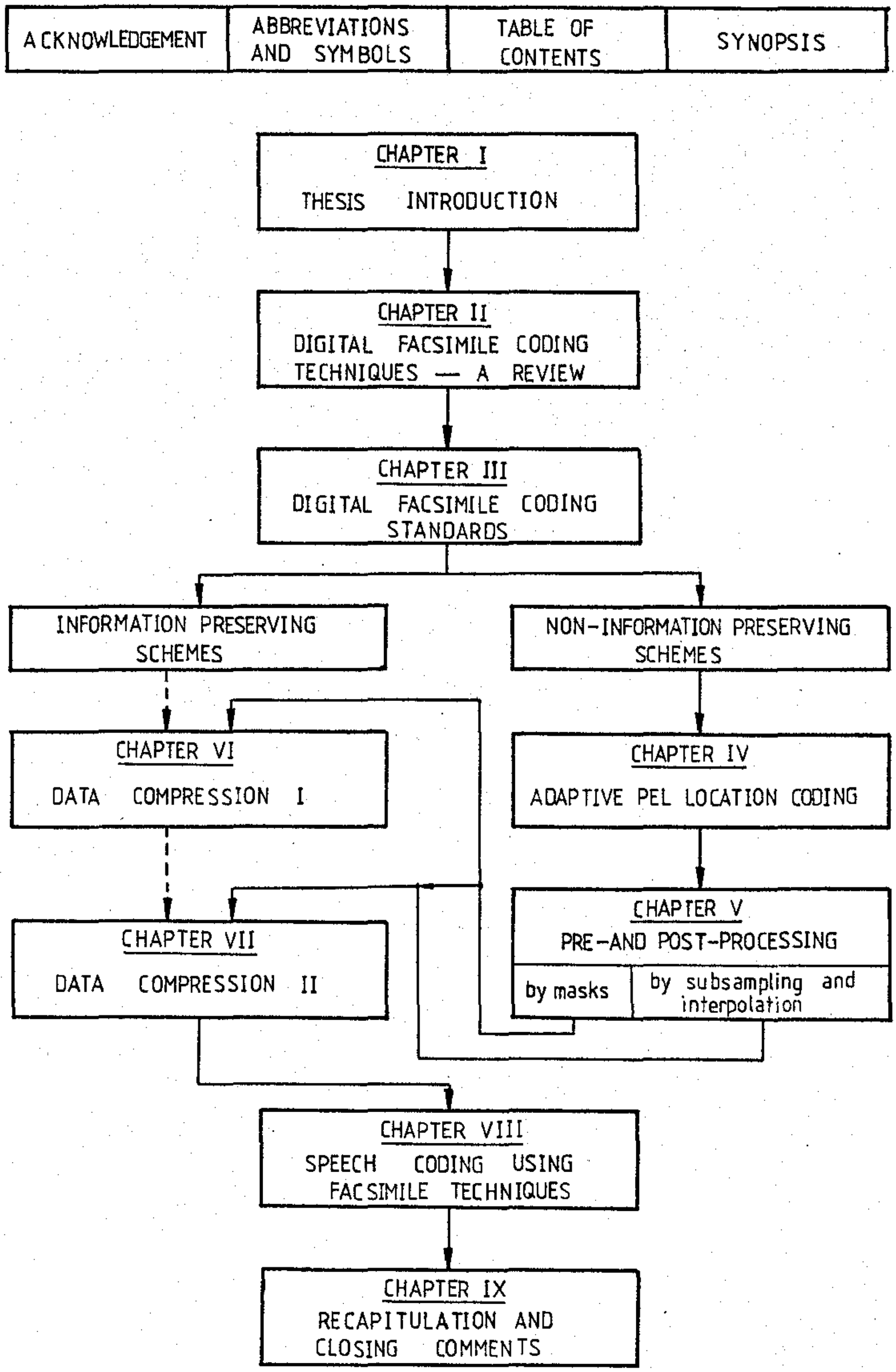

ACKNOWLEDGEMENT
ABBREVIATIONS AND SYMBOLS
TABLE OF CONTENTS
SYNOPSIS
CHAPTER I
THESIS INTRODUCTION
CHAPTER II
DIGITAL FACSIMILE CODING TECHNIQUES — A REVIEW
CHAPTER III
DIGITAL FACSIMILE CODING STANDARDS
INFORMATION PRESERVING SCHEMES
NON-INFORMATION PRESERVING SCHEMES
CHAPTER VI
DATA COMPRESSION I
CHAPTER IV
ADAPTIVE PEL LOCATION CODING
CHAPTER VII
DATA COMPRESSION II
CHAPTER V
PRE-AND POST-PROCESSING
by masks
by subsampling and interpolation
CHAPTER VIII
SPEECH CODING USING FACSIMILE TECHNIQUES
CHAPTER IX
RECAPITULATION AND CLOSING COMMENTS

Fig. 1.2 - Thesis Lay-out

is an information lossy scheme and operates in either one or two dimensions. In a manner similar to block coding, each block is inspected to see whether it contains all black or all white pels, when appropriate prefix bits are assigned to it. If the block contains one white or one black pel, the scheme ignores them and assumes that the block is completely black or white respectively. Computer simulations show that improvements in compression ratios can be obtained in this way.

Chapter V describes two pre- and post-processing schemes; the first uses a set of masks, whilst the second employs, respectively, subsampling and interpolation. These schemes are later employed in two data compression schemes explained in Chapter VI and Chapter VII. In the first scheme, a set of masks with predetermined patterns is applied sequentially against the source image to remove redundant information, like notches and pinholes, that is present. Another set of masks at the receiver restores the image as closely to the original as possible. The second scheme is more involved, and various subsampling schemes are considered. The simplest technique, producing by far the most acceptable quality, is subsampling by taking alternate picture elements both horizontally and vertically. The overall 4:1 sample rate reduction results in high compression ratios being obtained. Three interpolation schemes with appropriate restoration, for the enlargement of the subsampled image, are investigated. A subjective experiment to assess the order of preference of the restored images is also conducted.

Whilst Chapter IV and Chapter V concentrate on non-information preserving schemes, Chapter VI and Chapter VII deal with data compression schemes which are, in essence, information preserving. Data compression scheme I, as described in Chapter VI, uses a set of masks as its preprocessing stage. Called Classified Adaptive Block/Run-Length Coding (CABC), it is a two-dimensional line-by-line sequential coding scheme where each scan line is effectively processed by making use of the vertical correlation between the current line and the preceding line. Assuming that the source image is an output of a Markov source, redundancy is reduced by employing a 7th order Markov model predictor in which the present pel is predicted on the basis of seven previous pels. The prediction errors are assumed to result from a memoryless source, and each scan-line containing those prediction errors is adaptively divided into smaller blocks using the probability of correct prediction as a criterion. The choice of block size is shown to be nearly optimum. The effect of transmission errors on the system is also considered.

Chapter VII discusses another data compression scheme, in which the input image is subsampled prior to coding. Called Adaptive Block/Location Coding (ABLC), it is essentially an area coding algorithm which takes, at a time, a square block of m x m pels. A detailed mathematical analysis of block coding and location coding based on complexity of an image is presented. Four different configurations of ABLC are examined and the effect of transmission errors on one of the systems is investigated. The results of using the CCITT coding standards on the subsampled documents are also reported.

In Chapter VIII, as a further development, coding of speech using the technique described in Chapter VI is considered. The aim is to transmit both speech and text using the same encoder. Delta modulation is used as a preliminary preprocessing step since its binary output is very similar to a facsimile signal and the two signals are therefore amenable to further similar coding. Different delta modulation systems are used and their performance evaluated using the previously described coding scheme. By exploiting the quasi-periodic nature of speech, a two-dimensional picture of the delta modulation bitstream (voiced region only) is formed in the hope of reducing the average transmission rate further. Simulation results show that a remarkable reduction in bit-rate can be achieved whilst still maintaining the quality of speech of the original delta modulation systems.

Finally, in Chapter IX the main results are discussed and suggestions for further research are made. The thesis is then concluded with a brief review of what has been achieved in the research, together with closing comments.

CHAPTER II

DIGITAL FACSIMILE CODING TECHNIQUES — A REVIEW

2.1. INTRODUCTION

Two significant events in the field of telecommunication took place early in 1926. One was the first public demonstration, in January, by John Baird, of television, in a little laboratory in London's Soho district. The other was the inauguration, three months later, of commercial transatlantic radio facsimile service (for the transmission of news photos) by the Radio Corporation of America(1). While television remained confined to the laboratory bench, its static (in the sense that it was concerned with still, as opposed to moving, images) counterpart, already had several decades of practical application behind it, in the course of which it had steadily progressed towards a high state of refinement.

Although television was to enjoy a somewhat more rapid technological growth than facsimile in the ensuing years, it was not until two decades later that it finally caught up as a fully fledged commercial reality, and it was not until the advent of communication satellites in the 1960's that it was able to expand on a limited commercial basis. At the same time, facsimile has established, and even gradually reinforced, its position as an indispensable telecommunications medium.(1)

Facsimile, broadly defined, is the process of transmitting and reproducing printed matter, still pictures, etc., from one place to another by electrical means, in which a reasonably faithful copy is permanently recorded at the receiving end in any one of several forms - either hard copy (printers) or soft copy (displays). Although facsimile includes techniques for transmitting colour pictures, monochrome images and the like, two-tone images, i.e. black and white documents, assumed to be digital, static and having two-levels rather than multilevel picture elements, have dominated and stimulated digital facsimile in the last fifteen years or so. In the early years, facsimile was confined to specialised users like newspaper agencies and huge multinational companies, but now the list of facsimile users is more broadbased, ranging from domestic users to big financial institutions.

Early machines were mainly of the analogue type. The output of a facsimile scanner, resulting from a succession of apparently random light reflectance variations, varies in both amplitude and frequency and consists primarily of an irregular chain of transitions as shown in Fig. 2.1. Furthermore, the frequencies vary from several thousand hertz right down to zero. It is this characteristic that necessitates modification of the facsimile signal in order for it to be suitable for transmission over voice grade telephone lines. The problem is easily overcome by modulating the baseband facsimile signal onto a carrier either by amplitude (AM) or frequency (FM) modulation techniques prior to transmission. The chief advantage of FM over AM is its relative immunity to the effects of electrical noise, unfortunately at the expense of additional bandwidth.

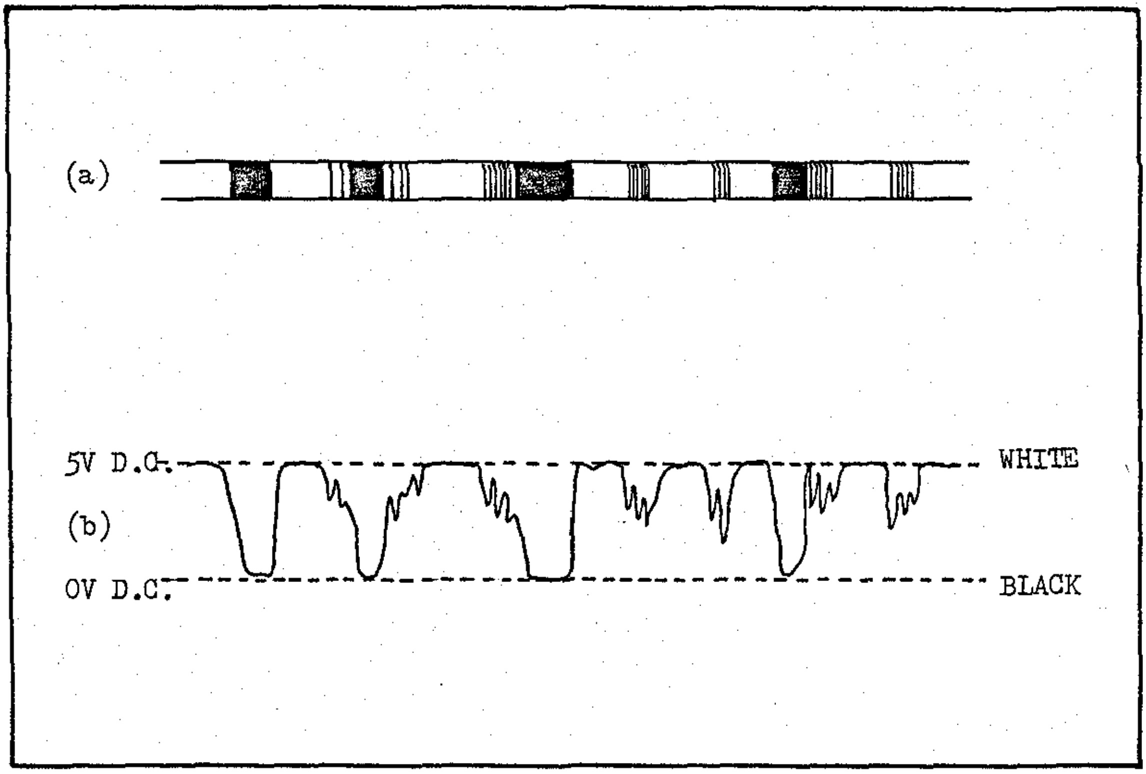


Fig. 2.1 - Analogue facsimile baseband signal (b) resulting from random light variations within a scanned segment (a) of a subject copy. (From Ref. (1))

As a by-product of modulation, sidebands are produced; double sidebands (DSB) in the case of AM and an infinite number of side-frequencies in the case of FM, though there are commercially available FM machines whose bandwidth requirements are similar to that of DSB-AM. The suppression of the upper sideband and the carrier of a DSB-AM signal results in a more efficient use of the transmission channel without (theoretically) affecting the information being transmitted. However, in practice such suppression is difficult to realise without sacrificing a portion of the lower sideband. A good compromise, and one that has been resorted to in a number of commercial facsimile systems, is vestigial sideband transmission (VSB). In VSB-AM, the entire lower sideband, the carrier and only a small portion of the upper sideband are transmitted, therefore still allowing a saving in bandwidth.

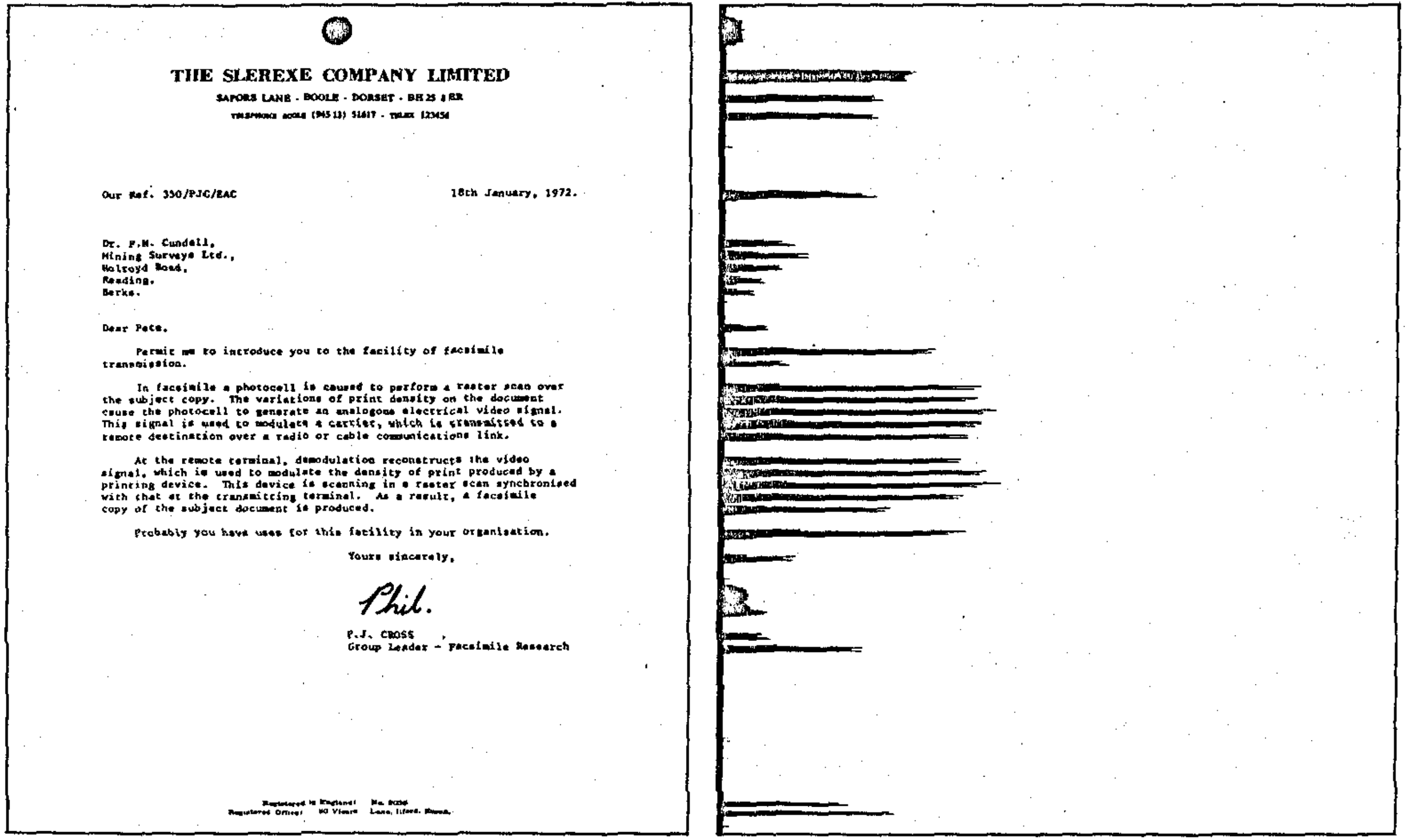
DSB-AM and FM systems are the basis of CCITT Group 1 machines which require a transmission time of six minutes for an A4 size page at a resolution of 3.85 pels/mm in the horizontal and vertical directions, whilst VSB-AM is used in Group 2 machines whose transmission time is half that the machines of Group 1.

Analogue transmission, however, suffers from a number of problems, particularly impairments caused by the transmission medium. External interference like lightning, power line induction and crosstalk are amongst the principal causes which can degrade a facsimile recording. Lightning and power line induction which occur spasmodically can obliterate essential details of the transmitted copy, but generally diagonal stripe[1] patterns are observed over the entire facsimile recording. Crosstalk, despite elaborate measures taken to minimise it, also occurs causing damage to the received image. Internal noise within the transmission path itself, for example contact noise in the switching system and transistor noise within terminal amplifiers and repeater stations, also have the potential to degrade the resulting recording. Level fluctuations, which are associated with the intrinsic characteristics of long telephone channels are beyond human control (specifically the fading encountered in long range radio reception), and tend to cause different grey levels to appear in the resulting copy. Echoes and delay distortion are transmission impairments which can visibly affect the quality of the received copy in the form of 'ghost' and 'smear' respectively. Other impairments include phase jitter, i.e. the accidental shifting of transmitted frequencies at some fixed rate, and harmonic distortion.

One way of overcoming the problems of analogue systems mentioned above is to convert the analogue signal to a digital code stream which can be regenerated periodically as it becomes weakened and distorted. This will permit the faithful reconstruction of the transmitted picture. Furthermore, the techniques of encryption and forward error correction can be applied to digital systems without much difficulty.

The chief disadvantage of analogue systems is their relatively slow transmission time. It takes about six minutes using DSB-AM and FM and about three minutes for VSB-AM to transmit an A4 size page at normal resolution. Faster transmission cannot be readily obtained using these analogue techniques because of the restricted bandwidth of the voice grade telephone network. In digital facsimile systems, however, data compression schemes can be readily applied to reduce the redundancy present in the digitised input image. Source encoding techniques in which efficient algorithms performing a completely reversible one-to-one transformation can be devised, and typically reduce the transmission time to about one minute, i.e., well below the three minute level offered by their analogue counterparts. An example of the performance of such a lossless transformation is illustrated in Fig. 2.2[2]. Fig. 2.2(a) shows a two-level image of a business document and Fig. 2.2(b) illustrates the binary pseudoimage representing the transformed data. The left justified black pels indicate the number of bits needed to describe each horizontal scan-line while the white pels represent the savings due to compression, i.e. the redundancy present. Note the correlation between Figs. 2.2(a) and 2.2(b), especially evident in the large number of bits required to encode scan-lines containing text and the small number of bits needed to represent empty scan-lines. The digital image of Fig. 2.2(a) can be reconstructed identically, given only the quantity of coded bits indicated by Fig. 2.2(b) for each scan-line.

Section 2.2 below discusses the evolution of digital facsimile coding schemes, while Section 2.3 describes the current state-of-the-art, in which techniques are classified as either non-information or information preserving. One- and two-dimensional information preserving schemes will be outlined in this section,



(a) Original image

(b) Compressed image

THE SLEREXE COMPANY LIMITED

SAPORS LANE - BOOLE - DORSET - BH 25 8 ER

TELEPHONE BOOLE (945 13) 51617 - TELEX 123456

Our Ref. 350/PJC/EAC 18th January, 1972.

Dr. P.N. Cundall,
Mining Surveys Ltd.,
Holroyd Road,
Reading.
Berks.

Dear Pete,

Permit me to introduce you to the facility of facsimile transmission.

In facsimile a photocell is caused to perform a raster scan over the subject copy. The variations of print density on the document cause the photocell to generate an analogous electrical video signal. This signal is used to modulate a carrier, which is transmitted to a remote destination over a radio or cable communications link.

At the remote terminal, demodulation reconstructs the video signal, which is used to modulate the density of print produced by a printing device. This device is scanning in a raster scan synchronised with that at the transmitting terminal. As a result, a facsimile copy of the subject document is produced.

Probably you have uses for this facility in your organisation.

Yours sincerely,

Phil.

P.J. CROSS
Group Leader - Facsimile Research

Registered in England: No. 2038
Registered Office: 60 Vicars Lane, Ilford. Essex.

Fig. 2.2- Example of binary image compression. [from ref(2)]

and other related coding techniques summarised in Section 2.4. The sensitivity of the compressed data to channel errors after the application of source encoding techniques is often an important consideration. Section 2.5 discusses the problem of the effects of errors on the decoded image and suggests ways of minimising these effects. Section 2.6 concludes this review chapter.

2.2. EVOLUTION OF DIGITAL FACSIMILE CODING SCHEMES

Activity in the field of facsimile image compression dates back at least to the formalising of information theory by Shannon(3) in 1948. Soon after, Laemmel(4) (1951) published an extensive report on binary image compression including many of the basic concepts used today. Because Laemmel's work was not published in a major journal, many of his algorithms were reinvestigated by others. Another significant contribution was the paper by Elias(5), who laid the foundation for predictive coding.

The first publication about measurements on binary image data and on compression hardware occurred in the late 1950s and early 1960s. Deutsch(6) made the first attempt to measure the statistical properties of a small sample of typewritten or printed material, which was later extended by Michel(7) to a full size page of text and line drawings. Michel concluded that theoretically a saving of approximately ten is possible with run-length coding systems which employ an optimum code. In 1957, Michel et. al.(8) described a complete system design with binary image compression, followed some time later by Wyle et. al.(9). Both systems were based on run-length coding. In 1959, Capon(10) described a probabilistic model for run-length coding of pictures based on the first-order Markov process. Further developments in compression algorithms were made in the 1960s, including the work of Wholey(11) who designed a general method of coding two-level pictorial data using two-dimensional predictive coding. Papers reporting statistical measurements on facsimile images became commonplace and actual implementation became a reality. Facsimile was always seen as a major application for binary image compression.

The emergence of low-cost integrated circuits stimulated further research. The 1970s saw the appearance of new improved compression algorithms(12-13) which steadily reduced the time needed to transmit an image. The introduction of large number of digital facsimile machines into national networks generated international efforts to standardise them. This in turn further stimulated and accelerated more efficient algorithm development. At present, the subject of binary image compression has reached such a high state of development and refinement that even higher performance algorithms are beginning to appear offering even greater compression.

2.3. CLASSIFICATION OF DIGITAL FACSIMILE CODING TECHNIQUES

A digital facsimile transmission system is a particular type of communication system - particular in the sense that a knowledge of physical characteristics of the image source and image destination are vital in its system design. Image source characteristics include the way in which the input source is sampled, quantised and thresholded to generate a binary picture. Models[2] describing the source need to be formulated in order to represent the input image as efficiently as possible in its coded form. Proper modelling of the destination or receiver of images also is vital to the effective design of a complete system. Cognizance of the human observer is important in the sense that image distortion should be controlled such that 'it cannot be seen' even though it is actually present[14]. In practice, the coding system design is optimised to present minimal visible distortion by exploiting the limitations of human vision, for example, in facsimile the application of certain kinds of masks which will 'cosmetically' improve the final copy is possible. In general, efficient digital facsimile transmission systems cannot be realised by a 'black box' design approach based on communication theory alone.

The fundamental structure of a digital facsimile transmission system is shown in Fig. 2.3. The input document is first scanned using, for example, a charged coupled device (CCD), after which each picture element is thresholded to generate one of two values, black or white. A preprocessing step may be incorporated to remove redundant information that is present after the thresholding process. Redundant information includes random isolated pels or notches and pinholes caused due to indecision in the thresholding process. Next, the source encoder transforms the source image into a form with minimal transmission requirements, which is then converted to a format suitable for transmission. This step involves modulation of the transmission carrier and, often, error correction coding for channel errors. At the receiver, the signal transmitted is

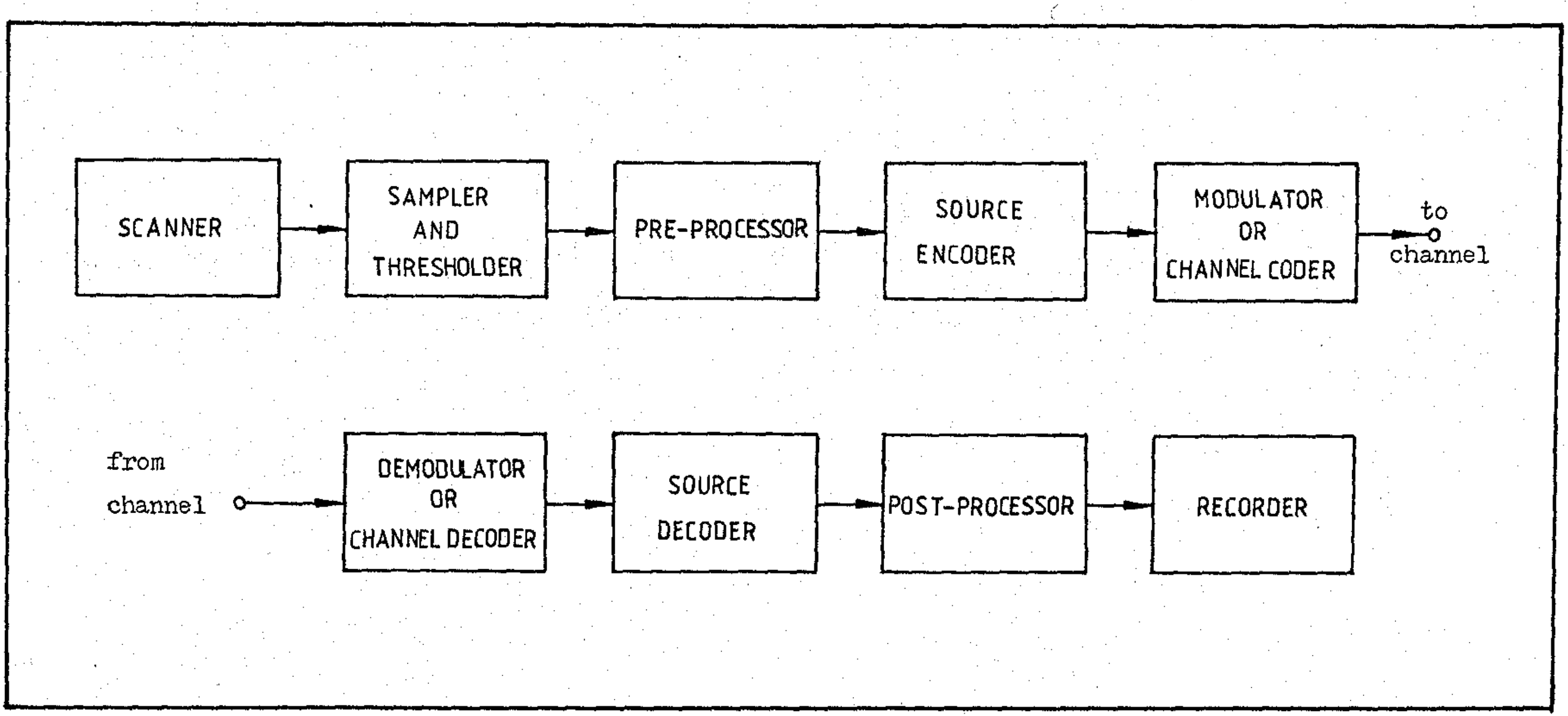


Fig. 2.3 - Fundamental structure of a digital facsimile transmission system

demodulated and channel decoded, after which the source decoder inverts the coding process to reproduce the reconstructed document. If a preprocessing stage was included prior to transmission, then naturally some form of post-processing may be required. Finally, the reconstructed document is printed by a facsimile recorder or stored for later use.

With reference to the fundamental structure of Fig. 2.3, facsimile data compression schemes can be classified into two main divisions: non-information preserving, and information preserving, coding, as shown in Fig. 2.4. Non-information preserving coding uses an irreversible preprocessing operation which has no corresponding post-processing operation which would completely restore the original image. On the other hand, in information preserving coding, there exists a reversible one-to-one transformation in which the original input image is recovered without any loss of information.

Information preserving coding can be divided into two categories: one-dimensional and two-dimensional coding. In the former category, coding is carried out only within each scan-line, whilst in the latter category, coding is performed over two or more scan-lines. Since two-dimensional coding schemes make use of vertical and horizontal correlation, they are able to achieve higher compression factors than one-dimensional coding schemes but become more vulnerable to transmission errors. Two-dimensional coding schemes can be further divided into simultaneous coding of N lines, line-by-line sequential coding and area coding algorithms. In simultaneous coding of N lines, N consecutive scan-lines are taken and processed

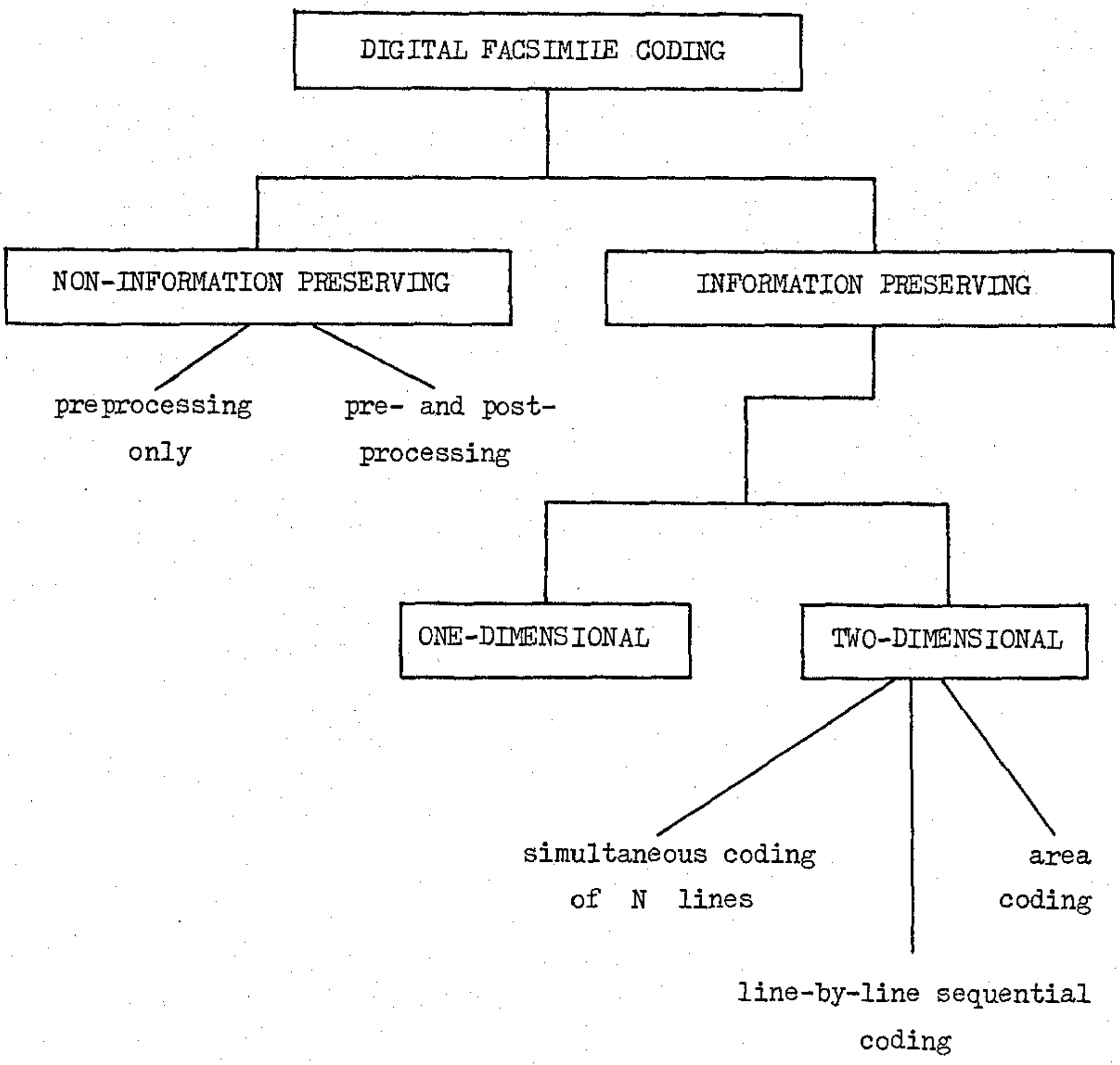


Fig. 2.4 - Classification of Digital Facsimile Coding Techniques

independently, whilst in line-by-line sequential coding, the vertical correlation that exists in facsimile images is exploited, and each scan-line is successively processed by reference to one or two preceding lines. Area coding algorithms, on the other hand, do not make use of previous lines as a reference for coding but rather take successive areas of the source image to be coded independently. In other words, the vertical correlation that exists between successive scan-lines is not fully utilised.

2.3.1 Non-Information Preserving Schemes - Pre- and Post-processing

Non-information preserving schemes, in most cases, are irreversible preprocessing techniques which are employed prior to actual coding. Since they are information lossy, degradations are inevitably introduced into the reproduced image in the sense that the original image cannot be recovered fully. In most cases, there exists a corresponding post-processing technique which will try to restore the image as best as possible. However, the word degradation is sometimes misleading because in digital facsimile systems, the original analogue input goes through the process of sampling and thresholding which in itself is accompanied by some kind of distortion. The distortion appears in the form of jagged edges along the characters, or random isolated dots resulting from impulse noise introduced during the scanning process. There are preprocessing techniques(15-17) available which have been developed to handle this kind of situation. It cannot, therefore, be said that the use of irreversible preprocessing always causes more degradation than reversible processing from the viewpoint of overall image quality. Besides, higher data compression efficiency can always be obtained if preprocessing is incorporated.

It is observed from Fig. 2.4 that non-information preserving schemes can be categorised in two ways. The first category is where only the transmitter employs a preprocessing stage - corresponding to no post-processing at the receiving end. The second category incorporates both preprocessing and post-processing stages at the transmitter and receiver respectively. In terms of compression efficiency, the latter category is more efficient because more drastic signal modification can be employed. However, because machines of this category need a corresponding post-processor at the receiver, they cannot communicate with other machines without the same facility. The inclusion of techniques in the former category, on the contrary, overcome this disadvantage but offer smaller improvements in compression efficiency than those of the latter category.

Amongst the more simple preprocessing techniques are those described by Netravali(18) and Margner and Zamperoni(19). In both cases, single element runs are discarded without causing serious degradation to the resulting copy. A reduction in entropy by about 10 percent is reported. Other simple techniques, such as isolated black point removal, and bridging(20-21), were proposed in the mid-70s. Kunt and Johnsen(22) proposed what is known as psychovisual coding in their block coding strategies with the aim of increasing the compression ratio by distorting the image in an unnoticeable or at least tolerable way. This is done by considering all blocks having not more than B ones (black pels), as completely white blocks. The parameter B is called the distortion factor, and for the information lossless case, B is equal to zero. An increase in compression ratio by about 8 percent and 17 percent, for B equal to one and two respectively, has been reported[22]. The notchless bilevel quantiser described by Fukinuki[23] is one of the more complex techniques for preprocessing. Most practical facsimile scanners contain a two-level quantiser whose threshold level is normally constant. Since the conditions for scanning a document are variable, the analogue signal level may go up and down around the threshold level. The output of the quantiser is, therefore, dependent on how close the analogue signal is to the threshold. This inevitably causes notches along the horizontal and vertical lines of the digitised document. These notches not only degrade picture quality but also reduce coding efficiency because long runs are broken down into shorter runs and line-to-line correlation is reduced. The solution to the drawback of constant threshold level is the application of the notchless bilevel quantiser, introduced by Fukinuki, in which the threshold level is varied according to the logical state of the preceding pels. An improvement in picture quality is observed, and an increase of more than 20 percent in coding efficiency is achieved using this method. Sometimes, digital facsimile machines are called upon to handle noisy originals such as those from an electrostatic copier. Electrostatically reproduced images are characterised by the presence of impulse noise, which decreases the efficiency of the coder and also affects the final copy.

In other cases, a badly-adjusted type head and worn-type ribbon may cause tiny perforations in the input copy. The application of the notchless bilevel quantiser with logical feedback for these types of input documents will not be effective enough to remove the noise. Ting and Prasada(15-16) provided the solution, with the aim of cleaning up noise already inherent in the input document. Known as Majority Logic Smoothing, it uses a 3 x 3 mask in which the central element is changed depending on the value of the majority of the pels in the neighbourhood around it. This method has the slight drawback in that connectivity is sometimes lost and smaller characters tend to be filled in or fragmented. Majority Logic Smoothing with Contour Checks, in which the central pel is determined by majority logic except for those circumstances when a contour is detected, avoids this difficulty. Simulations in which clean originals are subjected to noise show an increase in compression ratio of the order of 30 percent for one-dimensional run-length coding and about 74 percent for two-dimensional coding(16). Other related schemes include that described by Billings(24). The use of preprocessing partly to solve buffer problems in order to achieve a more constant rate output has been described by Margner and Zamperoni(25).

Some of the more important non-information preserving techniques in the second category, in which a corresponding post-processing step is required, include those described by Takagi et. al.(26), Usubuchi et. al.(27) and Judd(28). Takagi's method takes advantage of the inherent redundancy present in sufficiently high resolution documents and intentionally modifies the facsimile signal to aid bandwidth compression. Used together with two-dimensional predictive coding, some of the local patterns in the digital image are modified so as to reduce the prediction error probability, hence improving coding efficiency. It has been shown[26] that this technique reduces the bit rate by an amount ranging from 18 to 57 percent. Being information lossy, the degradation caused by the preprocessing technique is mitigated by the inclusion of a post-processing technique at the receiver in order to restore the naturalness of the resulting image. On the other hand, the use of parallel thinning algorithms for digital pictures[29] is the basis of Usubuchi et. al's[27] method. Here, preprocessing is achieved by first smoothing and then thinning the width of lines or characters to a single picture element. In other words, the 'skeleton' of the characters is extracted for further coding. At the receiving end, post-processing is carried out by thickening the thinned lines to improve picture quality. A similar scheme was later proposed by Judd[28]. Entropy measurement[27] for this technique shows that there is a reduction in average entropy of about 60 percent compared with that of the original picture.

Other non-information preserving schemes which do not specifically fall into the category employing pre- and post-processing include the Combined Symbol Matching (CSM)[30-32] facsimile system proposed by Chen, Pratt and others and vector coding[33] described by Ramachandran. CSM is a new concept of hybrid symbol matching/

run-length coding in which a document is dynamically segmented into symbol and graphic regions. Alphanumeric characters are efficiently coded by symbol recognition techniques using a prototype library code(31-32) whilst graphics are represented by two-dimensional run-length codes. Vector coding, on the other hand, extracts vector information from raster scanned data and is mainly aimed at computer-aided design and draughting applications. Two further information lossy schemes will be described in full in Chapter V.

2.3.2 Information Preserving Schemes

(A) One-Dimensional Coding Schemes

There are many variations of one-dimensional coding of black and white facsimile signals, most of which, however, are based on the concept of run-length coding. Nevertheless, there are other one-dimensional codes available in the literature which can be regarded as alternatives to run-length coding. Amongst the more important schemes are one-dimensional white block skipping (WBS)(68) and autoadaptive block coding(34-35). This subsection will be devoted to various one-dimensional run-length coding schemes.

The sequence of pels produced by a scanner on a line-to-line basis can be viewed as a succession of pel forming segments of the same colour, with alternating colour from one segment to the next. In this model, the information source produces segments of different lengths that are usually called runs. Codes based on these run-lengths are called run-length codes. In run-length coding, it is assumed that the lengths of black and white element runs are statistically independent.

The probabilistic model for run-length coding of black and white images was first proposed by Capon(10). Based on the first-order Markov model, each pel X_i is dependent on the previous pel X_{i-1}, and the entropy is given by:-

$$H_{pel} = \sum_{X_{i-1}} \sum_{X_i} P(X_{i-1}, X_i) \log_2 P(X_i/X_{i-1}) \qquad (2.1)$$

This led to the development of a run-length source model in which the probability of a white run of length r can be expressed as r-1 transitions from white pel to white pel followed by the transition from white pel to black pel. Under the assumption of infinite length, the white run-length distribution is therefore:-

$$P_w(r) = P(w/w)^{r-1}\ P(b/w) \qquad (2.2)$$

where $P(w/w)$ is the probability of a white pel given that a white pel has occurred and $P(b/w)$ is similarly defined. Similarly, the black run-length distribution is:-

$$P_b(r) = P(b/b)^{r-1}\ P(w/b) \qquad (2.3)$$

where $\sum P_w(r) = 1$ and $\sum P_b(r) = 1$.

The average white and black run-lengths are given respectively by:-

$$\bar{r}_w = \sum_r r \cdot P_w(r)$$
$$\bar{r}_b = \sum_r r \cdot P_b(r) \qquad (2.4)$$

The entropies of the white and black runs are given by:-

$$H_w = - \sum_r P_w(r) \log_2 P_w(r)$$
$$H_b = - \sum_r P_b(r) \log_2 P_b(r) \qquad (2.5)$$

Therefore, the entropy per pel, H_{pel}, and the maximum theoretical compression factor, Q_{max}, for a given set of run-length values are given by the expression:-

$$Q_{max} = \frac{1}{H_{pel}} = \frac{\bar{r}_w + \bar{r}_b}{H_w + H_b} \qquad (2.6)$$

H_{pel} in Eqn. (2.6) is usually higher than H_{pel} for Capon model given in Eqn. (2.1) and indicates that the latter model includes some of the higher order dependencies between successive pels of the same colour.

Many different codes have been designed for use in run-length coding. However, the problem of assigning codewords to the run-lengths in an optimal way can be solved by using Huffman's procedure$^{(36)}$, which minimises the average codeword length for a given probability distribution of the run lengths. Since the probability distribution is different for white and black runs, individual codeword sets must be provided for coding the white and the black runs. The distinguishing feature of the Huffman code is that the codeword lengths are related to the frequencies of the source runs, the more frequent runs being assigned the shorter codewords. The size of the source alphabet (all possible run-lengths) rules out the use of pure Huffman code for run-length coding. Fortunately, it turns out that there exist other variable length codes which perform nearly as well and which are easily implemented in hardware, for example, the A-codes and the B-codes(37). The A-codes are a class of codes for which each run-length is assigned a binary codeword consisting of one or more fixed length blocks. For A_N, the blocks have N bits and the possible codeword lengths are multiples of this number. The A-codes are nearly optimal for exponentially distributed run-lengths, and can be made adaptive on a line-to-line basis, where each scan line is prescanned to determine the mean lengths of the white and black runs separately, and then coded using the corresponding optimal block length. The B-codes, on the other hand, are a class of codes for which the codeword length increases roughly as the logarithm of the run-length. They have been claimed(37) to be very nearly optimal for the run-length distribution occuring in practice. As in the case of A-codes, the codewords for the B-codes consist of one or more fixed length blocks. The block length for B_N is $N + 1$ bits. The first bit of each block, which indicates the colour of the run, is a 'continuation bit' and the following N bits are the'information bits'. The code is uniquely decodable, though not instantaneous.

It has been reported that the redundancy (defined as the ratio of redundant to total information) for optimal block length A-codes is about 50 percent while the B-code with the block length of 2, i.e. the B_1 code, has a redundancy of 30 percent. Both codes are easily implementable since the rules for constructing the codewords are simple.

Another code based on run-length coding is the intermediate ternary code (ITC)(38-39). The main idea behind this code is the use of a ternary representation of both black and white runs as an intermediate step. In the ITC scheme, the binary numbers representing the values of the run-lengths are converted to a ternary state which distinguishes between black and white runs. Ternary pentades are then converted into binary octades which form the coded data. In its basic form, this code does not require a code table and therefore its implementation is simple. Also, since the codewords are all eight bits long, there can be no loss of codeword synchronisation. The performance of ITC has been evaluated and on average, it is better than either the B-codes or the A-codes.

The remaining run-length codes are mainly based on the use of some form of dictionary look-up table. Amongst the more prominent ones are the Easily Decodable Run-length code (EDC)(40), the Truncated Huffman code (THC)(41), the Reduced Huffman code (RHC)(42) and the Modified Huffman code (MHC)(42-43). The design of these codes makes use of the statistical probability distribution in an optimum manner and yet keeps the complexity of implementation to a minimum. In other words, the memory requirement for the codeword table is considerably less than that of Huffman coding(36) without losing much of the compression gain of the latter. Comparisons have been made between these codes(41-42) with the result that the Modified Huffman code performs better, on average, than the others except for EDC. EDC seems to be marginally superior but because the Modified Huffman code has been extensively studied, performs well generally and has been implemented in some commercial machines, it has been adopted as the one-dimensional facsimile coding standard. A detailed description of MHC will be given in Chapter III.

(B) Two-Dimensional Coding Schemes

The main advantage of two-dimensional coding schemes over their one-dimensional counterparts is that they offer higher data compression factors, especially when the documents are scanned at high resolution, without significantly increasing system cost. Although, the system becomes more vulnerable to transmission errors the decrease in quality is not large enough to discourage their use.

It has been mentioned earlier in the chapter that two-dimensional coding schemes can be classified into three sub-divisions. The first is simultaneous coding of N lines, in which two or more scan-lines are taken and coded together. Coding schemes belonging to this group include Mode Run-length Coding(44-45), Zig-zag Scanning(46) and Cascade Division Coding(47). In Mode Run-length Coding, two scan-lines are simultaneously examined. When black and white documents are considered, there are four possible patterns or 'modes' present. The run-lengths of each mode are encoded using efficient run-length codes matched to the distribution of the 'modes'. This method of coding has been adopted as a standard for the Japanese administrative digital facsimile communication system (ADMIX)(48) and is reported to have as good a compression as line-by-line coding schemes where the K* factor equals two. The Kalle-Infotec Code(49), which assigns codewords having lengths between two and eight bits according to the local statistics of the facsimile picture and adaptive vector coding(50), in which more than two scan-lines are taken to form vectors, are some of the variations of Mode Run-length Coding. Coding by zig-zag scanning either in a sawtooth-like or wave-like manner(51) is another way of processing two-scan-lines simultaneously. Here, an ordinary binary image with L lines and M pels/line can be regarded as a binary image with L/2 lines and 2M pels/line. In other words, zig-zag scanning effectively transforms the original image into a version which is elongated sideways. One dimensional run-length coding schemes can then be applied to these elongated scan-lines. In Cascade Division Coding(47), two successive scan-lines are stored in a buffer and then adaptively coded block by block. Blocks which do not contain black pels are assigned the prefix code '0', whilst blocks containing black pels are further sub-divided and assigned a prefix '1'. The process continues until blocks of four pels are reached where, if the block contains black pels, the binary pattern of the block together with a prefix is transmitted. Although, the scheme has been independently proposed by Usubuchi et.al.(47), the method is similar to the block coding techniques of Kunt, Johnsen and deCoulon(34-35).

* defined on page 36

Line-by-line sequential coding schemes belong to the second category of two-dimensional coding techniques, which generally produce higher compression ratios than simultaneous coding of N lines. However, since each coded line is reconstructed at the receiver utilising the already regenerated preceding lines, error propagation in the vertical direction is inevitable in the event of channel errors. This can be overcome by inserting a one-dimensionally coded line every K lines, thereby limiting vertical error propagation. Line-by-line sequential coding is currently recognised to be the most favourable approach to two-dimensional coding.

The earliest line-by-line coding algorithm was a line-difference code proposed by Laemmel(4). Here, two preceding lines of image data are used to predict the location of both ends of a run on the basis of preceding runs of the same colour. Deviations from these prediction are expressed in terms of the number of pels needed to locate the run ends correctly. Huang proposed another line-difference algorithm called predictive differential quantizing (PDQ)(52-53) in which the differences between corresponding run-lengths of successive scan-lines are transmitted. Basically, in PDQ, the changes in white to black transition locations (Δ') and the changes in black run-lengths (Δ'') together with messages (New start and Merge) indicating the start and the end of a black area are coded and transmitted. Later, Yamazaki et. al. proposed a modified version of PDQ called Relative Address Coding (RAC)(54). In RAC, transition elements, i.e. those pels at a transition from black to white or vice-versa in the horizontal direction, are first defined. RAC is best described with the aid of a diagram as shown in Fig. 2.5 in which two-scan-lines and the six transition elements P, P', Q, Q', R and R' are depicted.

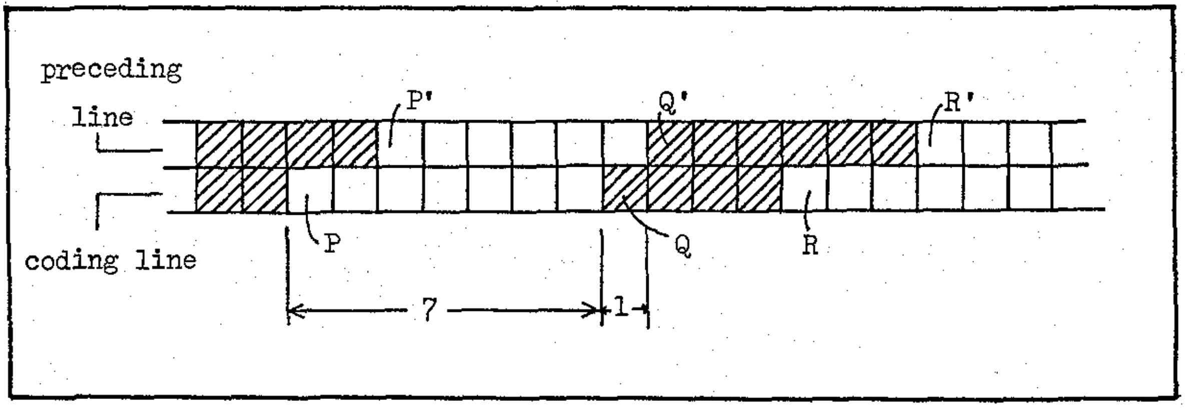


Fig. 2.5 - Principle of Relative Address Coding

Referring to Fig. 2.5, suppose transition element Q is to be coded. The encoder selects two reference transition elements which have already been encoded. They are transition elements P and Q' which are the preceding transition element on the same line and the transition element on the reference line that has the same colour as that of transition element Q respectively. The coding procedure of transmitting the shorter of two distances, either that between Q and P or that between Q and Q'. If both distances are equal to one, then the distances QQ' is transmitted. However, if both distances are equal to M where M>1, then the distance PQ is transmitted. If QQ' is selected for transmission, then a sign is included to indicate that Q is to the left or right of Q'. In order to encode this distance, variable length coding is used.

As a further development of RAC and PDQ, another line-by-line coding scheme called Edge-Difference Coding (EDIC)(55) has been proposed. Offering greater coding efficiency than RAC and PDQ, EDIC makes use of pairs of transition elements which are classified into three different states. In state S_1, one of the transition elements of the pair occurs on the preceding line and the other on the current line. In state S_2, both of the transition elements of the pair occur on the preceding line and state S_3, both the transition elements of the pair occur on the current line. The states of transition element pair are coded to some coding rule. Further details of EDIC can be obtained in (56).

A 'marriage' between RAC and EDIC has given birth to the Relative Element Address Designate (READ)(57) coding algorithm. A feature of RAC, in which the reference pels for encoding the transition elements are adaptively selected, and a feature of EDIC, in which coding is performed by classifying a pair of transition elements into three states, are both employed in READ. The READ coding algorithm was the Japanese contribution to the CCITT SG XIV as a two-dimensional coding standards. However, the CCITT modified the original READ algorithm (called the Modified READ code)(43) in order to simplify its implementation without significantly changing its compression efficiency. A detailed description of the Modified READ code is given in the next Chapter.

Other variations of line-difference coding are those described by Woods(58) and Mitchell and Goertzel(59). Woods's technique called Two-dimensional Delta-Mod Facsimile coding(58), is information lossy. If the run-length difference between two successive scan-lines is less than or equal to a threshold, the difference is quantised to a value of -1, 0, or +1 and coded as '00', '01' and '10' respectively. If not, run-length coding is used with a prefix '11'. Mitchell and Goertzel's technique is very similar to the RAC scheme and is the basis for IBM's proposal to the CCITT as a two-dimensional standard .

During the period when Laemmel(4) was reporting line-difference coding, Elias(5) was working on a generalised predictive coding scheme for information sources. Elias described the use of previous data samples to predict the coming data samples and then encoding only the prediction errors. Thereafter, Wholey(11) explored the application of predictive coding to binary image data in which the value of the current pel is predicted based on previous pels on the same line and on the previous lines, hence generating a prediction error domain. Preuss(60) later described what can be regarded as one of the most significant investigations on predictive coding of facsimile signals. His scheme is motivated by the idea of extending Capon's first order Markov model to a two-dimensional Nth order Markov model. In this model, the present pel is assumed to be statistically dependent on N previously transmitted pels.

Data compression does not directly result from the prediction itself but rather when the resultant error signal is encoded using a suitable code. Run-length coding matched to the statistics of the prediction errors [redacted] is normally employed(26). However, Preuss(60) also suggested that if the prediction errors are separated according to each of their source states and then run-length coding matched to each source state is used, higher compression can be readily obtained. He also proposed a simplified model in which only two source states are generated in order to simplify implementation. As a result of this, other schemes based on Preuss's work were proposed. Amongst the more important ones are the Ordering schemes(61-64), and the Classified Pel Pattern method(65-66). In the basic ordering scheme, prediction of the present element is made using the surrounding previously transmitted picture elements and then the states used for prediction are classified into two categories, 'good' or 'bad'. 'Good' states are those for which the probability of the prediction being in error, conditional on that state, is less than or equal to a given threshold. All other states are 'bad'. The values of the prediction errors corresponding to the 'good' state are placed towards the left hand side of a memory equal to the size of a scan-line while 'bad' states are placed towards the right hand side of the memory. The reordered prediction errors are then coded using run-length coding. Results reported by Netravali et. al.(64) show that, on average, ordering techniques reduce the number of bits required for transmission by about 41 percent when compared with the one-dimensional Modified Huffman code.

The basic idea of the Classified Pel Pattern method (CLAP) is similar to Preuss's technique. The prediction errors are classified into two modes: the Strong and the Weak mode, where the probability of correct prediction is larger in the Strong, and smaller in the Weak, mode. The prediction errors in these two modes are then regarded as produced by a memoryless binary information source, after which an efficient coding scheme is employed. The CLAP technique has been reported to be even more efficient than RAC and EDIC(51,66) and equipment based on this scheme has been on the market since 1977.

The third category of two-dimensional coding schemes is area coding algorithms, In contrast to simultaneous coding of N lines and line-by-line sequential coding, area coding algorithms are truly two-dimensional and assume the existence of all (or, at least, a meaningful part) of the image in a random-access memory which can be coded independently of other areas. One of the more well-known techniques is block coding(34) in which picture elements are grouped into blocks of size n x m, where n and m are the number of pels in the horizontal and vertical directions, respectively. These blocks are then coded according to their probabilities of occurrence, using short codewords for the most likely, and longer codewords for the less likely, block configuration, so that on average compression is obtained. Huang and Hussain(68) later explored the optimal block size for one-dimensional block coding. A tutorial review of block coding has recently been published by Kunt and Johnsen(22). Contour coding is another technique which can be regarded as a truly two-dimensional approach to binary image compression. Schreiber et. al.(70) pointed out that for binary images, all of the information is contained in the outlines of objects. Given these outlines or 'contours', filling in the spaces that they enclose regenerates the original image. The implementation of this idea is somewhat involved, especially as a one-to-one transformation is desired between any possible image and its contours. Morrin(71) developed such an algorithm, based on coding the outer edges of all objects and holes in an image. Pattern recognition, in particular optical character recognition (OCR), can sometimes be viewed as a powerful form of data compression. However, such a system is non-information preserving, in which a completely one-to-one transformation does not exist. Although this section deals with only information preserving schemes, pattern recognition belongs to a class of area coding algorithms and its impact on facsimile coding has recently been acknowledged by Pratt et. al.(32). Ascher and Nagy(72) were amongst the first to recognise the potential of OCR for printed text. They proposed a hybrid compression and cognition scheme in which characters are saved (i.e. become prototypes) if they are not highly correlated with any of the previous saved characters. Characters which are highly correlated are transmitted using only the identification code for the saved prototype. Pratt et. al.(86) made measurements on the application of such system to typewritten data. They used a segmentation algorithm for their OCR scheme that enclosed all objects within a minimum size rectangular block. When objects are assumed to be unrecognisable, they are coded accordingly, whilst recognised characters are assigned address bits to locate their position. Based on this scheme, a Combined Symbol Matching (CSM)(30-32) technique was later proposed which is a refined version of Ascher's and Nagy's scheme. The CSM technique possesses the advantage of both symbol recognition and run-length coding. In operation, a symbol blocking operator isolates valid alphanumeric characters and document symbols. The first symbol encountered is placed in a library, and as each new symbol is detected, it is compared with each entry of the library. If the comparison is satisfied to within a given tolerance, the library identification code is transmitted along with the symbol location coordinates. Otherwise, the new symbol is placed in the library and its binary pattern is transmitted. Non-isolated symbols are left behind as a residue, and are coded by run-length coding. OCR systems are generally difficult to realise practically but the advantage in performance is significant. It is in this area that significant compression performance improvements will probably be made in future.

2.4. OTHER RELATED CODING TECHNIQUES

In this section, some of the coding techniques closely related to, but not directly connected with, ordinary facsimile coding are discussed. Some of the problems involve the coding of dithered binary pictures and also the coding of newspaper pages, which are normally scanned at very high resolution.

Dithering is an image processing technique which creates a two-level picture that gives the illusion of a multilevel picture by appropriately arranging the spatial density of the two levels (usually black and white) on the picture(73-74). The dithering technique consists of comparing a multilevel image with a position dependent threshold and setting pels to white when the input signal exceeds the threshold. Other pels are set to black. The matrix of threshold values (called the dither matrix) is repeated over the entire picture to provide the threshold pattern for the whole image. Various techniques have been developed for the purpose of displaying continuous-tone images on a bilevel display(75). Netravali et. al.(76) studied the use of the ordering techniques to dithered pictures. By predicting the present picture element based on four previous elements and rearranging the relative order of the picture elements in such a way as to increase the average white and black run-lengths, bit rates were decreased to about 0.20 bits per dithered pel(76). Johnsen(77), later proposed a new predictor for coding dithered pictures which gives a performance, on average, 20 percent better than that of Netravali et. al (76). The use of the CCITT one- and two-dimensional coding standards on dithered images in conjunction with ordering techniques has been studied by Johnsen and Netravali(78). Results obtained(78) show that a compression of about 3.5 is possible.

The transmission of a newspaper page using analogue facsimile can take as long as about five minutes via a 48 Khz channel and the need for digital data compression techniques in that application is apparent. Furthermore, since the newspaper industry is becoming more automated (binary pattern processing by computer), it is even more desirable to transmit newspaper pages via digital channels.

The use of ordinary facsimile coding techniques has been shown(79) to give poor performance on newspaper pages due to their very different statistical properties when compared with printed text. Newspaper pictures are normally composed of a periodic dot structure which is not found in ordinary facsimile documents. Using this property, Usubuchi et. al.(79) proposed a new technique for the compression of newspaper pages by developing a predictor using the Dth previous pel and its neighbouring elements as well as those pels close to the picture element to be predicted, where the distance D represents the period of the dot structure. Using this technique, it has been shown that the transmission time can be reduced from five to about 1.8 minutes[79].

2.5. THE EFFECTS OF ERRORS AND METHODS FOR THEIR MINIMISATION

Efficient source coding of any data stream inevitably increases the sensitivity of the compressed data to channel errors. In facsimile, channel error effects become worse as higher data compression is achieved. The effect of errors is different for different coding schemes, being more drastic for area coding algorithms like CSM and more manageable for one-dimensional run-length coding. The increased sensitivity to channel errors is partially compensated for by the fact that after compression, fewer bits represent an image and hence there are fewer chances of the signal being corrupted by noise. Although this is so, the occurrence of a single error will still have quite a devastating effect on the resulting copy.

The effect of channel noise on one-dimensional run-length coding scheme is normally characterised by a spatial shift of the subsequent picture information due to loss of synchronisation. However, the damage caused by an error can be confined to the scan-line in which the error occurs by transmitting a special synchronising sequence called the end-of-line (EOL) codeword at the end of each coded line. This codeword is unique since it consists of a sequence of digits which cannot occur naturally anywhere in a scan-line of coded data. It can therefore be easily recognised and although a coded line may be damaged by transmission disturbances, all the subsequent lines can be correctly received and decoded. There is an additional advantage in that the receiver can sum the decoded run-lengths between two EOLs and if the value is different from the nominal scan-line length, an error can safely be assumed to have occurred. The receiver can than employ an error concealment technique to reduce the subjective effect of the damage to the document.

Although the inclusion of EOL codewords prevents propagation from one line to another in a one-dimensional run-length coding scheme, by forcing resynchronisation at the end of each coded line, there are problems associated with channel errors occurring in EOLs. Three events are often considered. They are lost EOL, premature EOL and false EOL. Lost EOL occurs when an error corrupts the EOL codeword in such a way that it cannot be recognised. The effect is loss of a scan-line, and this can be reduced by applying error concealment techniques. Premature EOL occurs when an error happens to occur in the FILL bits (See Chapter III and ref. (43)). It creates a spurious or false EOL and can be easily recognised since coded lines with fewer bits than specified by the minimum scan-line time (MSLT)(43) are produced. In the case of false EOL, the decoder recognises two lines instead of one, both showing the wrong number of pels per line. As a result, an extra line is added to the document.

The introduction of an EOL codeword is the first measure for limiting the effects of an error to only one scan-line. However, visually to improve the quality of the resultant document, error concealment techniques may be applied to the erroneous lines. For example:-

(a) replacing the damaged line by an all white line

(b) repeating the previous line

(c) printing the damaged line

(d) using a line-to-line processing or correlation technique

to construct as much of the line as possible.

The effect of errors sometimes appears in the form of a long streak of black runs which is very noticeable and disturbing. In cases like this, it is preferable to adopt methods (a) and (b) to improve visual quality. Sometimes only a small displacement in the characters occurs and for those case, method (c) is used, i.e. the errors are left alone. Usually only a few of the codewords of a scan-line are disturbed, and, instead of replacing the complete erroneous line either by a white line or the previous line, much of the scan-line can be retained if the erroneous zones can be located. Since there is high line-to-line correlation between scan-lines, this property can be used to localise the erroneous area. Rothgordt(80) employed this technique by measuring the correlation between groups of pels on the damaged line and corresponding groups on the adjacent lines above and below. Poncin and Botrel(81) also used a similar scheme. Once the erroneous areas have been localised, methods (a) and (b) can then be employed on those areas only. Eto et. al.(82) developed a different strategy in localising the erroneous zones by using the instantaneous change in the received carrier, in their case, 4-phase differential PSK. Subjective results(82) show that there is a further improvement in quality compared with the case when methods (a) and (b) are employed outright.

In the case of two-dimensional line-by-line sequential coding schemes, the resulting effect of transmission error is error propagation both in the horizontal and vertical directions. Vertical error propagation can be limited to only a few scan-lines by transmitting, every K lines, a one-dimensional run-length coded line. For normal and high resolution, respectively, the CCITT has standardised the values of K as two and four, to prevent further obliteration of the received document. For example, if K is four, the occurrence of an error in the coded data just after a one-dimensionally coded line would cause vertical error propagation throughout the next K-1 (in this case, three lines). The usual way of handling this kind of error propagation is to replace the erroneous lines with the last correctly decoded line, but this will create annoying 'elongated' characters in the vertical direction. One way of overcoming this is to replace the first two erroneous lines with the previous correctly decoded line and the third line is then replaced with the next one-dimensionally coded line. In other words, up to five lines need to be stored. Another way is to localise the erroneous areas and only replace the affected areas using the corresponding areas of the previously decoded line. This will improve the resulting copy dramatically. The overall results of error concealment techniques will be less drastic if channel errors happen to occur on the second or third scan-lines after a one-dimensionally coded line.

The effect of channel errors on area coding algorithms such as block coding, contour coding etc. is more catastrophic for the resultant document than in the case of one-dimensional and line-by-line sequential coding, for the occurrence of an error will completely destroy the total document content. (See Chapter VII, for some pictorial illustrations for the ABLC scheme). Chen(83) studied the effect of transmission errors on the Combined Symbol Matching facsimile coding scheme and found that a single error degrades the quality beyond recognition. Simple error concealment techniques will not be effective and a different error protection strategy is required for such a system. He proposed a tailored error protection scheme in which the most sensitive code elements of the source codes are isolated in order to minimise a catastrophic loss of codeword synchronisation. This is done at the expense of additional overhead bits which effectively increase transmission time.

In addition to error concealment techniques and error correction schemes operating at the receiver, detection of errors and re-transmission of blocks of data in error using an automatic repeat request (ARQ) system and/ or correction using a forward-acting error correction code (FEC) can be incorporated. Such schemes are suitable for systems operating in error-prone environments and are also applicable to systems which, by their nature, are extremely sensitive to channel noise, CSM for example.

ARQ error protection schemes basically operate by detecting errors in the data and then requesting retransmission. The transmitted data is grouped into 'blocks', and for each block, a 'checkword' is calculated. The checkword algorithm is designed such that the checkword's exact value is a very sensitive function of the bit pattern of the block to be transmitted. When the block is received, an identical algorithm re-calculates the checkword using the possibly corrupted data block, and compares it with the transmitted checkword. If the checkwords are identical, it is assumed that no errors have occurred, otherwise retransmission is requested. ARQ systems have the advantage of being very reliable and insensitive to changes in the channel error rate but, however, their inclusion leads to a reduction in the effective transmission rate.

FEC codes, on the other hand, have higher effective transmission rates but must be specifically designed depending on the type of errors likely to be encountered in the channel. There are several efficient schemes available for forward error correction purposes(84). One of the more familiar schemes used to correct random errors is the use of a convolutional coder together with a Viterbi decoder. The error correction capability of this system is dependent upon the complexity of the decoder and the quantity of overhead bits one is willing to allow. One of the most common schemes used to correct burst errors is the Reed-Solomon coder/decoder. Recently, an implementation of the Reed-Solomon code using a microprocessor has been demonstrated(85) which has the capability of correcting a single burst error spread over up to 17 bits in a block of 255 bytes.

Although ARQ and FEC methods are effective in protecting the data from channel errors, they do add extra cost to the equipment, and increase the overall transmission time.

2.6. CONCLUSION

In this chapter, the subject of facsimile has been introduced from the historical point of view, leading up to sophisticated present day digital facsimile systems. At the beginning of the chapter, the advantages of digital facsimile systems over their analogue counterparts are emphasised, and problems related to transmission impairment, and the ability to transmit faster using digital techniques, are discussed. The chapter also surveys the current 'state of the art' in facsimile coding methods, for both non-information, and information, preserving schemes. The survey is by no means exhaustive but gives an adequate indication of the techniques available. Areas which have not been dealt with in a detailed way are those pertaining to optical character recognition and the coding of newspaper and dithered images.
A digital facsimile system is not fully characterised unless its response to channel errors is documented, and some of the common error protection schemes currently used in conjunction with digital facsimile are discussed here. Finally, the change over to digital data networks from the present use of voice grade telephone networks, makes digital facsimile more convenient, and can be expected to stimulate further development in equipment, systems and applications.

CHAPTER III

DIGITAL FACSIMILE CODING STANDARDS

3.1. INTRODUCTION

Although the concept of facsimile has been in existence for more than a century, it is only in the last decade that much success has been achieved in standardising its operation. For many years, facsimile communications was utilised by huge multinational companies like Western Electric, United Press International (UPI) and Nippon Telegraph and Telephone Public Corporation (NTT). Most of their special purpose facsimile equipment, used on leased lines, was of the analogue type, corresponding to Group 1 (G1) machines (Recommendation T.2) of the CCITT standard, which require six minutes to send an A4 size page over the voice-grade channel.

In the early seventies, there was a sudden revival of facsimile communication, particularly in Japan, where there was a growing need for efficient transmission of Kanji characters. A new era of facsimile communications had emerged, and in 1972 NTT opened its Public Switched Telephone Network (PSTN) for data and facsimile communications. The growth of facsimile comunication was rapid and over these years, a new type of analogue equipment, standardised by the CCITT under the category of so-called Group 2 (G2) machines (Recommendation T.3), appeared, which was capable of sending an A4 size page in three minutes over the voice-grade channel. These added a new dimension to facsimile applications and resulted in widespread use of facsimile equipment in such areas as government agencies, financial businesses and manufacturing industry. There was now much interest in facsimile communications, especially by business concerns, and this led to the need for further reduction in transmission time and cost, and the ability to communicate between companies with equipment having an improved resolution. However, faster transmission could not readily be realised using analogue techniques because of the restricted bandwidth of voice band telephone circuits. These requirements prompted the development of a new type of facsimile machine, capable of sending an A4 size page within a minute using digital processing and redundancy reduction techniques.

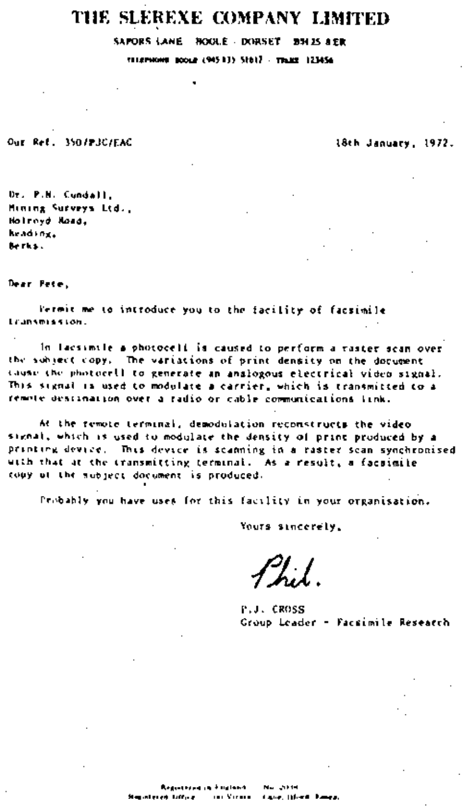
By 1974, CCITT Study Group XIV was laying the foundation for the establishment of a new international Recommendation for digital facsimile. In general, equipment of the type considered here is classified as Group 3 apparatus and is defined within the CCITT as:-

> "Group 3 : Apparatus which incorporates means for reducing the redundant information in the document signal prior to the modulation process and which can achieve a transmission time of about 1 minute for a typical document of ISO A4 size via a telephone type circuit. The apparatus may incorporate bandwidth compression of the line signal." (104)

At the same time, several digital facsimile coding techniques were proposed but the specifications applied by manufacturers making commercially marketed facsimile terminal units still remain mutually incompatible. In fact, the question of 'interbrand' comptibility had taken a decided step backwards.(1) Hence, there was an urgent need to establish an international standardisation for digital facsimile equipment.

In 1976, T.3, the Group 2 Recommendation and T.30(105), the Protocol Recommendation, were agreed by the Sixth Plenary Assembly of the CCITT. With T.3 and T.30 settled, greater emphasis was now directed toward digital facsimile so that by the end of 1977, T.X (a temporary title for the Group 3 Recommendation) was provisionally agreed. Towards the end of 1979, a new Recommendation (T.4) was drafted for Group 3 type apparatus with the aim of achieving compatibility between digital facsimile machines connected to the PSTN. It was only through widespread collaboration and agreement between many companies and national telecommunication administrations, under the auspices of the CCITT, that made it possible to draft a satisfactory Recommendation (T.4). The CCITT also provided eight reference documents, as shown in Fig. 3.1, for assessing the computer simulation results required to arrive at a satisfactory recommendation. By now, the draft Recommendation T.4 should have been ratified by the CCITT.

In order to achieve the sub-minute transmission time, source encoding methods are employed to reduce the amount of redundant information required to represent an image. Included in the draft Recommendation T.4 is a one-dimensional run-length source encoding



THE SLEREXE COMPANY LIMITED

SAPORS LANE · BOOLE · DORSET · BH 25 8 ER
TELEPHONE BOOLE (945 13) 51617 · TELEX 123456

Our Ref. 350/PJC/EAC 18th January, 1972.

Dr. P.N. Cundall,
Mining Surveys Ltd.,
Holroyd Road,
Reading,
Berks.

Dear Pete,

Permit me to introduce you to the facility of facsimile transmission.

In facsimile a photocell is caused to perform a raster scan over the subject copy. The variations of print density on the document cause the photocell to generate an analogous electrical video signal. This signal is used to modulate a carrier, which is transmitted to a remote destination over a radio or cable communications link.

At the remote terminal, demodulation reconstructs the video signal, which is used to modulate the density of print produced by a printing device. This device is scanning in a raster scan synchronised with that at the transmitting terminal. As a result, a facsimile copy of the subject document is produced.

Probably you have uses for this facility in your organisation.

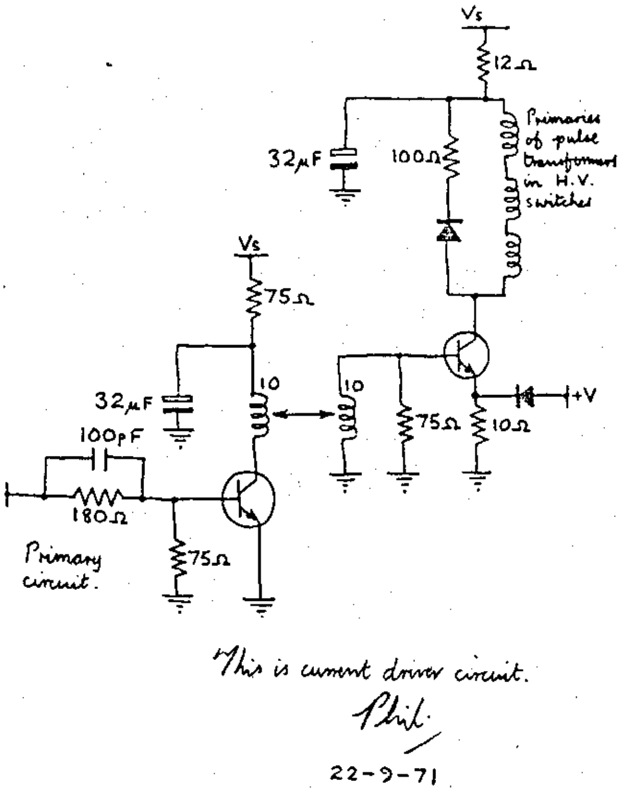
Yours sincerely,

Phil.

P.J. CROSS
Group Leader - Facsimile Research

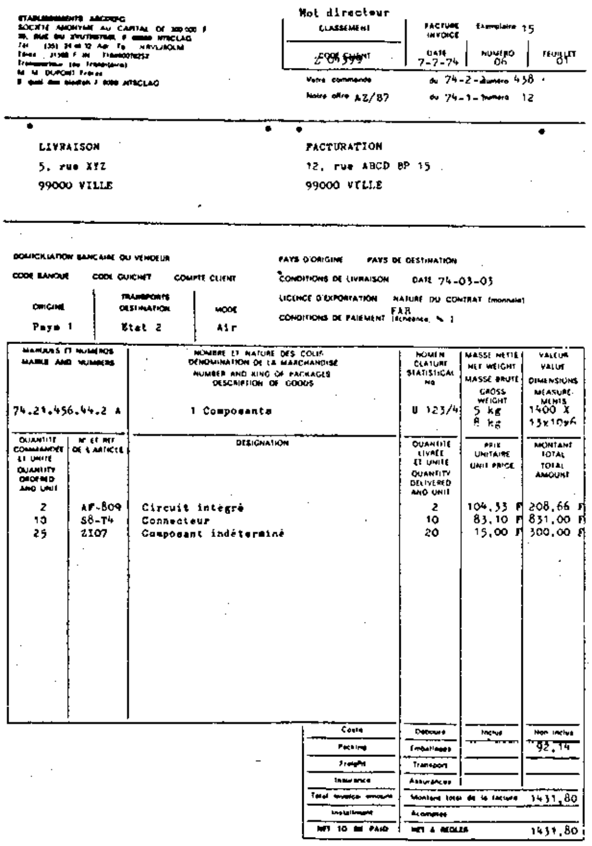
Registered in England No. 2038
Registered Office 60 Vicara Lane, Ilford Essex.

Document 1



This is current driver circuit.
Phil.
22-9-71

Document 2



ETABLISSEMENTS ABCDEFG
SOCIETE ANONYME AU CAPITAL DE 300 000 F
30, RUE DU [illegible] F [illegible] NTSCLAG
Tél (35) 24 et 12 [illegible]
Télex 31 500 F [illegible]
Transporteur (ou Transitaire)
M. M. DUPONT Frères
8 quai des [illegible] F [illegible] NTSCLAG

Mot directeur

CLASSEMENT	FACTURE INVOICE		Exemplaire 15
CODE CLIENT Z 04 599	DATE 7-7-74	NUMERO 06	FEUILLET 01

Votre commande du 74-2- numéro 438
Notre offre AZ/87 du 74-1- numéro 12

LIVRAISON
5, rue XYZ
99000 VILLE

FACTURATION
12, rue ABCD BP 15
99000 VILLE

DOMICILIATION BANCAIRE DU VENDEUR
CODE BANQUE CODE GUICHET COMPTE CLIENT

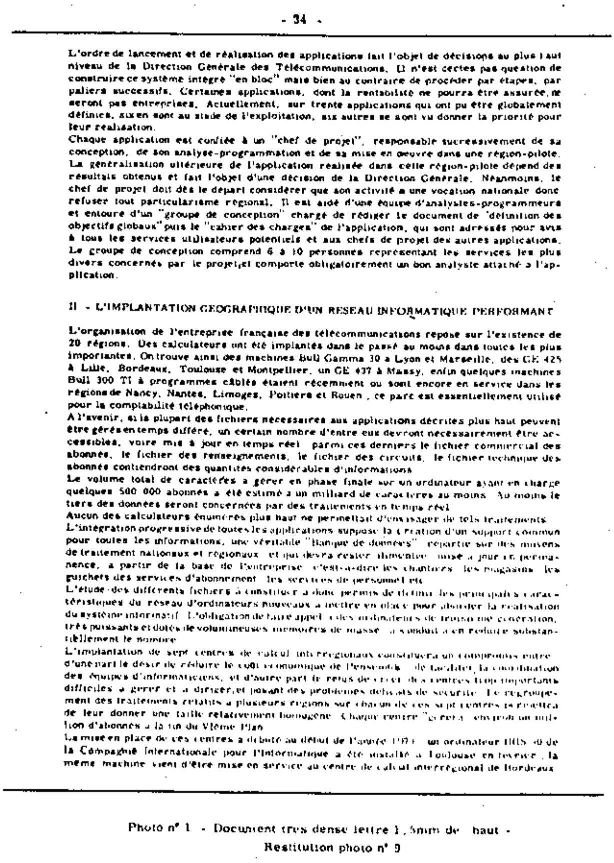
ORIGINE	TRANSPORTS DESTINATION	MODE
Pays 1	Etat 2	Air

PAYS D'ORIGINE PAYS DE DESTINATION
CONDITIONS DE LIVRAISON DATE 74-03-03
LICENCE D'EXPORTATION NATURE DU CONTRAT (monnaie)
CONDITIONS DE PAIEMENT (échéance, %) FAR

MARQUES ET NUMEROS MARKS AND NUMBERS		NOMBRE ET NATURE DES COLIS DENOMINATION DE LA MARCHANDISE NUMBER AND KIND OF PACKAGES DESCRIPTION OF GOODS	NOMEN CLATURE STATISTICAL No	MASSE NETTE NET WEIGHT MASSE BRUTE GROSS WEIGHT	VALEUR VALUE DIMENSIONS MEASUREMENTS
74.21.456.44.2 A		1 Composants	U 123/4	5 kg 8 kg	1400 X 15x10x6
QUANTITE COMMANDEE ET UNITE QUANTITY ORDERED AND UNIT	No ET REF DE L'ARTICLE	DESIGNATION	QUANTITE LIVREE ET UNITE QUANTITY DELIVERED AND UNIT	PRIX UNITAIRE UNIT PRICE	MONTANT TOTAL TOTAL AMOUNT
2	AF-809	Circuit intégré	2	104,33 F	208,66 F
10	58-T4	Connecteur	10	83,10 F	831,00 F
25	2107	Composant indéterminé	20	15,00 F	300,00 F

Costs	Débours	Inclus	Non inclus
Packing	Emballages		92,14
Freight	Transport		
Insurance	Assurances		
Total invoice amount	Montant total de la facture		1431,80
Instalment	Acomptes		
NET TO BE PAID	NET A REGLER		1431,80

Document 3



- 34 -

L'ordre de lancement et de réalisation des applications fait l'objet de décisions au plus haut niveau de la Direction Générale des Télécommunications. Il n'est certes pas question de construire ce système intégré "en bloc" mais bien au contraire de procéder par étapes, par paliers successifs. Certaines applications, dont la rentabilité ne pourra être assurée, ne seront pas entreprises. Actuellement, sur trente applications qui ont pu être globalement définies, six en sont au stade de l'exploitation, six autres se sont vu donner la priorité pour leur réalisation.

Chaque application est confiée à un "chef de projet", responsable successivement de sa conception, de son analyse-programmation et de sa mise en oeuvre dans une région-pilote. La généralisation ultérieure de l'application réalisée dans cette région-pilote dépend des résultats obtenus et fait l'objet d'une décision de la Direction Générale. Néanmoins, le chef de projet doit dès le départ considérer que son activité a une vocation nationale donc refuser tout particularisme régional. Il est aidé d'une équipe d'analystes-programmeurs et entouré d'un "groupe de conception" chargé de rédiger le document de "définition des objectifs globaux" puis le "cahier des charges" de l'application, qui sont adressés pour avis à tous les services utilisateurs potentiels et aux chefs de projet des autres applications. Le groupe de conception comprend 6 à 10 personnes représentant les services les plus divers concernés par le projet, et comporte obligatoirement un bon analyste attaché à l'application.

II - L'IMPLANTATION GEOGRAPHIQUE D'UN RESEAU INFORMATIQUE PERFORMANT

L'organisation de l'entreprise française des télécommunications repose sur l'existence de 20 régions. Des calculateurs ont été implantés dans le passé au moins dans toutes les plus importantes. On trouve ainsi des machines Bull Gamma 30 à Lyon et Marseille, des GE 425 à Lille, Bordeaux, Toulouse et Montpellier, un GE 437 à Massy, enfin quelques machines Bull 300 TI à programmes câblés étaient récemment ou sont encore en service dans les régions de Nancy, Nantes, Limoges, Poitiers et Rouen ; ce parc est essentiellement utilisé pour la comptabilité téléphonique.

A l'avenir, si la plupart des fichiers nécessaires aux applications décrites plus haut peuvent être gérés en temps différé, un certain nombre d'entre eux devront nécessairement être accessibles, voire mis à jour en temps réel : parmi ces derniers le fichier commercial des abonnés, le fichier des renseignements, le fichier des circuits, le fichier technique des abonnés contiendront des quantités considérables d'informations.

Le volume total de caractères à gérer en phase finale sur un ordinateur ayant en charge quelques 500 000 abonnés a été estimé à un milliard de caractères au moins. Au moins le tiers des données seront concernées par des traitements en temps réel.

Aucun des calculateurs énumérés plus haut ne permettait d'envisager de tels traitements.

L'intégration progressive de toutes les applications suppose la création d'un support commun pour toutes les informations, une véritable "Banque de données", répartie sur des moyens de traitement nationaux et régionaux, et qui devra rester alimentée, mise à jour en permanence, à partir de la base de l'entreprise, c'est-à-dire les chantiers, les magasins, les guichets des services d'abonnement, les services de personnel etc.

L'étude des différents fichiers à constituer a donc permis de définir les principales caractéristiques du réseau d'ordinateurs nouveaux à mettre en place pour aborder la réalisation du système informatif. L'obligation de faire appel à des ordinateurs de troisième génération, très puissants et dotés de volumineuses mémoires de masse, a conduit à en réduire substantiellement le nombre.

L'implantation de sept centres de calcul interrégionaux constituera un compromis entre : d'une part le désir de réduire le coût économique de l'ensemble, de faciliter la coordination des équipes d'informaticiens, et d'autre part le refus de créer des centres trop importants difficiles à gérer et à diriger, et posant des problèmes délicats de sécurité. Le regroupement des traitements relatifs à plusieurs régions sur chacun de ces sept centres permettra de leur donner une taille relativement homogène. Chaque centre "gèrera" environ un million d'abonnés à la fin du VIème Plan.

La mise en place de ces centres a débuté au début de l'année 1971 : un ordinateur IRIS 50 de la Compagnie Internationale pour l'Informatique a été installé à Toulouse en février ; la même machine vient d'être mise en service au centre de calcul interrégional de Bordeaux.

Photo n° 1 - Document très dense lettre 1,5mm de haut -
Restitution photo n° 9

Document 4

Fig. 3.1 - CCITT test documents

Cela est d'autant plus valable que $T\Delta f$ est plus grand. A cet égard la figure 2 représente la vraie courbe donnant $|\Phi(f)|$ en fonction de f pour les valeurs numériques indiquées page précédente

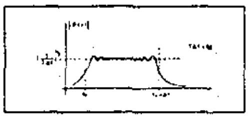


Fig. 2

Dans ce cas, le filtre adapté pourra être constitué, conformément à la figure 3, par la cascade :

— d'un filtre passe-bande de transfert unité pour $f_0 \leq f \leq f_0 + \Delta f$ et de transfert quasi nul pour $f < f_0$ et $f > f_0 + \Delta f$, *filtre ne modifiant pas la phase* des composants le traversant ;

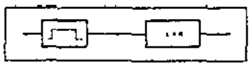


Fig. 3

— filtre suivi d'une ligne à retard (LAR) dispersive ayant un temps de propagation de groupe T_g décroissant linéairement avec la fréquence f suivant l'expression :

$$T_g = T_0 + (f_0 - f)\frac{T}{\Delta f} \quad (\text{avec } T_0 > T)$$

(voir fig. 4)

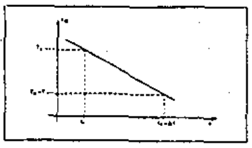


Fig. 4

telle ligne à retard est donnée par :

$$\varphi = -2\pi \int_0^f T_g \, df$$

$$\varphi = -2\pi\left[T_0 + \frac{f_0 T}{\Delta f}\right] f + \pi \frac{T}{\Delta f} f^2$$

Et cette phase est bien l'opposé de $\varphi(f)$,

à un déphasage constant près (sans importance) et à un retard T_0 près (inévitable).

Un signal utile $S(t)$ traversant un tel filtre adapté donne à la sortie (à un retard T_0 près et à un déphasage près de la porteuse) un signal dont la transformée de Fourier est réelle, constante entre f_0 et $f_0 + \Delta f$, et nulle de part et d'autre de f_0 et de $f_0 + \Delta f$, c'est-à-dire un signal de fréquence porteuse $f_0 + \Delta f/2$ et dont l'enveloppe a la forme indiquée à la figure 5, où l'on a représenté simultanément le signal $S(t)$ et le signal $S_r(t)$ correspondant obtenu à la sortie du filtre adapté. On comprend le nom de récepteur à compression d'impulsion donné à ce genre de filtre adapté : la « largeur » (à 3 dB) du signal comprimé étant égale à $1/\Delta f$, le rapport de compression est de $\frac{T}{1/\Delta f} = T\Delta f$

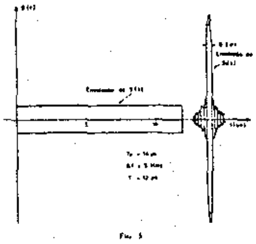


Fig. 5

On saisit physiquement le phénomène de compression en réalisant que lorsque le signal $S(t)$ entre dans la ligne à retard (LAR) la fréquence qui entre la première à l'instant 0 est la fréquence basse f_0, qui met un temps T_0 pour traverser. La fréquence f entre à l'instant $t = (f - f_0)\frac{T}{\Delta f}$ et elle met un temps $T_0 - (f - f_0)\frac{T}{\Delta f}$ pour traverser, ce qui la fait ressortir à l'instant T_0 également. Ainsi donc le signal $S(t)$

Document 5

292 QUESTIONS — COMMISSION XII

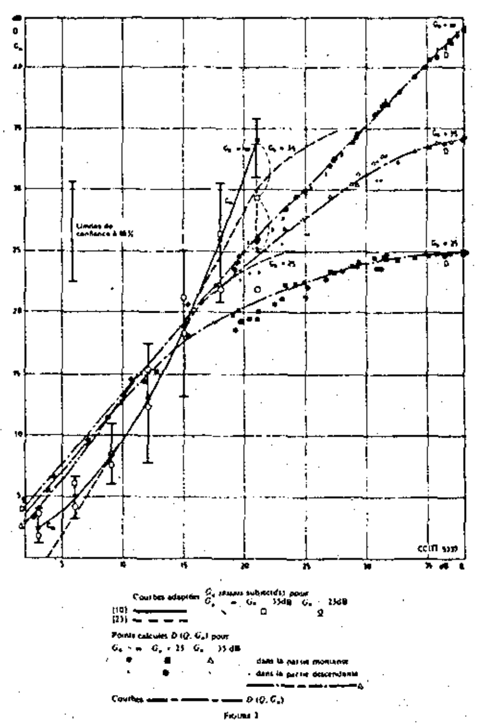
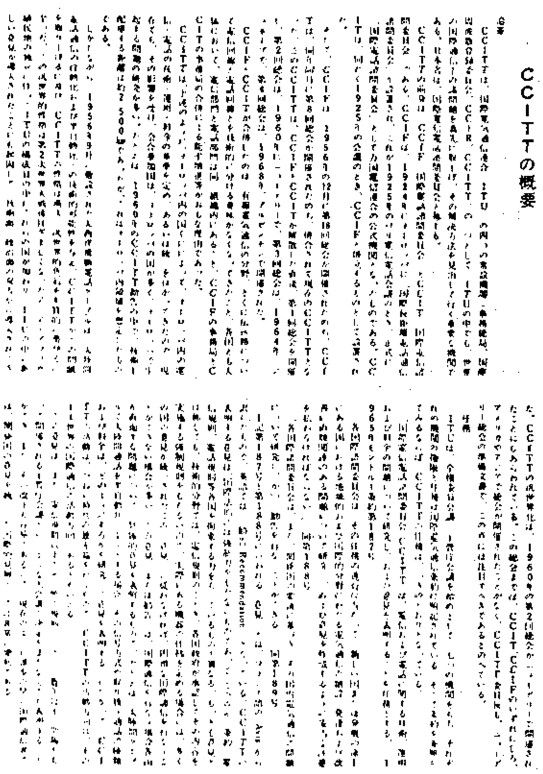


Figure 3

TOME V — Question 18/XII, Annexe 6

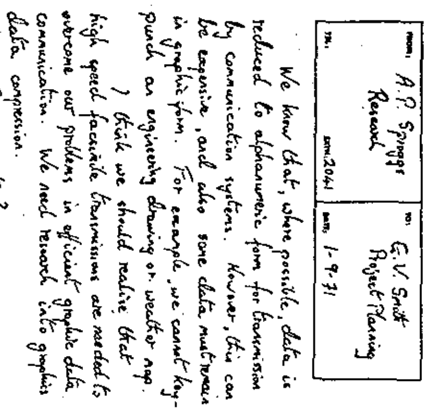
Document 6



CCITTの概要

[illegible]

Document 7



memorandum

FROM: A.P. Spriggs
Research

EXTN. 2041

TO: G.V. Smith
Project Planning

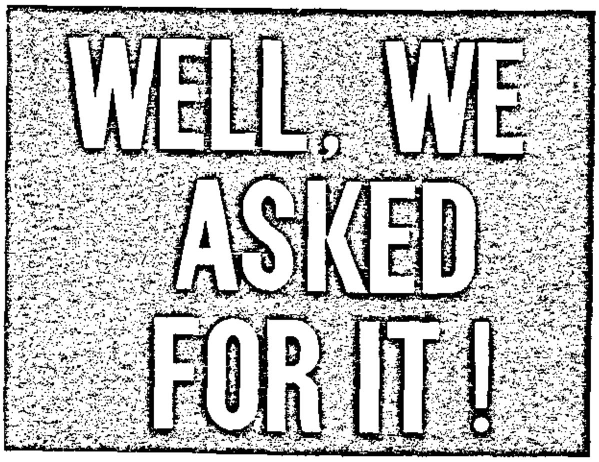
DATE: 1-9-71

We know that, where possible, data is reduced to alphanumeric form for transmission by communication systems. However, this can be expensive, and also some data must remain in graphic form. For example, we cannot key-punch an engineering drawing or weather map.

I think we should realise that high speed facsimile transmissions are needed to overcome our problems in efficient graphic data communication. We need research into graphic data compression.

Any comments?

Albert.



Document 8

Fig. 3.1 (con't) - CCITT test documents

scheme using the Modified Huffman code. This code was agreed by CCITT Study Group XIV as a viable compromise between high values of compression (with great susceptibility to transmission errors), and low implementation costs. Later in 1979, an optional two-dimensional coding scheme was appended to the Recommendation as an extension to the one-dimensional coding scheme. The two-dimensional code allows greater compression efficiency to be achieved for many documents, particularly when they are scanned at twice the normal resolution.

Section 3.2 describes the standard parameters required for Group 3 apparatus while Section 3.3 summarises the Group 3 one-dimensional coding standard. Section 3.4 outlines the criteria used for the choice of the two-dimensional coding scheme and also discusses the scheme in detail.

3.2. STANDARD PARAMETERS FOR GROUP 3 APPARATUS

To achieve international compatibility and to provide interworking between any Group 3 machines over the PSTN, various specifications are required and are given in CCITT Recommendation T.4 and T.30. Recomendation T.4 contains the standard parameters required for Group 3 apparatus, for example, document size, resolution, scanning rate, modulation and source encoding methods. Basically, Recommendation T.30 is not a digital facsimile coding standard. It specifies the 'handshaking' or protocol requirements for Group 1, 2 and 3 apparatus, and defines the possible interactions between these groups. The portion of T.30 which is of most interest to digital facsimile manufacturers and users is the one describing the orderly exchange of data between a calling and called station. Some of the elements covered are, for example, call setup, premessage procedures for identifying and selecting the required facilities, message transmission, post-message procedure and call release. Further information regarding T.30 can be obtained from Ref. (105).

3.2.1 Specifications defined in T.4

(A) Dimension of Apparatus

(i) Input document size:- The equipment should be able to accept ISO A4 (210 x 297 mm) size documents. As an option, documents up to A3 in size may also be transmitted with the same resolution.

(ii) Optical characteristics:- Two level (black and white) images are to be recognised and reproduced.

Multiple shades of grey are not required by the standard, therefore, only one bit per picture element (pel) is needed.

(iii) Resolution:- The standard vertical resolution is 3.85 lines/mm with an optional higher resolution of 7.7 lines/mm.

The primary reason for the choice of 3.85 lines/mm was to allow the Group 3 equipment to be compatible with Group 1 or Group 2 apparatus, as T.2 and T.3 specify the same value. A higher resolution is included to allow higher quality copies to be obtained.

(iv) Sampling density:- Each scan-line on an A4 document is divided into 1728 black and white pels corresponding to a length of 215 mm. The number of pels may be optionally increased to about 2600 to allow documents up to A3 size to be transmitted at the same resolution.

It is observed that the horizontal resolution is nearly twice that of the standard vertical resolution. This is to ensure that no staircase appearances are seen on vertical black and white edges due to the sampling and quantising processes.

(B) Minimum Scan Line Times and Message Format

(i) Time per scan-line:- The recommended standard minimum scan-line times (MSLT) is 20 milliseconds (equivalent to a minimum of 96 coded bits at a transmission rate of 4800 bits/sec). Optional MSLTs of 10 ms, 5 ms and 0 ms are also recognised.

The MSLT specifications allow for mechanical limitations of both the transmitter and receiver sections of some machines. Most printers require a minimum grace period to print the necessary information on a scan-line.

(ii) Error control:- Transmission error control is provided for the Group 3 standard with the addition of a unique End-of-Line (EOL) code at the end of each line of information. The EOL is made up of eleven '0s' followed by a '1'.

The basic premise for having the EOL code is to prevent errors propagating from one scan-line to another if a one-dimensional coding scheme is used. It also allows the decoder to resynchronise and start afresh on a new line.

(iii) Fill bits:- In the event of a scan-line being coded using fewer bits than the specified number, FILL bits are added to extend the time. FILL bits are defined as 'Os' which are inserted between DATA and EOL.

Fig. 3.2 shows the inclusion of FILL bits in order to achieve the required MSLT. The end of document transmission is indicated by six consecutive EOLs which form the return-to-control (RTC) signal.

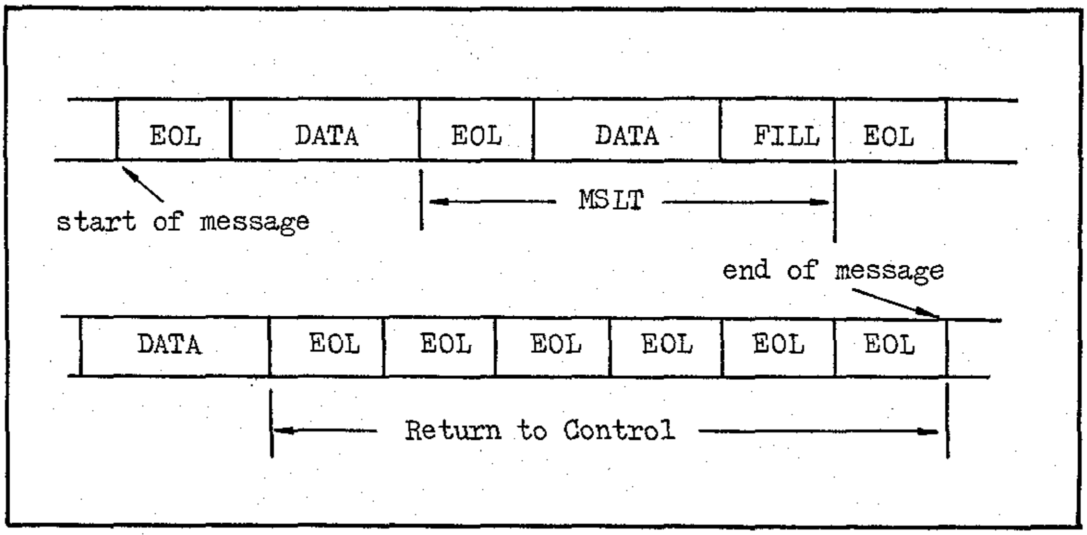


Fig. 3.2 - Message Format

(C) Modulation and Demodulation

It is recommended that Group 3 equipment should use data rates of 4800 bits/sec and 2400 bits/sec when operating on the PSTN. The modulation and demodulation characteristics of a Group 3 device are identical to the characteristics, including scrambling, equalising and timing, of CCITT Recommendation V27 ter. V27 ter. is the standard used to define data transmission on the PSTN at 4800 bits/sec. The use of Group 3 equipment at higher speeds is also recognised. It may operate optionally at 9600 bits/sec if 'good' lines are available using the modulation method defined by CCITT Recommendation V29.

(D) Coding

A statistically derived one-dimensional encoding approach using the Modified Huffman code has been agreed as the Group 3 basic standard. As a further development, a two-dimensional standard, using the Modified READ code, was later included in order to achieve greater compression when documents are scanned at higher resolution. Details of the Modified Huffman code and the Modified READ code will presented in the following sections.

3.3 ONE-DIMENSIONAL CODING STANDARD

It is well-known that Huffman coding[42], which is based on the statistical probability distribution of run-lengths gives optimum results for a one-dimensional coding scheme. However, if the source message has a large range in its probability distribution with some of the messages highly improbable, then the required codebook size becomes unmanageable. A modified Huffman procedure with reduced memory requirement is therefore required.

The Modified Huffman code was first suggested by the Plessey

Company but was later revised by the BFICC (British Facsimile Industry Compatible Committee) and the EIA (Electronic Industries Association) for proposal to the CCITT. The revised version was later accepted by CCITT SG. XIV and is now a basic Group 3 standard. To conform with the CCITT standards whereby A3 size paper can be optionally used, British Telecom extended the code table further to meet this requirement.

Some of the important factors which led to the acceptance of the code by the CCITT were compression efficiency, error susceptibility and complexity of implementation. Since the code design was based on run-length statistics averaged over many typical documents, it is reasonably insensitive to significant changes in the source statistics, and most documents can be transmitted with high efficiency. Furthermore, being a comma free code, in the event of an error disturbing the data bit stream, the code has the capability of self resynchronisation. Thus, the effect of a channel error would be spatially to shift the coded line. By having a unique end-of-line (EOL) code, the disturbance is confined to only one scan-line. Thus, line-to-line error propagation is not possible. Since the code is a modified version of the Huffman code, it requires less memory than the original Huffman code and thus simplifies the problem of implementation.

3.3.1 The One-Dimensional Coding Scheme

In this scheme, each scan-line is regarded as a sequence of alternating black and white lines. All lines are assumed to begin with a white run to ensure that the receiver maintains colour synchronisation. If the first actual run on a line is black, then a white run of zero length is transmitted at the beginning of the line

To represent the black and white runs, separate code tables are used and these are given in Table 3.1. Each code table can represent a value of run-length up to a maximum of one scan-line (1728 pels), and contains two types of codewords: terminating codewords (TC) and make-up codewords (MUC). Runs between 0 and 63 pels in length are transmitted using a single terminating codeword. Run with lengths between 64 and 1728 elements are transmitted by a MUC followed by a TC. The MUC represents a run-length value of 64 x N (where N is an integer between 1 and 27) which is equal to, or shorter than, the value of the run to be transmitted. The following TC specifies the difference between the MUC and the actual value of the run to be transmitted.

The coding of each scan-line continues until all runs on the line (i.e. a total of 1728 pels) have been transmitted. Each coded line is followed by the EOL codeword, which is a unique sequence of bits consisting of eleven '0s' followed by a '1'. Thus, even if a transmission error corrupts some of the coded scan-line data, the error cannot prevent the EOL from being detected and the error is therefore confined within a scan-line.

If the number of coded bits in a scan-line is fewer than a certain agreed minimum, then FILL bits consisting of varying length sequences of '0s' are inserted between the line of coded data

Table 3.1
Modified Huffman Code Table

- Terminating Codewords -

Run-length	white runs	black runs
0	00110101	0000110111
1	000111	010
2	0111	11
3	1000	10
4	1011	011
5	1100	0011
6	1110	0010
7	1111	00011
8	10011	000101
9	10100	000100
10	00111	0000100
11	01000	0000101
12	001000	0000111
13	000011	00000100
14	110100	00000111
15	110101	000011000
16	101010	0000010111
17	101011	0000011000
18	0100111	0000001000
19	0001100	00001100111
20	0001000	00001101000
21	0010111	00001101100
22	0000011	00000110111
23	0000100	00000101000
24	0101000	00000010111
25	0101011	00000011000
26	0010011	000011001010
27	0100100	000011001011
28	0011000	000011001100
29	00000010	000011001101
30	00000011	000001101000
31	00011010	000001101001
32	00011011	000001101010
33	00010010	000001101011
34	00010011	000011010010
35	00010100	000011010011
36	00010101	000011010100
37	00010110	000011010101
38	00010111	000011010110
39	00101000	000011010111
40	00101001	000001101100
41	00101010	000001101101
42	00101011	000011011010
43	00101100	000011011011
44	00101101	000001010100
45	00000100	000001010101
46	00000101	000001010110
47	00001010	000001010111
48	00001011	000001100100
49	01010010	000001100101
50	01010011	000001010010
51	01010100	000001010011
52	01010101	000000100100
53	00100100	000000110111
54	00100101	000000111000
55	01011000	000000100111
56	01011001	000000101000
57	01011010	000001011000
58	01011011	000001011001
59	01001010	000000101011
60	01001011	000000101100
61	00110010	000001011010
62	00110011	000001100110
63	00110100	000001100111

- Make-up Codewords -

Run-length	white runs	black runs
64	11011	0000001111
128	10010	000011001000
192	010111	000011001001
256	0110111	000001011011
320	00110110	000000110011
384	00110111	000000110100
448	01100100	000000110101
512	01100101	0000001101100
576	01101000	0000001101101
640	01100111	0000001001010
704	011001100	0000001001011
768	011001101	0000001001100
832	011010010	0000001001101
896	011010011	0000001110010
960	011010100	0000001110011
1024	011010101	0000001110100
1088	011010110	0000001110101
1152	011010111	0000001110110
1216	011011000	0000001110111
1280	011011001	0000001010010
1344	011011010	0000001010011
1408	011011011	0000001010100
1472	010011000	0000001010101
1536	010011001	0000001011010
1600	010011010	0000001011011
1664	011000	0000001100100
1728	010011011	0000001100101
EOL	000000000001	000000000001

Table 3.2
Extended Modified Huffman Code Table

Run Length (Black or White)	Make-up Codeword
1792	00000001000
1856	00000001100
1920	00000001101
1984	000000010010
2048	000000010011
2112	000000010100
2176	000000010101
2240	000000010110
2304	000000010111
2368	000000011100
2432	000000011101
2496	000000011110
2560	000000011111

From ref. (43)

and the EOL codeword.

To cater for the transmission of an optional larger document up to A3 (about 2600 pels/line) in size using Group 3 machines, an extended code table, as shown in Table 3.2, was constructed by adding 13 extra MUC to the basic code table of Table 3.1. Coding a scan-line of 2600 pels is carried out in the similar manner to that explained above. The use of the extended code table is signalled as per the Recommendation T.30 control procedures.

3.3.2 Construction of the Modified Huffman Code Tables

The properties of the EOL codeword can be further understood by considering the construction of the Modified Huffman code tables. Each code table was initially designed according to Huffman's procedure and to contain the codeword '0000000' (seven '0s') which was designated to signal the end of a scan-line. Redundant bits were then added to the codeword of seven '0s' to form a codeword of ten '0s' followed by a '1'. From Table 3.1, it can be seen that no codeword ends in a sequence of more than three '0s', or begins with a sequence of '0s' larger than six, therefore the EOL of ten '0s' followed by a '1', forms a unique sequence which cannot be produced by concatenation of codewords. The final '1' of the EOL is included to indicate the start of the next coded line, since FILL bits may extend the sequence beyond ten '0s'.

The extended black and white code tables were formed using seven '0s' as the prefix for the 13 extra MUC. The seven '0s' codeword, originally designated to signal the end-of-line, now needs to be increased to eight 'Os'. Redundant bits are then added to this codeword to form the EOL consisting of eleven 'Os' followed by a '1'. This EOL codeword is unique for both the basic and extended code tables. This process can be carried out without altering any of the other codewords of Table 3.1. The same 13 extra codewords can be added to each code table without loss in efficiency since long runs occur very infrequently.

The performance of the Modified Huffman code, pertaining to its compression efficiency and error sensitivity, is well-documented in Ref. (43) and Ref. (106) and has also been briefly mentioned in the review chapter (section 2.3.2 (A)), in which the code compares favourably with the best one-dimensional coding scheme. Further description of its performance will not be made in this chapter.

3.4. TWO-DIMENSIONAL CODING STANDARD

One of the main advantages of the two-dimensional coding schemes is that they provide greater redundancy reduction compared to their one-dimensional counterparts by exploiting the line-to-line correlation that exists in most facsimile images. However, the two-dimensional schemes are more vulnerable to transmission errors which now tend to propagate down the page as well as horizontally. Error propagation can, however, be limited to only a few lines and it is felt that the degradation is not large enough, in general, to prevent their use.

In December 1978, Japan(57) contributed a two-dimensional coding scheme called the Relative Element Address Designate (READ) code to the CCITT SG. XIV. This scheme combines the features of two earlier coding techniques, one called Relative Address Coding (RAC)(54) and the other Edge Difference Códing (EDIC)(56) to improve overall coding performance. Japan proposed that the READ code should be considered for inclusion as an additional option in the T.4 Recommendation for Group 3 equipment, and that the code should be incorporated as an extension to the Group 3 one-dimensional coding scheme.

After the Japanese proposal, CCITT SG. XIV subsequently received further contributions regarding two-dimensional coding schemes from IBM Europe(107), the 3M Company(108), AT&T(109), the British Post Office, now British Telecom(110) (all of whose codes are direct extensions of the basic one-dimensional code), the Federal Republic of Germany(111) and the Xerox Corporation(112) (schemes using predictive coding). All these codes were extensively tested by CCITT SG. XIV delegates in terms of their coding efficiency and error susceptibility. Their relative performances were then assessed in Kyoto, Japan, in December 1979. Comparatively, there was little difference between these proposals, but the READ code was strongly recommended purely because it has been realised in a large number of commercial machines. However, modifications to the READ coding procedure were suggested to simplify its implementation without seriously changing its compression efficiency. The modifications to the original READ code, proposed by Japan, were as follows:-

(i) Vertical Mode coding (explained later) should be restricted so that the examination of the reference line (i.e. the preceding line) does not extend beyond ± 3 pels. The statistics for the coding elements obtained for the READ code show that the Horizontal Mode coding (also explained later) was nearly always more efficient than Vertical Mode coding when the examination of reference line is extended beyond 3 pels. This restriction simplifies implementation since it is not necessary to code every changing element (See next section) by both Horizontal and Vertical Mode coding.

(ii) The necessity of inserting bits (bit stuffing) into the coded data should be avoided. It was generally agreed that the use of insertion bits in the READ code to ensure a unique line synchronisation sequence would add extra cost and implementation complexity.

(iii) The EOL codeword should be made the same as that used in the one-dimensional coding procedure. This ensures that the code retains its synchronisation properties and avoids the need for bit stuffing.

(iv) The code should cater for future extensions, in particular provision for an uncompressed mode. Later it may also be desirable to include more sophisticated coding procedures, such as feature extraction or pattern recognition techniques, or the coding of gray or coloured areas.

A suitable compromise was finally reached. Called the Modified READ code and proposed by the Japanese delegation, it was readily supported by CCITT SG. XIV.

3.4.1 The Two-Dimensional Coding Scheme

The Modified READ code is a line-by-line sequential coding scheme in which the position of each changing element on the scan-line is coded with respect either to the position of a corresponding changing element on the reference line, which lies immediately above the line being coded, or the preceding changing element on the line. One-dimensional coding using the Modified Huffman code is applied to the first line of every successive K lines, where K equals two at normal resolution and K equals four at high resolution, in order to localise the disturbed area in the event of transmission errors. Prior to describing the coding procedure, some preliminary definitions regarding the changing pels and the modes of coding are necessary.

(A) Definition of Changing Picture Elements

Definition: A changing picture element is an element whose colour (black or white) is different from that of the previous element on the same line. The coding algorithm makes use of 5 changing elements situated on the coding and reference lines. These are defined below for the example given in Fig. 3.3.

a_0 : The reference or starting changing element on the coding line. At the start of the coding line, a_0 is set to an imaginary white changing element situated just before the first actual element on the coding line.

a_1 : The next changing element to the right of a_0 on the coding line. This has the opposite colour to a_0 and is the next changing element to be coded

a_2 : The next changing element to the right of a_1 on the coding line.

b_1 : The next changing element on the reference line to the right of a_0 and having the same colour as a_1.

b_2 : The next changing element on the reference line to the right of b_1.

If any of the coding elements a_1, a_2, b_1, b_2 are not detected at any time during the coding of the line, then they are set on an imaginary element positioned just after the last actual element on the respective scan-line.

(B) Definition of Coding Modes

Three coding modes are defined and are selectively used according to the coding procedure described subsequently.

(i) Pass Mode : As shown in Fig. 3.3(a), the state where the changing pels b_1 and b_2 on the reference line are detected between the starting pel a_0 and the changing

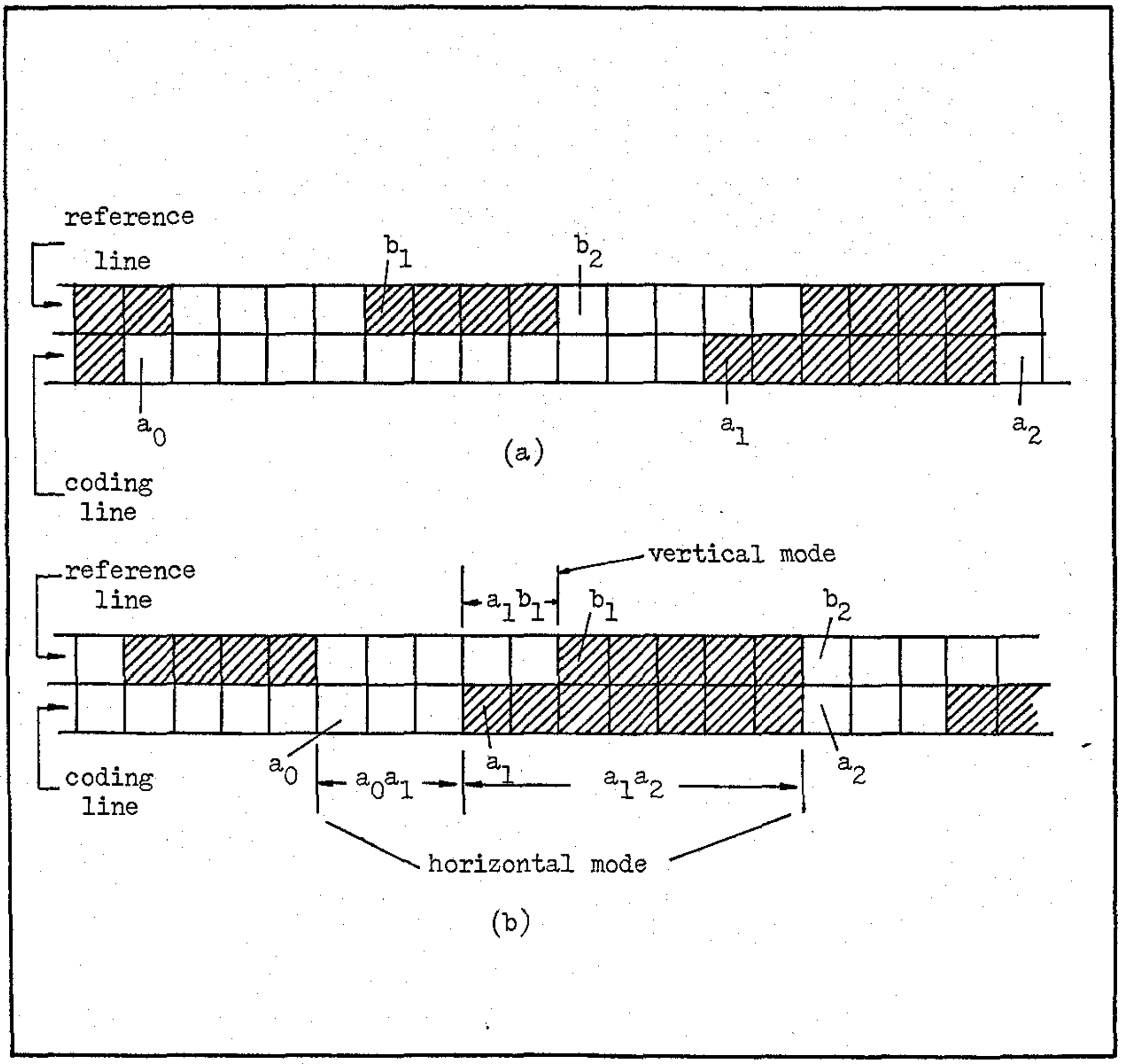


Fig. 3.3 - (a) Pass Mode (b) Horizontal and Vertical Mode

pel a_1 is defined to be the Pass Mode. The state where b_2 occurs at the same point horizontally as a_1 is not regarded as a Pass Mode. The purpose of the Pass Mode is to identify white or black runs on the reference line which are not adjacent to corresponding white or black runs on the coding line. The Pass Mode is represented by a single codeword as shown in Table 3.3.

(ii) Vertical Mode : As shown in Fig. 3.3(b), the coding of the distance a_1b_1, where the position of a_1 is coded relative to the position of b_1, is defined as the Vertical Mode. The relative distance a_1b_1 can take on one of seven values $V(0)$, $V_R(1)$, $V_R(2)$, $V_R(3)$, $V_L(1)$, $V_L(2)$, $V_L(3)$, each of which is represented by a separate codeword. The subscripts R and L indicate that a_1 is to the right or left of b_1 respectively, and the number in brackets indicates the value of the distance a_1b_1.

(iii) Horizontal Mode : The coding of distance a_0a_1 and the distance a_1a_2 on the coding line is defined as the Horizontal Mode. The distances are coded using the codeword $H + M(a_0a_1) + M(a_1a_2)$. H is a flag codeword '0001' (See Table 3.3) and $M(a_0a_1)$ and $M(a_1a_2)$ are Modified Huffman codes representing the colours and values of the run-lengths a_0a_1 and a_1a_2.

(C) The Coding Procedure

The coding procedure can be best explained with the aid of the flow-chart of Fig. 3.4. Having determined the next changing elements

MODE	ELEMENTS TO BE CODED		NOTATION	CODEWORD
PASS	b_1, b_2		P	0001
HORIZONTAL	a_0a_1, a_1a_2		H	001+M(a_0a_1)+M(a_1a_2)
VERTICAL	a_1 JUST UNDER b_1	a_1b_1=0	V(0)	1
	a_1 TO THE RIGHT OF b_1	a_1b_1=1	$V_R(1)$	011
		a_1b_1=2	$V_R(2)$	000011
		a_1b_1=3	$V_R(3)$	0000011
	a_1 TO THE LEFT OF b_1	a_1b_1=1	$V_L(1)$	010
		a_1b_1=2	$V_L(2)$	000010
		a_1b_1=3	$V_L(3)$	0000010
2-D EXTENSIONS 1-D EXTENSIONS				0000001XXX 000000001XXX
END-OF-LINE CODEWORD (EOL)				000000000001
1-D CODING OF NEXT LINE 2-D CODING OF NEXT LINE				EOL + '1' EOL + '0'

M(a_0a_1) and M(a_1a_2) are codewords taken from the Modified Huffman code tables given in Table 3.1 and Table 3.2. The bit assignment for the XXX bits is 111 for the uncompressed mode.

Table 3.3 - The two-dimensional code table

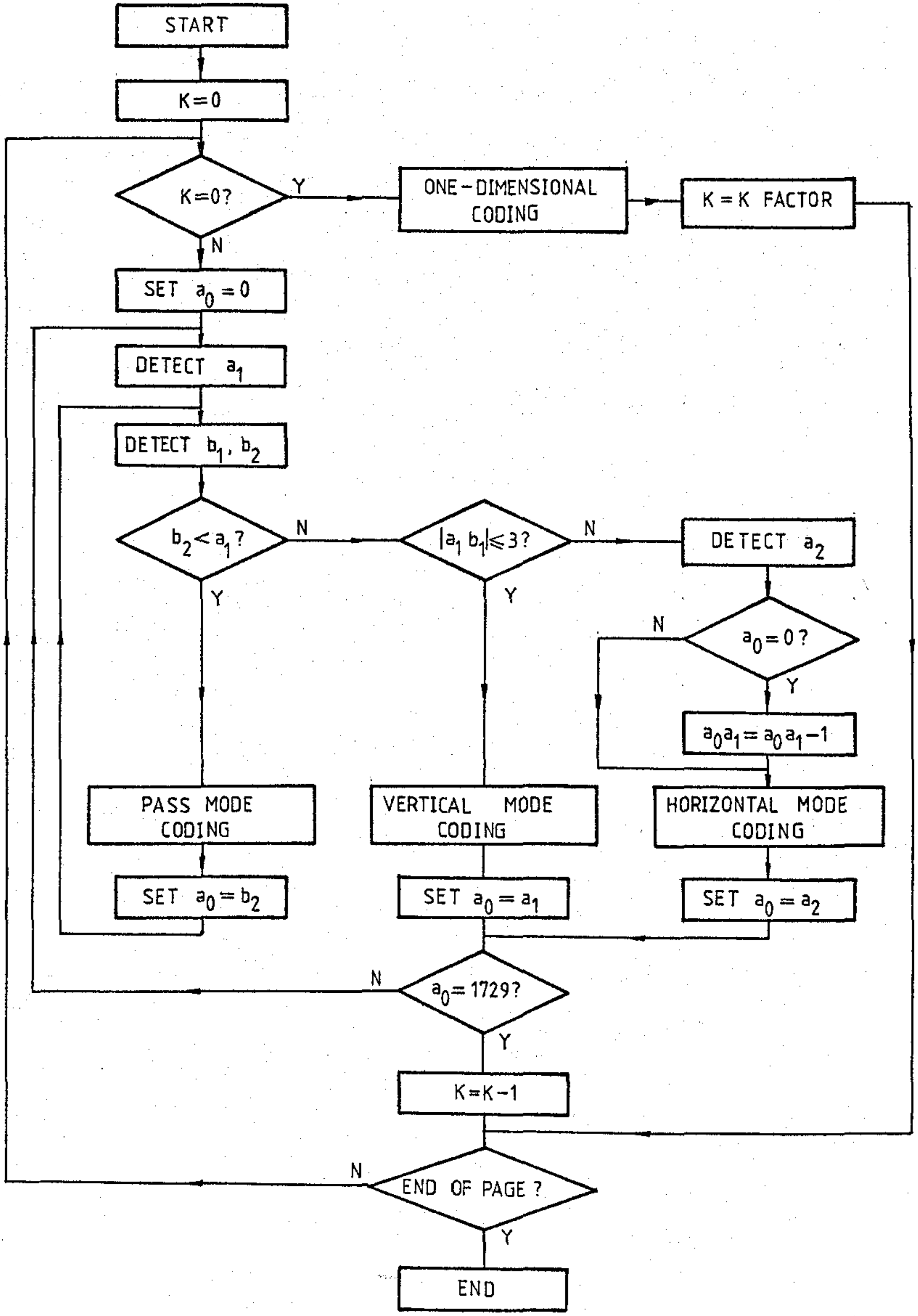


Fig. 3.4 - Flow-Chart for two-dimensional code

a_1, a_2, b_1, b_2, the coding procedure identifies the next coding mode, selects the appropriate codeword and resets the reference element a_0. The procedure is as follows:

(i) If b_2 is detected before a_1, then a Pass Mode has been detected and the codeword '0001' is sent. The reference element a_0 is set according to the element below b_2 in preparation for the next coding.

(ii) If the Pass Mode is not detected, the number of elements which separate a_1 and b_1 is determined. If $\left| a_1b_1 \right| \leqslant 3$, then the relative distance a_1b_1 is coded by Vertical Mode coding. Element a_0 is set according to the position of a_1 to be the new reference element. If $\left| a_1b_1 \right| > 3$, then the positions of a_1 and a_2 are coded using Horizontal Mode coding. The codeword $H + M(a_0a_1) + M(a_1a_2)$ is therefore transmitted, and a_2 is then regarded as the new position of the reference element a_0.

(D) Coding the First and Last Elements on a Line

The first starting pel a_0 on each coding line is imagined to be set at a position just before the first picture element, and is regarded as a white picture element. In this case, if the Horizontal Mode coding is used to code the first element on the coding line, then the value a_0a_1 is replaced by $a_0a_1 - 1$, to ensure that the correct run-length value is transmitted. Therefore, if the first element on a line is black, then the first codeword $M(a_0a_1)$ will be that which represents a white run of zero length.

Coding of each scan-line is ended after coding the changing pel a_1 or a_2 imagined to be positioned next to the last pel. If b_1 or b_2 is not detected on the reference line, an imaginary pel just after the last pel is assumed to be present.

(E) EOL Codeword, Tag Bits, FILL Bits, and Return to Control

The EOL codeword is uniquely constructed consisting of eleven '0s' and a '1'. Each EOL codeword is followed by a single tag bit, a '1' or a '0', which indicates that the next line is one- or two-dimensionally coded, respectively. FILL bits consisting of variable length strings of '0s' are inserted, when required, at the end of a coded line and before the EOL and the tag bit. The return to control (RTC) signal consists of six consecutive EOL codewords, each of which is followed by a '1' tag bit.

(F) Uncompressed Mode

To prevent any cases of data expansion, in which the number of bits assigned to a coding line exceeds the number of pels, IBM(113) proposed the uncompressed mode as an option to the two-dimensional coding scheme. Entry to the uncompressed mode on a one- and two-dimensionally coded line is achieved by using the one and two-dimensional extension codewords, respectively, given in Table 3.3, with the bits XXX set to 111. The other combinations of XXX are reserved for other as yet unspecified extensions. In the uncompressed mode, a group of five successive '0s' must be followed by an insertion of bit '1' to allow controlled exit from the uncompressed mode. The uncompressed mode is still under review by SG. XIV and has not yet been fully tested (late 1981).

As stated earlier, in terms of compression efficiency, there was little difference in performance between the Modified READ code and the other proposed coding schemes. Compared with the one-dimensional Modified Huffman code, at high resolution, K equals infinity and MSLT equal 0 ms, the Modified READ code is about 45 percent more efficient than that of the former. This represents a substantial improvement in transmission times and it is due to this fact that the CCITT has decided to accept a two-dimensional coding scheme for consideration as an international standard. The performance of the Modified READ code is described extensively in Refs. (43), (106) and (51) and will not be discussed further in this chapter.

3.5. CONCLUSION

In this chapter, a brief description of the international facsimile coding standards has been given. The one-dimensional Modified Huffman code and the two-dimensional READ code are the basis of the CCITT standard (Recommendation T.4) which will allow the interworking of digital facsimile equipment both on the national and international PSTN. The performance of these codes is well-documented and no attempt has been made to describe them in full, though mentioned briefly, but rather to describe the coding procedure itself.

The establishment of the Group 3 standards has increased the degree of compatibility between facsimile machines (including Groups 1 and 2) and, it is to be hoped, will stimulate the growth of facsimile communications as well.

CHAPTER IV

ADAPTIVE PEL LOCATION CODING

This chapter describes a relatively simple method of facsimile coding called Adaptive Pel Location Coding (APLC). Operating in one and two dimensions, the scheme is a deviation from conventional run-length coding methods and uses a block coding technique similar to that of deCoulon and Kunt(35). Being an information lossy coding scheme, it preserves image intelligibility and achieves high compression ratios at the expense of slight image degradation. Comparison with other coding schemes is also made here.

4.1. INTRODUCTION

Facsimile document coding, as the name implies, involves the coding of documents which will enable the receiver to reproduce exact replicas. Generally however, some degradation is acceptable, the amount depending on the system's requirements to achieve higher compression. Practically, the presence of a small amount of distortion is inevitable in a real system, and the requirement that the receiver produce exact replicas is artificial. Thus, in selecting the encoding algorithm, there is a trade-off between accuracy of reproduction and the number of bits transmitted.

Described here is a scheme called Adaptive Pel Location Coding (APLC), that achieves significant compression of the number of pels required for transmission, and provides good but not perfect replication of the original document. APIC is an adaptive scheme in that the number of bits transmitted varies as a function of the local distribution of black and white pels in the document being coded. It also takes advantage of the fact that most documents are inherently predominantly white and exploits the two-dimensional correlation that exists in most facsimile images.

This simple method operates on a block of data containing N pels, where N is even, and determines if it is completely white or completely black. If the condition is satisfied, a two bit prefix code is assigned to the block to ensure proper decoding at the receiver. To achieve higher compression with only a slight loss of fidelity, some preprocessing is incorporated within the algorithm itself. If one odd pel were present in a block, be it white or black, then the odd pel is considered as random noise and thus coded as if it were a block containing wholly black pels or white pels respectively. If the above condition are not satisfied, the block is further subdivided and this information is transmitted as a one bit prefix code. The scheme operates in both one and two dimensions. It is made adaptive by initially determining the prefix bits as before for the block of N pels, then dividing the block equally and assigning the prefix bits to each sub-block. These sub-blocks are repeatedly divided until a basic picture block containing λ elements, typically 4, when pel location coding is introduced. For the basic picture block of 4 pels, if the number of white pels is equal to the number of black pels, then the location of the black pels is transmitted to the receiver together with a one bit prefix code. There are one of $\binom{\lambda}{\alpha}$ possibilities for coding the location of the black pels within the block of λ pels, where α is the number of black pels. For each possibility, there are $\log_2\binom{\lambda}{\alpha}$ bits per address location. An isolated pel within a block is considered as noise and the block is therefore coded as if it were completely homogeneous.

Section 4.2 describes one-dimensional APLC while two-dimensional APLC is explained in Section 4.3. The results, including subjective results, are summarised in Section 4.4 and finally, conclusions are drawn in Section 4.5.

4.2. ONE-DIMENSIONAL APLC

In one-dimensional APLC, blocks of pels are processed sequentially along the scan-lines. Consider an initial block of N pels $\{x_k\} = x_1,x_2,......x_N$, where each pel is either white or black, represented by a logical 0 or 1, respectively, and N is an even number. To reduce the number of bits from N, we form a new codeword whose prefix is:-

$$P_{1,1} = 10, \qquad \text{if} \quad \sum_{i=1}^{N} x_i = 0 \text{ or } 1 \qquad (4.1)$$

$$P_{1,1} = 11, \qquad \text{if} \quad \sum_{i=1}^{N} x_i = N \text{ or } N-1 \qquad (4.2)$$

Thus, if $P_{1,1} = 10$, $\{x_k\}$ consists of all white pels, or has one black pel. $P_{1,1} = 11$ means all black pels, or one white pel among the black pels. The N pels are therefore transmitted with only two bits, and one pel in N may be in error on decoding. If Eqns. (4.1) and (4.2) are not satisfied, then $P_{1,1} = 0$ after which the blocks of N pels is divided into two blocks of N/2 pels and the process is repeated where the prefix bits are:-

$$P_{2,1} = 10, \quad \text{if} \quad \sum_{i=1}^{N/2} x_i = 0 \text{ or } 1 \qquad (4.3)$$

$$P_{2,1} = 11, \quad \text{if} \quad \sum_{i=1}^{N/2} x_i = N/2 \text{ or } N/2-1 \qquad (4.4)$$

otherwise $P_{2,1} = 0$, and

$$P_{2,2} = 10, \quad \text{if} \quad \sum_{i=N/2+1}^{N} x_i = 0 \text{ or } 1 \qquad (4.5)$$

$$P_{2,2} = 11, \quad \text{if} \quad \sum_{i=N/2+1}^{N} x_i = N/2 \text{ or } N/2-1 \qquad (4.6)$$

otherwise $P_{2,2} = 0$.

The generalised equations are as follows:-

$$P_{m,j} = 10, \quad \text{if} \quad \sum_{i=(j-1)(N/2^{m-1})+1}^{jN/2^{m-1}} x_i = 0 \text{ or } 1 \qquad (4.7)$$

or

$$P_{m,j} = 11, \quad \text{if} \quad \sum_{i=(j-1)(N/2^{m-1})+1}^{jN/2^{m-1}} x_i = (N/2^{m-1}) \text{ or } (N/2^{m-1})-1 \qquad (4.8)$$

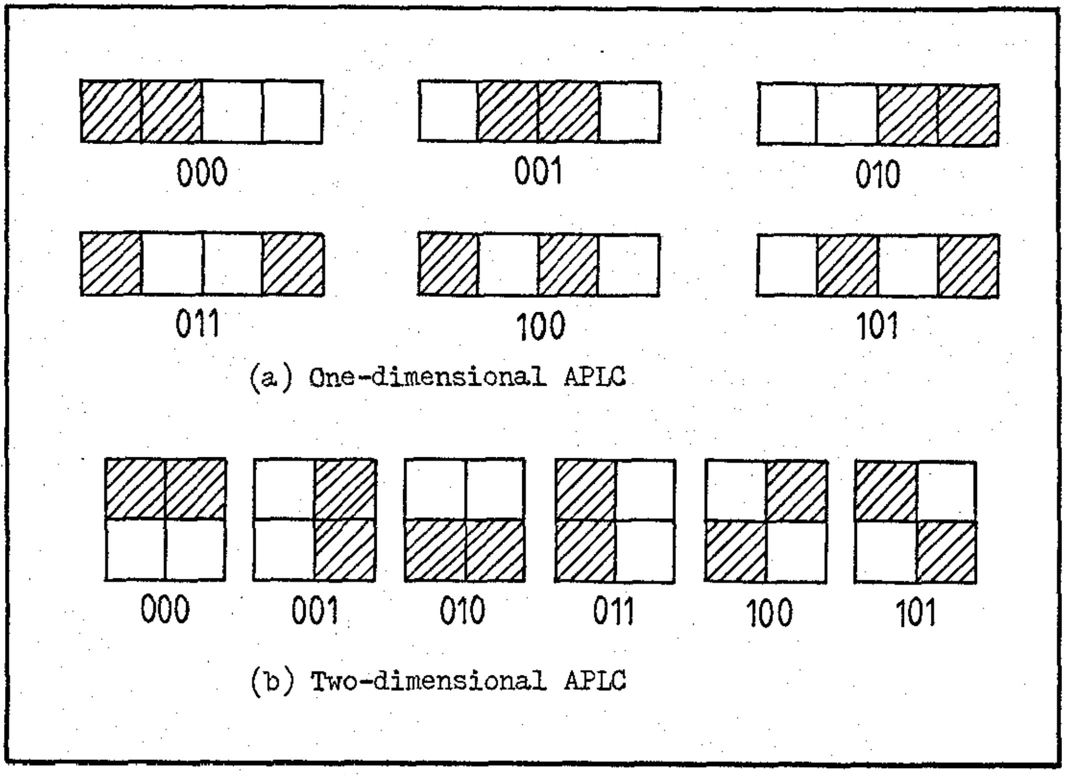
or

$$P_{m,j} = 0 \qquad (4.9)$$

The subscript m=1 refers to one block of size N; m=2 refers to two blocks of size N/2, and m=M refers to 2^{M-1} blocks of size $N/2^{M-1}$. Subscript j=1 identifies the block of size N, while subscripts $2 \leq j < 2^{m-1}$ apply to sub-blocks of size $\leq N/2$. When m=1, j=1, Eqns. (4.7) and (4.8) are identical to Eqns. (4.1) and (4.2). Observe that when m=2, there are two values of j (i.e. j=1 and j=2). For this situation, Eqns. (4.3), (4.4), (4.5) and (4.6) are identical to Eqns. (4.7)and (4.8). Suppose for m=2, $P_{2,1}$ and $P_{2,2}$ are both zero, then the blocks of N/2 are sub-divided, where m now equals 3 and j=1,2,3,4 ($j=1,2,....2^{m-1}$) corresponding to four sequential blocks each containing N/4 pels. If $P_{3,1}$, $P_{3,2}$, $P_{3,3}$, and $P_{3,4}$ are all zero, the process of sub-dividing the blocks by two continues until m=M, when the basic block size of λ pels is reached.

In the results to be presented, $\lambda = 4$ and there are $\binom{4}{2}$ possibilities which define the location of the black pels. The locations of the two black pels in the basic picture block are given by three bit codes as shown in Fig. 4.1(a). The decoder reverses the coding procedure, but decodes 10 and 11 as blocks of all whites and all blacks, respectively.

An example of one dimensional APLC is shown in Fig. 4.2 where N is taken to be 16. Comparison is also made with the auto-adaptive block coding (ABC) scheme of deCoulon and Johnsen(35). For APLC, the blocks with prefix beginning with a '1' require no further sub-division.



(a) One-dimensional APLC

(b) Two-dimensional APLC

Fig. 4.1 - Location of black pels in basic picture block

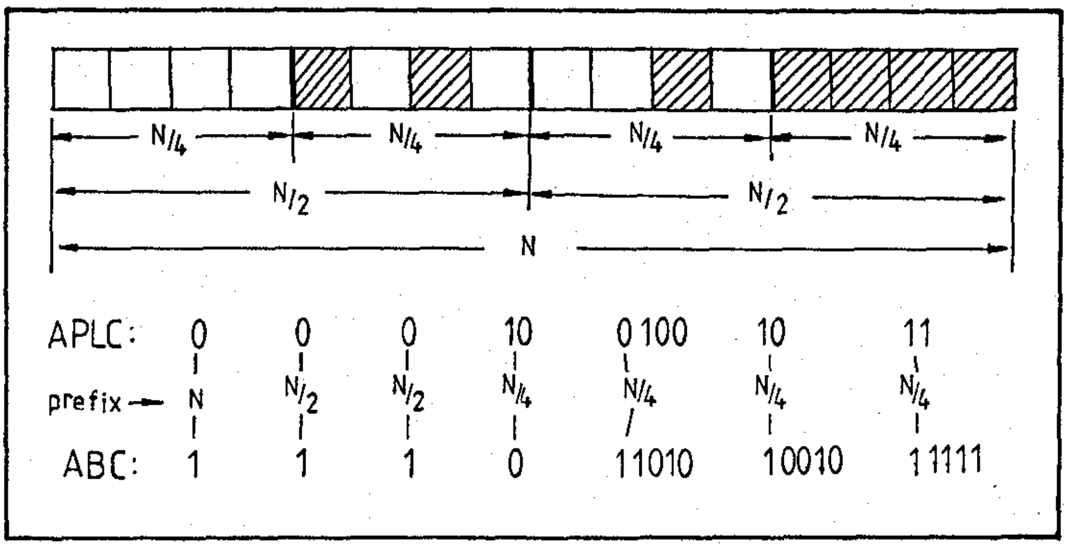

N/4
N/4
N/4
N/4
N/2
N/2
N
APLC: 0 0 0 10 0100 10 11
prefix → N N/2 N/2 N/4 N/4 N/4 N/4
ABC: 1 1 1 0 11010 10010 11111

Fig. 4.2 - One-dimensional APLC

4.3. TWO-DIMENSIONAL APLC

In the two-dimensional case, two dimensional correlation plays an important role in achieving higher compression ratios. The algorithm adapts to the local activity of the image and operates on the same basis of sub-dividing the blocks as the one-dimensional version. If a two dimensional block of N columns and N rows with N^2 pels is considered, then the prefix bit is:-

$$P_{1,1} = 10, \quad \text{if} \quad \sum_{i=1}^{N} \sum_{j=1}^{N} x_{i,j} = 0 \text{ or } 1 \qquad (4.10)$$

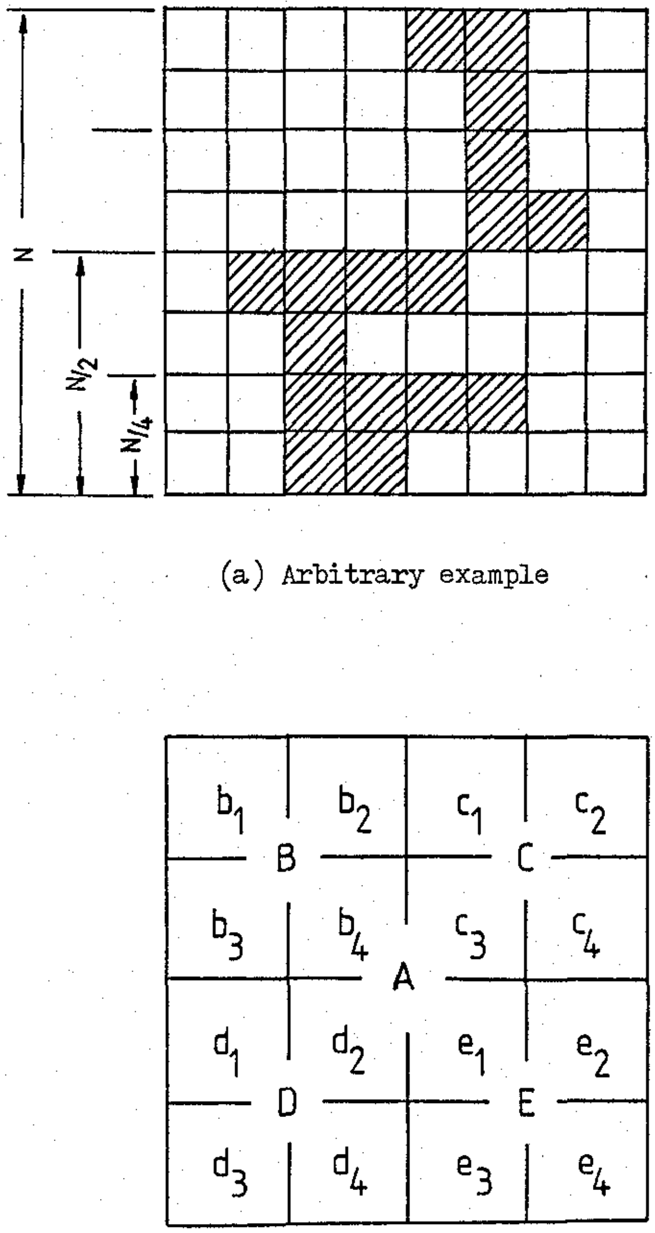
$$P_{1,1} = 11 \quad \text{if} \quad \sum_{i=1}^{N} \sum_{j=1}^{N} x_{i,j} = N^2 \text{ or } N^2-1 \qquad (4.11)$$

where $x_{i,j} = 0$ for white pels

$= 1$ for black pels.

Otherwise $P_{1,1} = 0$, and further sub-division is required.

Rather than by reproducing the detailed equations, the operation of two-dimensional APLC can be illustrated by the example shown in Fig. 4.3(a). For a large block A having N x N pels, the prefix bit (See Fig. 4.3(b)) is 0 as there are more than two black pels and two white pels. The block is sub-divided into four blocks, B, C, D, and E, each containing $N^2/4$ pels. B contains all white pels and is coded as 10. Since C does not satisfy the condition of the algorithm, it is represented by a prefix bit 0 and its sub-blocks c_1, c_2, c_3, and c_4 are coded as 11, 10, 0 001, 10,



(a) Arbitrary example

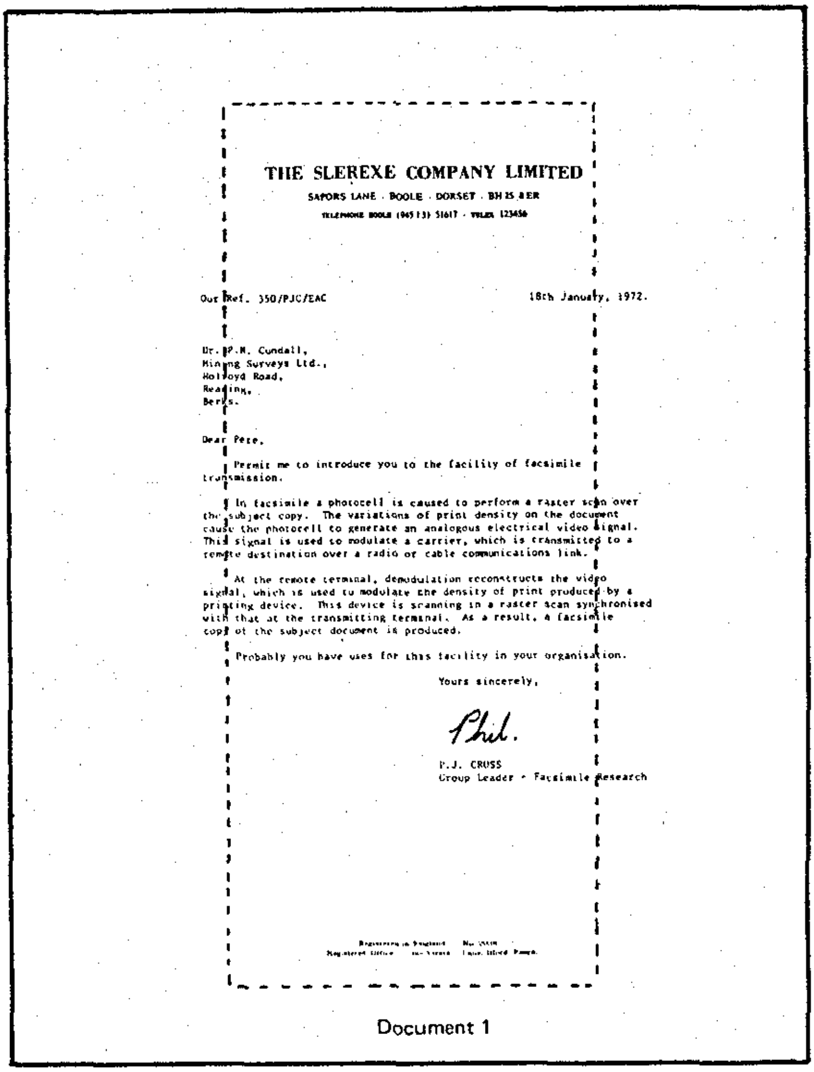
(b) Designation of blocks

Fig. 4.3 - Two-dimensional APLC

where c_3, having two black pels is found using the location code shown in Fig. 4.1(b). Notice that sub-blocks c_1 and c_4 contain one white and one black pel and are therefore coded with the prefix bit 11 and 10 respectively. The prefix bit for D is 0 and its sub-blocks are d_1, d_2, d_3, d_4, coded as 10,11,10,11. Lastly, the prefix bit for E is 0 and e_1, e_2, e_3, and e_4 are coded as 10, 10, 0 000 (see location code of Fig. 4.1(b)), and 10. As N=8, the original 64 pels have been compressed to 34 bits and because the receiver decodes 10 and 11 as meaning all white and all black pels respectively, 5 pels are decoded in error. In general, the error rate is P/4, where P($\simeq$ 0.3) is the probability of one pel being different from the other three in the basic picture block. It must be emphasised that Fig. 4.3(a) is an example to aid exposition and does not represent a typical facsimile image, for the probability of a typewritten letter being one element thick is very remote.

4.4. RESULTS

CCITT documents No. 1, 2, 4, and 5, digitised with 1728 pels per line, and 2376 lines, corresponding to a resolution of 8 pels/mm both horizontallly and vertically were used, although for convenience only 1024 pels of each line were processed (as shown in Fig. 4.4 for document No. 1). Compression ratios for ABLC were compared with those for the auto-adaptive block coding scheme(35) (ABC), run-length coding using the B_1 code(37) and the Modified Huffman code(42). The B_1 code is a class of code whose word length increases logarithmically with the run-length. The compression



THE SLEREXE COMPANY LIMITED

SAPORS LANE · BOOLE · DORSET · BH 25 8ER
TELEPHONE BOOLE (945 13) 51617 · TELEX 123456

Our Ref. 350/PJC/EAC 18th January, 1972.

Dr. P.N. Cundall,
Mining Surveys Ltd.,
Holroyd Road,
Reading,
Berks.

Dear Pete,

Permit me to introduce you to the facility of facsimile transmission.

In facsimile a photocell is caused to perform a raster scan over the subject copy. The variations of print density on the document cause the photocell to generate an analogous electrical video signal. This signal is used to modulate a carrier, which is transmitted to a remote destination over a radio or cable communications link.

At the remote terminal, demodulation reconstructs the video signal, which is used to modulate the density of print produced by a printing device. This device is scanning in a raster scan synchronised with that at the transmitting terminal. As a result, a facsimile copy of the subject document is produced.

Probably you have uses for this facility in your organisation.

Yours sincerely,

Phil.

P.J. CROSS
Group Leader - Facsimile Research

Registered in England No. [illegible]
Registered Office [illegible]

Document 1

Fig. 4.4 - Part of the document processed

ratio (CR) is defined as the total number of pels in the original image divided by the number of bits transmitted. The CR values, which do not take into consideration 'house-keeping' bits like EOL and channel coding bits, are shown in Table 4.1 and Table 4.2 for the one-dimensional and two-dimensional coding schemes respectively.

Looking at the results of Table 4.1 and Table 4.2, it can be seen that APLC is as efficient as the Modified Huffman code for the one-dimensional case and is superior by quite a significant amount for the two-dimensional case. Its superior performance is mainly due to the inherent two-dimensional correlation that exists in most facsimile documents. Although the Modified Huffman code is only slightly better in performance than one-dimensional APLC, in terms of implementation the latter method would prove to be easier. In the two-dimensional case, results obtained are for initial block size of 128 x 128. It is envisaged that the CR values will differ by a marginal amount if different block sizes are considered. APLC does cause some degradation to the document and for an initial block of size 1024 for the one-dimensional case, using document No. 1, it is observed that the CR of 9.29 for the ABC scheme increases to 10.52 if ABC is arranged to reject the isolated odd pel using the strategy adopted by APLC. This is inferior to 11.98 achieved by APLC. However, it is with the two-dimensional APLC system that large values of CR are achieved when compared to the other encoding methods that were examined.

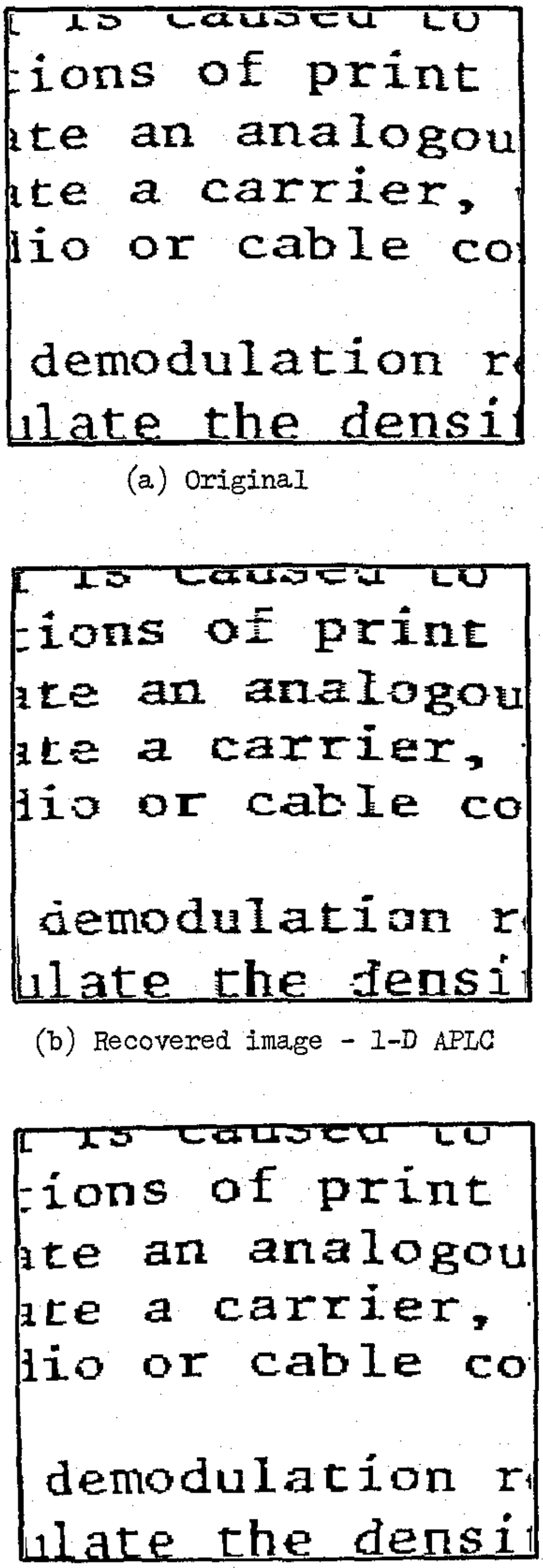
Fig 4.5 demonstrates the quality of the APLC system both in one

	DOCUMENT 1		DOCUMENT 2		DOCUMENT 4		DOCUMENT 5	
BLOCK SIZE	APLC	ABC	APLC	ABC	APLC	ABC	APLC	ABC
1 x 1024	11.98	9.287	13.82	9.264	3.992	2.984	7.696	5.856
1 x 512	11.94	8.712	13.97	8.832	4.149	2.813	7.725	5.544
1 x 256	11.83	8.653	14.04	8.856	4.154	2.816	7.780	5.574
1 x 128	11.53	8.491	13.95	8.825	4.157	2.817	7.807	5.588
1 x 64	10.91	8.149	13.34	8.610	4.151	2.814	7.711	5.539
M.H. Code	11.69		14.34		4.316		7.872	
B_1 Code	10.21		13.37		3.686		6.804	

Table 4.1 - C.R. values for one-dimensional coding

	DOCUMENT 1		DOCUMENT 2		DOCUMENT 4		DOCUMENT 5	
BLOCK SIZE	APLC	ABC	APLC	ABC	APLC	ABC	APLC	ABC
128 x 128	15.86	11.36	26.80	12.49	4.861	3.533	9.732	7.413

Table 4.2 - C.R. values for two-dimensional coding



is caused to
tions of print
ate an analogou
ate a carrier,
dio or cable co

demodulation r
ulate the densi

(a) Original

is caused to
tions of print
ate an analogou
ate a carrier,
dio or cable co

demodulation r
ulate the densi

(b) Recovered image - 1-D APLC

is caused to
tions of print
ate an analogou
ate a carrier,
dio or cable co

demodulation r
ulate the densi

(c) Recovered image - 2-D APLC

Fig. 4.5 - APLC Performance

and two dimensions. The picture size is 256 x 256 pels. Fig. 4.5(a) shows the original image. Notice that there are jagged edges present on the letters, probably caused by imperfections in the decision process of the scanner. These jagged edges not only degrade picture quality but also reduce coding efficiency. Fig. 4.5(b) shows the recovered image when one-dimensional APLC is applied. The initial block size used is 1 x 256 pels. The jagged edges are now more pronounced but the essential details are still preserved without any loss of intelligibility. Although the algorithm sometimes destroys the connectivity of the picture (especially in the one-dimensional case), the overall effect is not too objectionable to the eye. However, the slight degradation in picture quality can be traded-off against the increase in coding efficiency. When two-dimensional APLC is applied, the resulting image is as shown in Fig. 4.5(c). The initial block size used is 64 x 64. It not only improves the coding efficiency (since two-dimensional correlation is taken into account) but also, to a certain extent, the picture quality. All the jagged edges have been removed, resulting in an image which is well-defined but slightly 'blocklike' in structure. This, however, can be improved by employing some form of post-processing at the receiver.

4.5. CONCLUSION

In this chapter, the APLC algorithm has been described and it can be seen that its performance is superior to that of its counterparts especially in the two-dimensional case. It is an information lossy scheme which involves discarding isolated black or white pels to increase coding efficiency. Although some degradation can be observed, intelligibility is maintained. This is because the algorithm takes into consideration the fact that the probability of a typewritten letter one element thick vertically and horizontally is remote and exploits this property. In the two-dimensional case, two-dimensional correlation plays an important role in achieving even higher compression ratios. Finally, the adaptation of the algorithm to the local activity content of the image is a major factor contributing to its increase in coding efficiency.

4.6. NOTE ON PUBLICATION

The APLC scheme described in this chapter has appeared as a paper in IEE Electronics Letters, 9th May 1980, Vol. 16, No. 10. The paper is entitled "Adaptive Pel Location Coding for Bilevel Facsimile Signals" and was jointly authored by M. G. B. Ismail and R. Steele.

CHAPTER V

PRE - AND POST-PROCESSING

5.1. INTRODUCTION

Coding of two-tone facsimile images has gained considerable interest in recent years due to an increasing need for quick transmission of black and white documents. However, most of the results that have been reported have already shown a fairly high level of compression and any further substantial improvement in compression is thus likely to prove difficult. Preprocessing prior to actual encoding provides the solution by which the compression ratios of existing coding methods can be dramatically enhanced. The potential of this information lossy approach was quickly realised and in recent years numerous preprocessing techniques have emerged. Amongst the more important techniques are signal modification(26), a thinning process(27-28), and the notchless bilevel quantiser with logical feedback(23). The techniques of Netravali et. al.(18), Margner and Zamperoni(19), and Billings(24) are also worth mentioning. The coding scheme that was described in Chapter IV is another information lossy approach where psychovisual coding is used. In this approach, preprocessing is incorporated in the coding algorithm itself, where an isolated black or white pel within a basic picture block is discarded in order to increase coding efficiency. Another preprocessing technique, mainly aimed at cleaning up noisy originals, to ensure that the compression ratios remain high, has been extensively studied by Ting and Prasada[15-16].

The underlying notion which leads to the use of preprocessing prior to coding is not difficult to appreciate. Facsimile images sampled at sufficiently high density have an inherent redundancy and, even if the signal is locally changed from white to black or vice-versa, the amount of degradation which might cause an objectionable image is minimal. It therefore poses few problems for practical use. The inherent redundancy is mainly generated during the thresholding of a digital scanning process. Most modern facsimile machines employ solid state scanners, charged-coupleddevices (CCDs) for example, which quantise the image into two discrete levels. A bilevel quantisation of the signal ensures the representation of a document at a rate of 1 bit/pel prior to digital coding. The usual method for obtaining a bilevel representation is to use a certain threshold[67] and to classify the signal as black (level 1) when it is greater than or equal to the threshold and white (level 0) when it is less than the threshold. It performs fairly well for some documents but problems arise when there are variations in paper illumination or reflectance across the page. This results in a wrong decision by the scanner, which often introduces notches along the horizontal and vertical borders of the image detail. These notches not only degrade the image quality but also decrease coding efficiency since horizontal notches divide continuous horizontal runs into several smaller runs, and vertical notches reduce line-to-line correlation. Sometimes facsimile machines are called upon to handle noisy originals such as those produced by an electrostatic copier or those produced by a poor quality typewriter, which are characterised, respectively, by the presence of impulse noise or by tiny perforations visually observed on the characters. For these forms of input the efficiency of the coder generally decreases and the quality of the output copy is also affected. It is only through preprocessing that the effects can be minimised and coding efficiency considerably improved.

In this chapter, two preprocessing schemes, resulting in more efficient coding, are described. In Section 5.2, a simple preprocessing strategy using a set of 3 x 3 masks is presented. The value of the central picture element is altered from black to white or vice-versa, if the surrounding elements correspond to a predetermined pattern. A comparison is made with the Majority Logic Smoothing with Contour Preservation technique of Ting and Prasada[15]. A simple post-processing technique is also described.

In Section 5.3, a more involved method of preprocessing using subsampling is studied. Notches and pinholes in the original image are first removed using a set of masks, as described in Section 5.2, after which single element runs are doubled to preserve connectivity. The definition of connectivity will be explained later in Section 5.2.2. The image is then subsampled with different techniques being examined, including subsampling by the use of the Hadamard and Cosine Transforms. Enlargement of the subsampled image is carried out at the receiver where replication, bilinear and Cubic B-spline interpolation techniques are investigated. The edges of the interpolated images are then 'smoothed out' by a set of restoration masks, producing a visually acceptable image. A formal subjective experiment is also conducted to determine the order of subjective preference of the resulting restored images.

For both preprocessing schemes to be described, subjective results are presented. These schemes are prerequisites for two different coding strategies, which will be described in Chapters VI and VII.

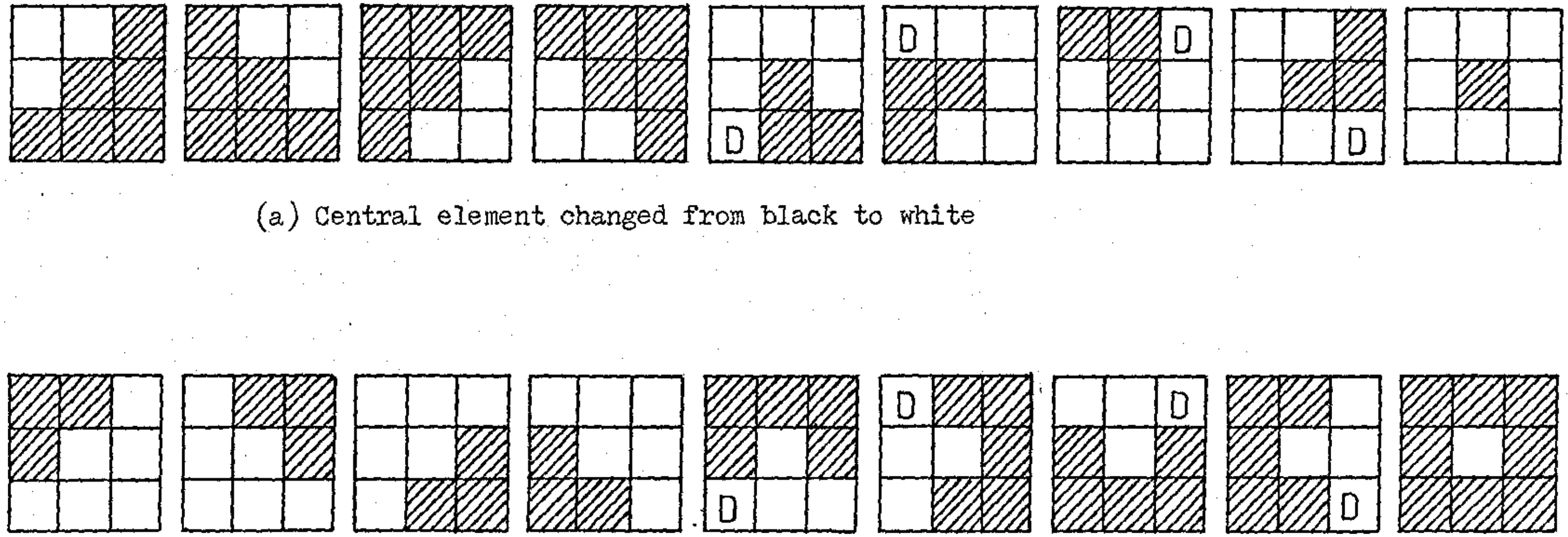
5.2. PRE- AND POST-PROCESSING USING MASKS

5.2.1 Preprocessing

Any preprocessing of a facsimile image prior to coding must satisfy the following requirements:-

(i) It must preserve the order of connectivity of the image.
(ii) It must not severely degrade the quality of the image.
(iii) It must improve the coding efficiency.

Based primarily on the thinning algorithm of Arcelli et. al.[87], the new preprocessing technique uses a set of masks whereby the central picture element is changed from black to white, or vice-versa, provided the surrounding elements correspond to a certain predetermined pattern. The set of masks used, shown in Fig. 5.1, satisfies the three requirements above. Each mask is tested against the source image within a 3 x 3 window and if the predetermined pattern is satisfied, the value of the central element is changed. For those elements shown in Fig. 5.1(a), the central element is changed from black to white, for those in Fig. 5.1(b),



(a) Central element changed from black to white

(b) Central element changed from white to black

'D' = Don't care situation

Fig. 5.1 - Masks employed in proposed preprocessing

from white to black. D is considered as a 'don't care' element where either one of the two values is acceptable. Figs. 5.1(a) and 5.1(b) are complementary and since there are only two levels to consider, implementation using logic gates is straightforward. The first four masks of Figs. 5.1(a) and 5.1(b) remove any rounded edges of the document text and at the same time preserve the contour direction. The next four masks remove unwanted notches in both horizontal and vertical directions. The last mask removes random noise either generated by the scanner or that already inherent in the original image.

5.2.2 Theoretical Analysis

The analysis begins with some definitions of connectivity[88]. Let (i,j) be a point of a given picture. Then (i,j) has four horizontal and vertical neighbours, namely the points,

(i-1,j), (i,j-1), (i,j+1), (i+1,j).

These points are called the 4-neighbours of (i,j) and are said to be 4-adjacent to (i,j). In addition, (i,j) has four diagonal neighbours namely,

(i-1,j-1), (i-1,j+1), (i+1,j-1), (i+1,j+1).

Both these and the 4-neighbours are called 8-neighbours of (i,j). A path from (i,j) to (h,k) is a sequence of distinct points joining (i,j) to (h,k). For example, if we denote the points of the path by 1's, then $\begin{matrix}1&1&1\\&&1\end{matrix}$ is a 4-path while $\begin{matrix}&1&1\\1&&\end{matrix}$ is an 8-path. If p=(i,j) and q=(h,k) are points of a picture subset S, p is said to be connected to q (in S) if there is a path from p to q consisting entirely of points in S. The concept of

4-connectivity and 8-connectivity are illustrated in Figs. 5.2(a) and 5.2(b), respectively.

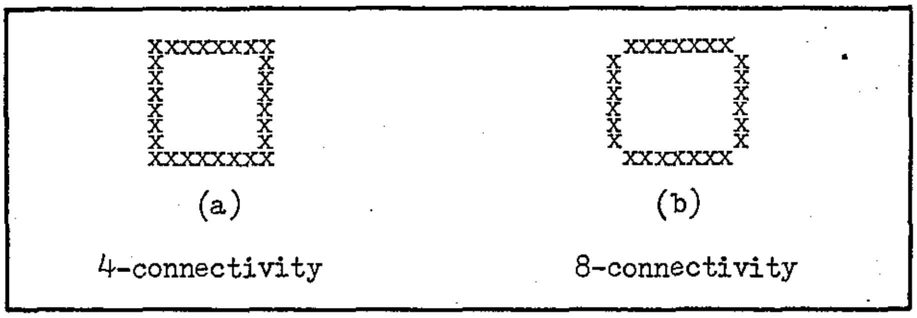


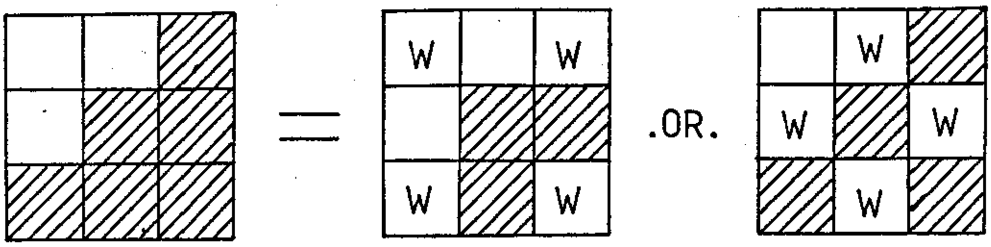
Fig. 5.2 - Figures showing the concept of connectivity

Consider a 3 x 3 mask as shown in Fig. 5.3 where x_0, x_1, x_2, ..., x_8 have digital values of either 1 or 0.

x_4	x_3	x_2
x_5	x_0	x_1
x_6	x_7	x_8

Fig. 5.3 - a 3 x 3 mask

Now consider the first mask of Fig. 5.1(a) to be represented as a logical .OR. of two sub-masks as shown in Fig. 5.4. Sub-mask (a) is characterised by having the property of 4-connectivity while sub-mask (b), has the property of 8-connectivity. Pels labelled W are considered as permanently white in both sub-masks to preserve their connectivity properties.



sub-mask (a)

sub-mask (b)

Fig. 5.4 - Logical .OR. representation of masks

Referring to Fig. 5.3 and Fig. 5.4, when $x_0 = 1$, the connected number (N_c) of x_0 for sub-mask (a) is defined as

$$N_c^{<4>} = \sum_{k \in S_1} (x_k - x_k \, x_{k+1} \, x_{k+2}) \tag{5.1}$$

and the connected number of x_0 for sub-mask (b), as

$$N_c^{<8>} = \sum_{k \in S_1} (\bar{x}_k - \bar{x}_k \, \bar{x}_{k+1} \, \bar{x}_{k+2}) \tag{5.2}$$

where $S_1 = \{1, 3, 5, 7\}$

The subscript of x is a modulo-8 sum. If the subscript of x is equal to or greater than 9, then 8 is subtracted from the value to give values less than 9. $\bar{x}$ means $(1-x)$ and the right shoulder superscripts $<4>$ and $<8>$ mean 4-connectivity and 8-connectivity as characterised by sub-mask (a) and sub-mask (b) respectively. It must be emphasised that Eqns. (5.1) and (5.2) strictly apply only for sub-mask (a) and sub-mask (b) respectively in which the pels marked W are regarded as permanently white.

Computing the values of $N_c^{<4>}$ and $N_c^{<8>}$ determines the topological properties of the central element x_0. Table 5.1 summarises these properties.

The values of $N_c^{<4>}$ or $N_c^{<8>}$	0	1	2	3	4
Property of x_0	internal or isolate	end	connect	branch	cross

Table 5.1 - Topological properties of x_0

For sub-mask (a) of Fig. 5.4, $N_c^{<4>} = 2$, while for sub-mask (b), $N_c^{<8>} = 3$, representing the topological properties of 'connect' and 'branch' respectively. With W's considered as pels whose colours remains white at all times, only the state of elements x_1, x_3, x_5 and x_7 for sub-mask (a) and elements x_2, x_4, x_6 and x_8 for sub-mask (b) can be changed. Table 5.2 gives all the possible topological properties of x_0 for both sub-masks where the states of the elements mentioned above are altered.

However, the connected numbers $N_c^{<4>}$ and $N_c^{<8>}$ for sub-masks (a) and (b) do not uniquely define the set of masks of Fig. 5.1 used in the preprocessing scheme. This difficulty can be overcome by weighting the state of elements x_1, x_3, x_5 and x_7 (for sub-mask (a)) and x_2, x_4, x_6 and x_8 (for sub-mask (b)). The weighting process considers elements x_1, x_3, x_5 and x_7 for sub-mask (b) and x_2, x_4, x_6 and x_8 for sub-mask (b) to be adjacent to one another. In other words, if sub-mask (a) is considered, x_1 is adjacent to x_3, x_3 is adjacent to x_5, ..., and x_7 is adjacent to x_1.

x_1	x_3	x_5	x_7	$N_c^{<4>}$	$N_w^{<4>}$	x_2	x_4	x_6	x_8	$N_c^{<8>}$	$N_w^{<8>}$
0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	1	1	1	0	0	0	1	1
0	1	0	0	1	1	0	1	0	0	1	1
1	1	0	0	2	3	1	1	0	0	2	3
0	0	1	0	1	1	0	0	1	0	1	1
1	0	1	0	2	2	1	0	1	0	2	2
0	1	1	0	2	3	0	1	1	0	2	3
1	1	1	0	3	6	1	1	1	0	3	6
0	0	0	1	1	1	0	0	0	1	1	1
1	0	0	1	2	3	1	0	0	1	2	3
0	1	0	1	2	2	0	1	0	1	2	2
1	1	0	1	3	6	1	1	0	1	3	6
0	0	1	1	2	3	0	0	1	1	2	3
1	0	1	1	3	6	1	0	1	1	3	6
0	1	1	1	3	6	0	1	1	1	3	6
1	1	1	1	4	10	1	1	1	1	4	10

Table 5.2 - All possible topological properties of x_0 for sub-mask(a) and sub-mask(b)

(Adjacency in this context is similar to the 'wrapround' concept in the Karnaugh map). Looking at Table 5.2, if a '1' is preceded by a '0', then the first '1' is weighted with a value 1, If a '1' is preceded by a '1', then the former is weighted with a value 2 and if a '1' is preceded by two adjacent '1's', then it is weighted with a value 3 and so on. This is best illustrated with an example:-

x_1	x_3	x_5	x_7	$N_c^{<4>}$	$N_w^{<4>}$
1	0	1	1	3	
3		1	2		6

weights

Notice that x_7 is taken to be adjacent to x_1. $N_w^{<4>}$ and $N_w^{<8>}$ are the weighted values of $N_c^{<4>}$ and $N_c^{<8>}$ respectively and the weighted topological properties of central element x_0 are shown in Table 5.2.

The weighted connected numbers $N_w^{<4>}$ and $N_w^{<8>}$ for sub-mask (a) and sub-mask (b) now uniquely define the set of masks of Fig. 5.1. Therefore, for the set of masks of Fig. 5.1(a), if the following unique conditions are satisfied, the central element is changed from black to white:-

(a) $N_w^{<4>} = 0$ and $N_w^{<8>} = 0$ —— isolated pel

(b) $N_w^{<4>} = 3$ and $N_w^{<8>} = 6$ —— 1st 4 masks

(c) $N_w^{<4>} = 1$ and $N_w^{<8>} = 1$ —— 2nd 4 masks with D = white

(d) $N_w^{<4>} = 1$ and $N_w^{<8>} = 3$ —— 2nd 4 masks with D = black

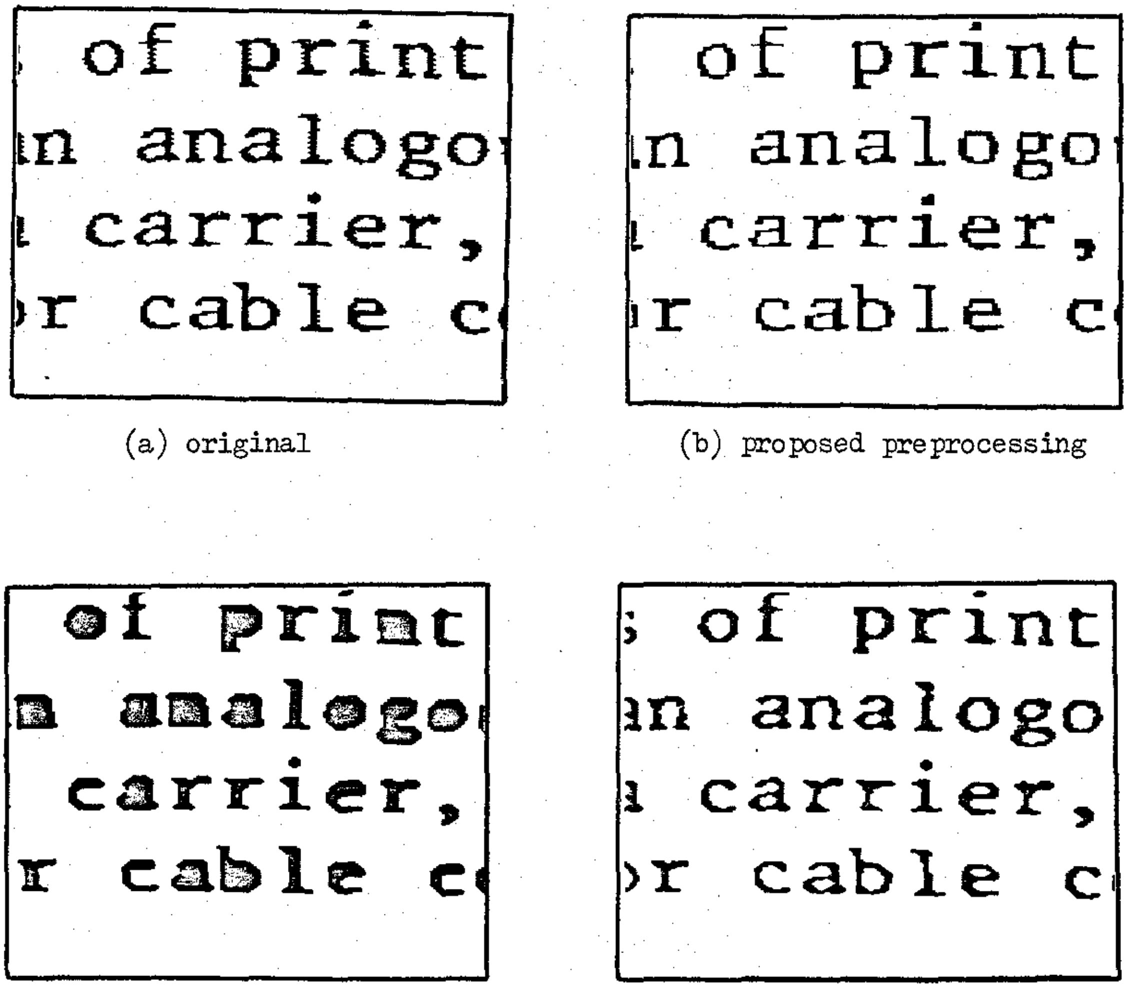
D is considered as the 'don't care' situation. Similar arguments hold for the set of masks shown in Fig. 5.1(b).

5.2.3 Results

Simulations were carried out on a 256 x 256 element area of CCITT document No. 1, where the resolution is 8 pels/mm both horizontally and vertically. The masks are sequentially employed in a sliding manner, whereby the central element of the present mask is dependent on the previously changed elements. This has the advantage of not requiring any extra memory capacity.

The result of using the proposed preprocessing technique is shown in Fig. 5.5, and comparison is made with the method of Ting et. al(15). The original image with jagged edges and notches is shown in Fig. 5.5(a). By employing the masks of Fig. 5.1, the quality of the image is considerably improved, as shown in Fig. 5.5(b). All the jagged edges have been removedand the rounded corners sharpened. At the same time both the connectivity and the essential detail of the image are also preserved. The application of the Majority Logic Smoothing with Contour Preservation technique(15) using a sliding window dependent on the previously changed elements results in the image shown in Fig. 5.5(c). The effect is one of 'filling up' the holes present in the text. However, if the central element does not depend on the previously changed elements, i.e. only on the original arrangement of elements, the image of Fig. 5.5(d) is obtained. This has the effect of thickening the image detail and not compensating for the unwanted notches. For implementation, extra memory capacity is required to store the preprocessed image.

As a measure of performance, the two-dimensional 7th order Markov



(a) original

(b) proposed preprocessing

(c) Majority logic smoothing with contour preservation. Present central pel dependent upon previous changed pels.

(d) Majority logic smoothing with contour preservation. Present central pel independent of previous changed pels.

Fig. 5.5

model predictor[60] shown in Fig. 5.6 is used to determine the improvement in coding efficiency which results when the input signal has been preprocessed using the described technique.

X_7	X_6	X_5	X_4	X_3
X_2	X_1	X_0		

Fig. 5.6 - 7th order Markov model predictor

The predictor predicts the present element X_0 using the previous seven elements, $X_1, X_2, \ldots\ldots\ldots X_7$, and the prediction errors generated are assumed to form a memoryless binary information source where the entropy is:-

$$H = - q \log_2 q - (1-q) \log_2 (1-q) \qquad (5.3)$$

bits/pel and q is the probability of correct prediction.

From Table 5.3, it can be seen that the proposed method of preprocessing yields the lowest entropy, there being an improvement of about 24 percent compared with that of the original, unprocessed image.

IMAGE	Original	Majority Logic Smoothing (Fig. 5.5d)	Proposed preprocessing
ENTROPY	0.255	0.226	0.194

Table 5.3 - Entropy comparison

5.2.4 Post-processing

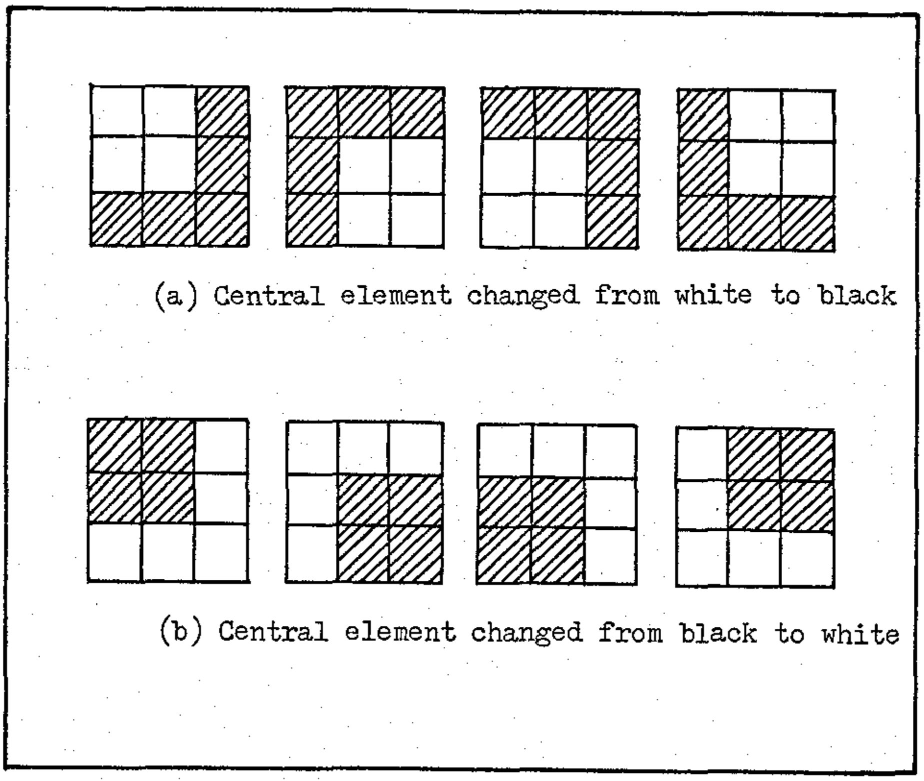
The pre-processing technique employed at the transmitter has the effect of squaring off the corners of the letters which makes the text look slightly unnatural. This can be remedied at the receiver by employing restoration masks as shown in Fig. 5.7.

Again these masks are applied sequentially to the preprocessed image with the central element changed from white to black or vice-versa if the predetermined pattern is satisfied. For Fig. 5.7(a), the central element is changed from white to black and for Fig. 5.7(b), from black to white. The result of applying the masks of Fig. 5.7(a) and Fig. 5.7(b) is shown in Fig. 5.8(a). All the sharp corners have been rounded off, but, at the same time, one or two unwanted notches have been recreated, which is undesirable. By employing the masks of Fig. 5.7(b) only, the image of Fig. 5.8(b) results. Informal subjective tests favour this image rather than the previous one. Clearly, this image is more pleasing to the eye and its naturalness is greatly enhanced. Since the preprocessing technique employed is irreversible, the original image cannot be restored completely and, in any case, it is undesirable to reintroduce the original notches.

5.3. PRE- AND POST-PROCESSING BY SUBSAMPLING AND INTERPOLATION

5.3.1 Preprocessing

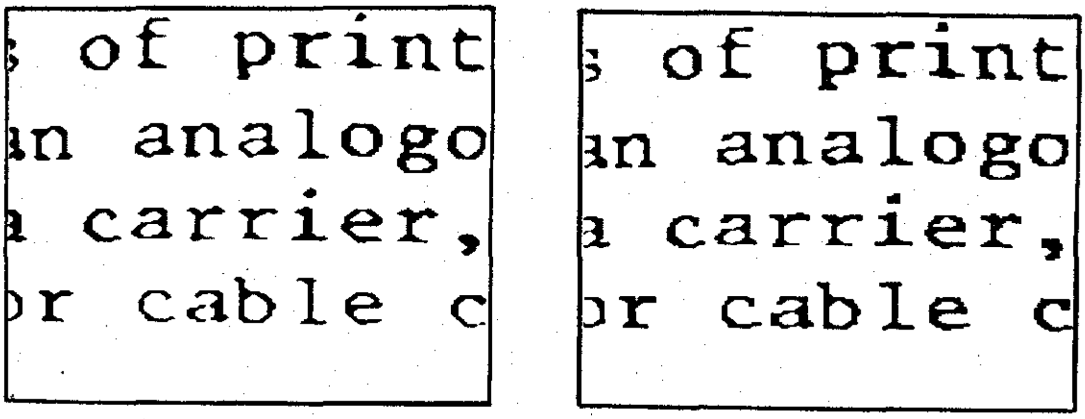
The basic stages of preprocessing by subsampling are shown in Fig. 5.9. In order to improve coding efficiency, notches and



(a) Central element changed from white to black

(b) Central element changed from black to white

Fig. 5.7 - Masks used for post-processing



(a) Application of masks of
Fig. 5.7(a) and 5.7(b)

(b) Application of masks of
Fig. 5.7(a) only

Fig. 5.8 - Restored images

pinholes in the original scanned image, as seen in Figs. 5.10(a) and 5.10(b) (which are parts of CCITT documents No. 1 and No. 2 respectively), are first removed by the notch/pinhole remover. Single element runs are then doubled in both the horizontal and vertical directions to preserve connectivity, after which subsampling is carried out.

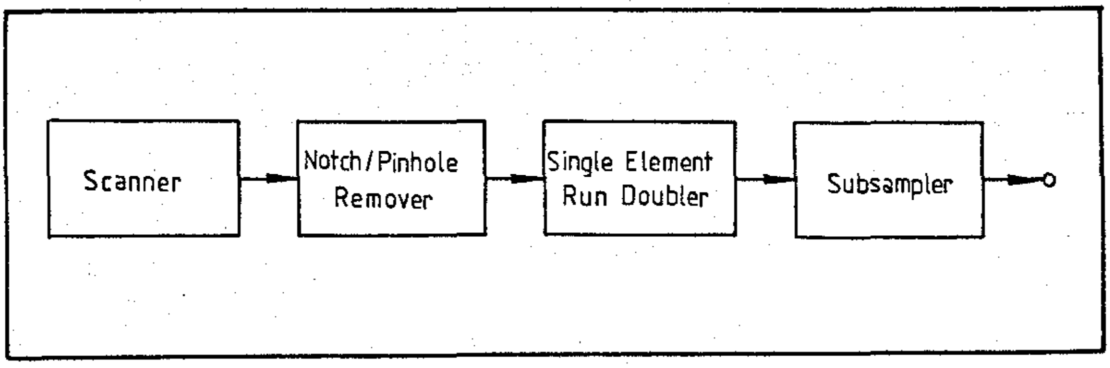


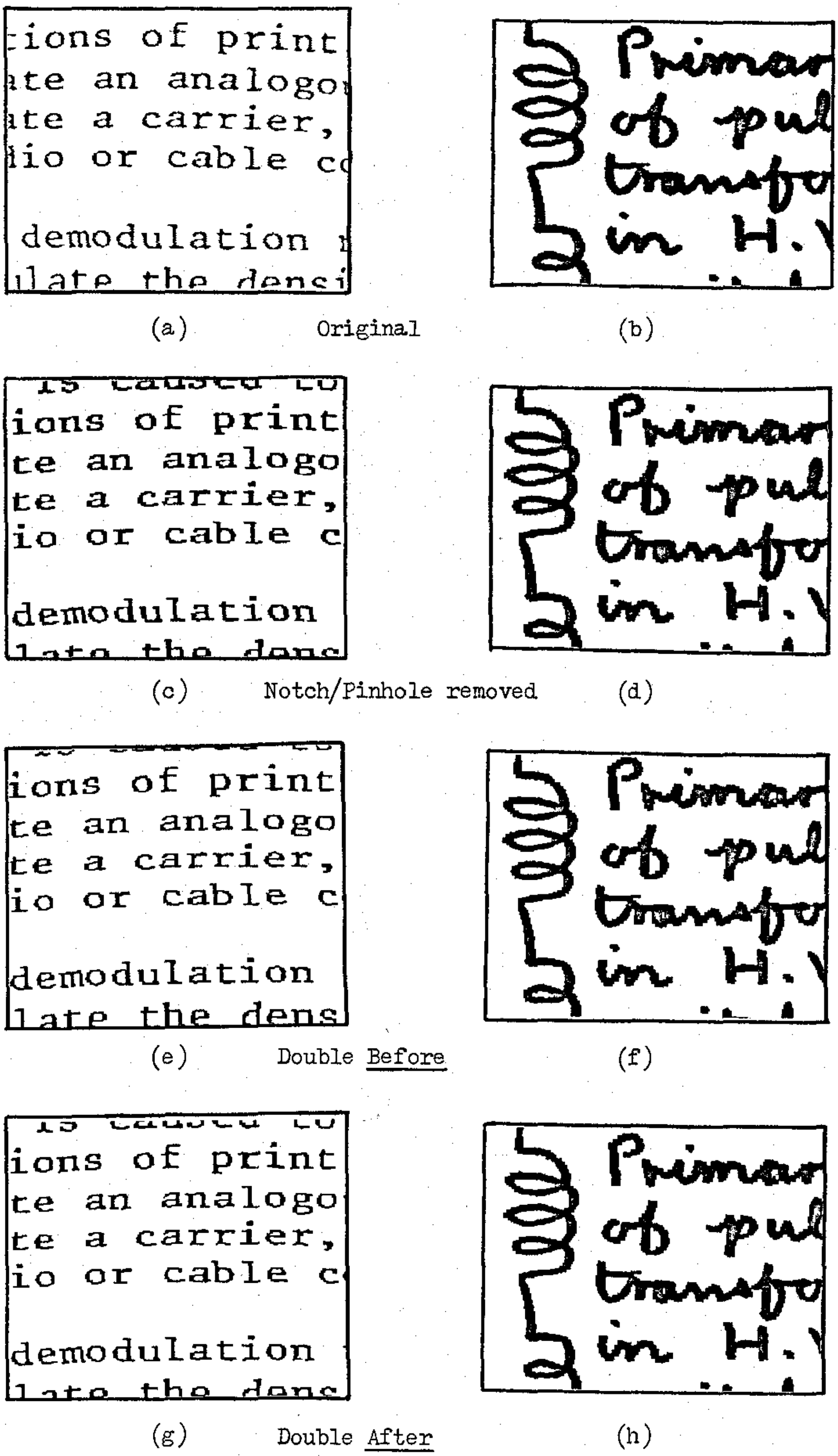
Fig. 5.9 - Preprocessing stages

(A) Notch/Pinhole Remover

Basically, this is the set of masks previously described in Section 5.2.1, which removes notches and pinholes resulting from imperfections in the scanning process. It is observed from Section 5.2.3 that the removal of these notches not only improves coding efficiency but, to a certain extent, image quality as well. Figs. 5.10(c) and 5.10(d) show the resulting images.

(B) Single Element Run Doubler

Since the subsampling process involves either taking alternate



Original

Notch/Pinhole removed

Double Before

Double After

Fig. 5.10

pels both horizontally and vertically (or in a zig-zag fashion), or the use of Hadamard or Cosine Transforms, it is necessary to double every single element black line or character. Although the probability of single element runs is low, the doubling process ensures that connectivity is preserved in the subsampled image. The preservation of connectivity is important because when the subsampled image is enlarged at the receiver, any 'disconnectivity' becomes more pronounced, thus lowering the image quality. Two doubling procedures were tried; placing a black pel before, and placing a black pel after, a one pel thick black line, in both the horizontal and vertical directions. The justification of doubling only black lines one pel thick instead of both black and white lines is due to the fact that most documents typically handled in an office environment have a white background with black printed or written characters. In any case, the volume of documents handled with a black background is normally small. Figs. 5.10(e) and 5.10(f) show the images which result when a black pel is placed before a one pel thick black line while Figs. 5.10(g) and 5.10(h) show the images for the case where a black pel is placed after a one pel thick black line. From informal observation, the former doubling process is preferred and is thus used in subsequent experiments.

(C) Subsampler

Different methods of subsampling have been investigated and their subjective results evaluated. Amongst these is subsampling by taking pels according to the configuration as shown in Fig. 5.11.

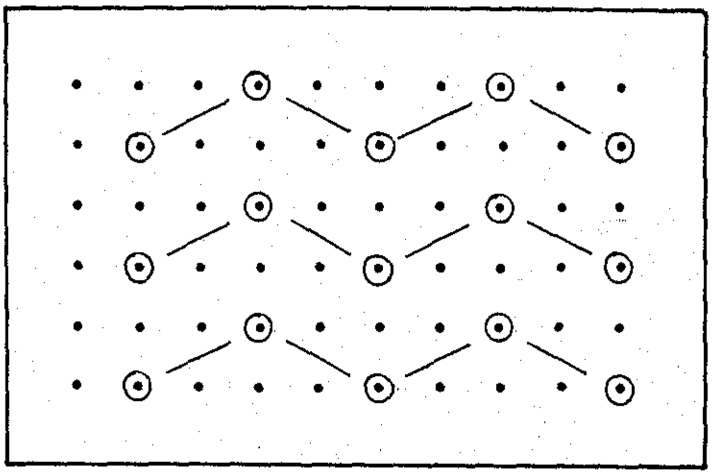


Fig. 5.11 - One of the subsampling configurations

This configuration was principally chosen with the intention of taking into account the skewness of the characters in order to preserve contour direction. The resulting image, as seen in Fig. 5.12(a), turned out to be very displeasing to the eye with some characters disconnected. It is observed that notches and pinholes are created, which will definitely reduce the coding efficiency. The impairment caused by this subsampling configuration is so severe that it can hardly be considered acceptable.

The application of Hadamard and Cosine Transforms as a means of subsampling multilevel monochrome pictures has been demonstrated by Ngan and Clarke(90). Furthermore, the theory and application of Hadamard and Cosine Transforms are well-documented in, for example, Pratt(91), Wintz(92) and Kak(93). Basically, in transform coding, a unitary mathematical transform is performed on the image data to produce a set of, at most, weakly correlated transform coefficients. These coefficients have an energy distribution

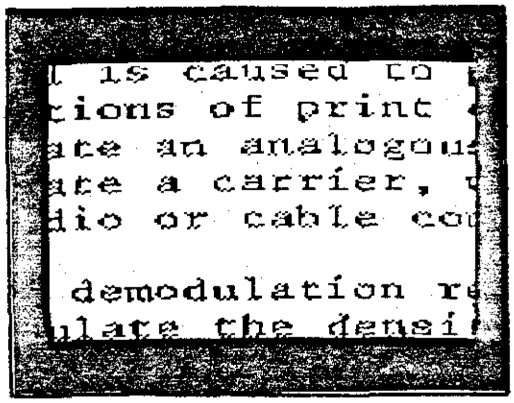
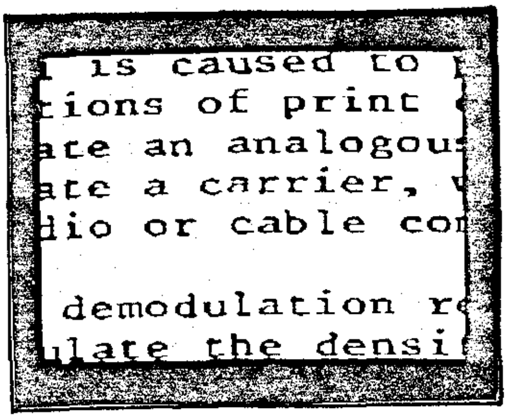


Fig. 5.12(a) - Zig-zag subsampling



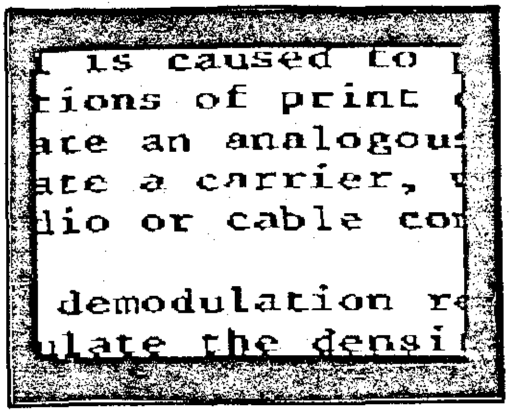
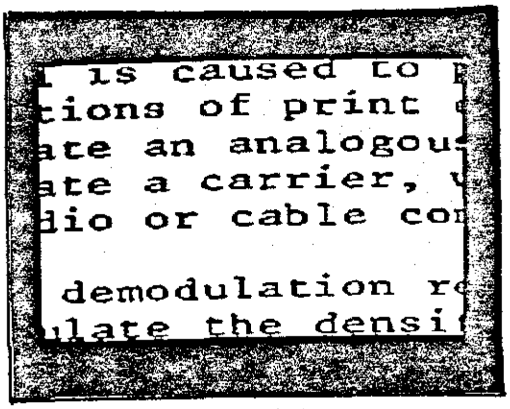


Fig. 5.12(b)
Before thresholding

Hadamard Transform
subsampling

Fig. 5.12(c)
After thresholding



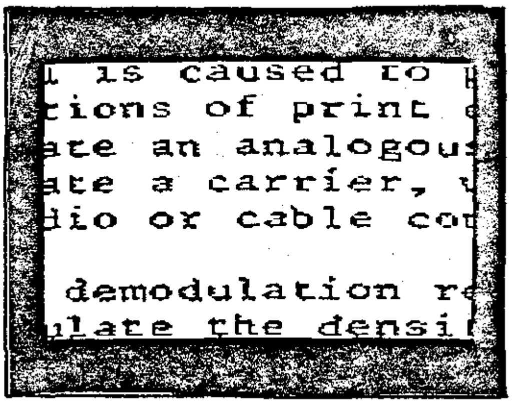


Fig. 5.12(d)
Before thresholding

Cosine Transform
subsampling

Fig. 5.12(e)
After thresholding

more suitable for coding than the original spatial domain representation. The energy in the transform domain tends to be concentrated into a relatively small number of transform coefficients and, by discarding those of low magnitude, substantial bandwidth reductions can be achieved without introducing serious image degradation.

Since these transforms have been successfully applied to subsample multilevel monochrome images, their application to binary images was considered to be an interesting proposition. The basic premise of subsampling stems from the fact that most of the energies in the transform domain appears in a certain prespecified zone, generally, in the upper left hand corner of the block adjacent to the D.C. coefficient. By discarding the less important coefficients in a predetermined fashion and inverse transforming the remaining coefficients using a smaller block size, the desired subsampling can be achieved. The subsampling which results from this operation is dependent upon the high energy compaction property of the transforms used.

Diagramatically, the subsampling operation using transform techniques is shown in Fig. 5.13. The binary image, after the doubling process, is divided into sub-blocks of size 32 x 32, after which the Hadamard or Cosine Transform is applied to each sub-block, resulting in blocks of 32 x 32 transform coefficients. Since most of the dominant coefficients are packed into the upper left hand corner, the first 16 x 16 transform coefficients (including the D.C. coefficients) are retained, as

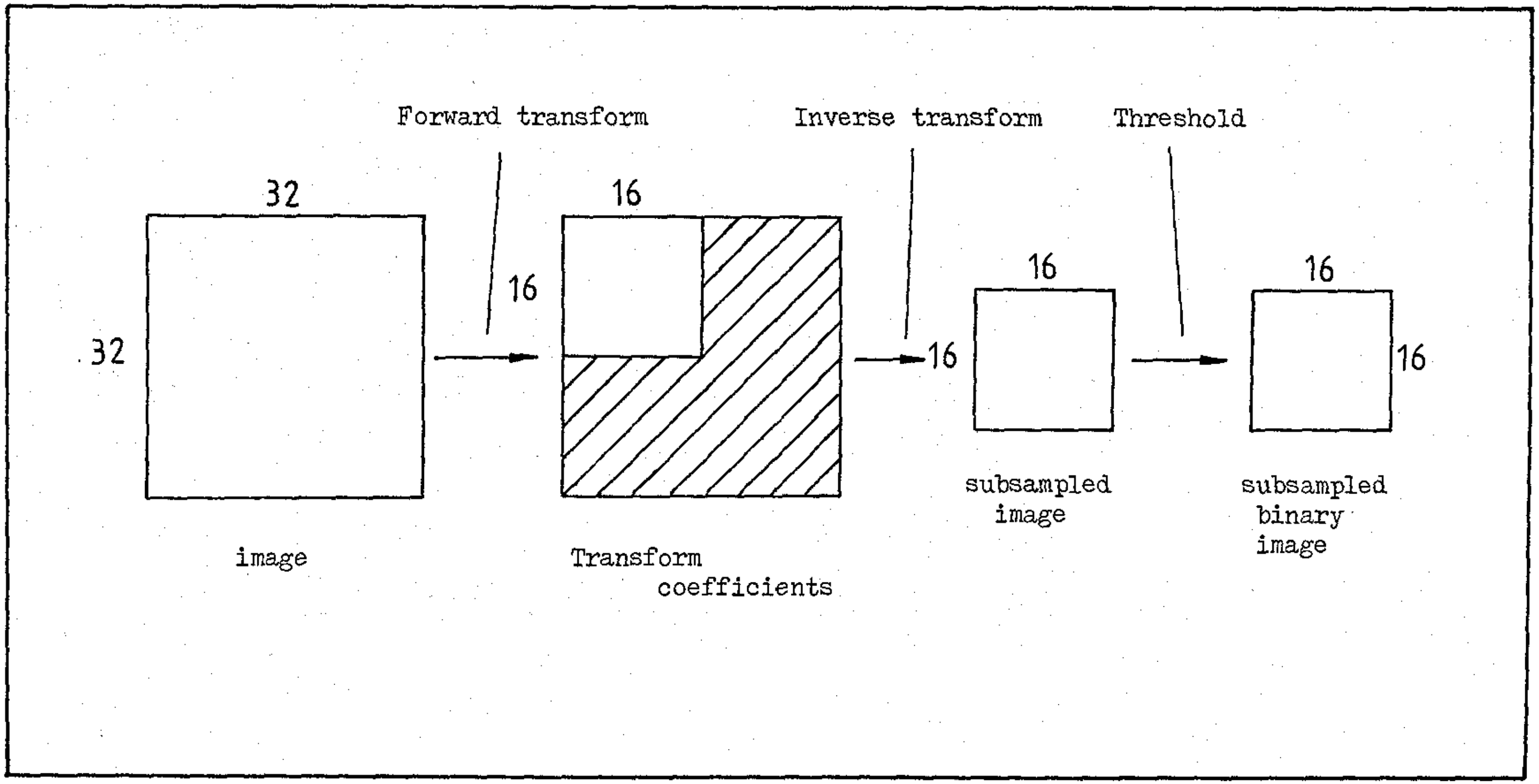


Fig. 5.13 - Subsampling using Transform techniques

shown in Fig. 5.13, while the rest (shown as the shaded area) are discarded, in the hope that their ommission will cause insignificant visual distortion. A 16 x 16 Hadamard or Cosine Inverse Transform is then applied to the 16 x 16 retained coefficients, and results in the subsampled image. This is then suitably thresholded to give a two level black and white output. The threshold was taken to be 140, i.e., values greater than or equal to 140 are taken to be black and those less than 140 are taken to be white.

The Hadamard and Cosine Transform subsampled images before and after thresholding are shown in Figs. 5.12(b) and (c) and Figs. 5.12(d) and (e) respectively. The subsampled images before thresholding looked promising at first although some ringing is apparent on the Cosine Transform subsampled image. After thresholding, notches and pinholes become more pronounced, making the results less desirable. Some of the characters are joined together while in others, 'disconnectivity' are observed. Another undesirable effect of subsampling by the use of transformation of black and white images is that the thickness of the characters becomes uneven, leading to the image showing significant distortion. Hence, it is concluded that, although a high degree of success is achieved using transform techniques for subsampling monochrome images, their application to binary images for the same purpose is not appropriate. Furthermore, the use of transforms would be complex to implement making the scheme less attractive still.

The simplest technique, giving by far the most acceptable quality, is subsampling by taking alternate pels horizontally and vertically as shown in Fig. 5.14.

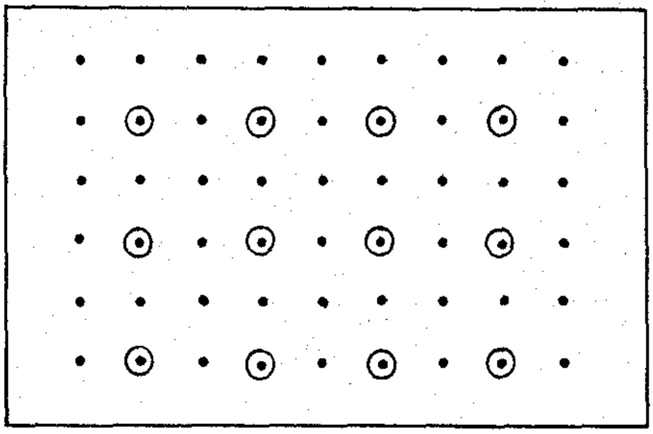
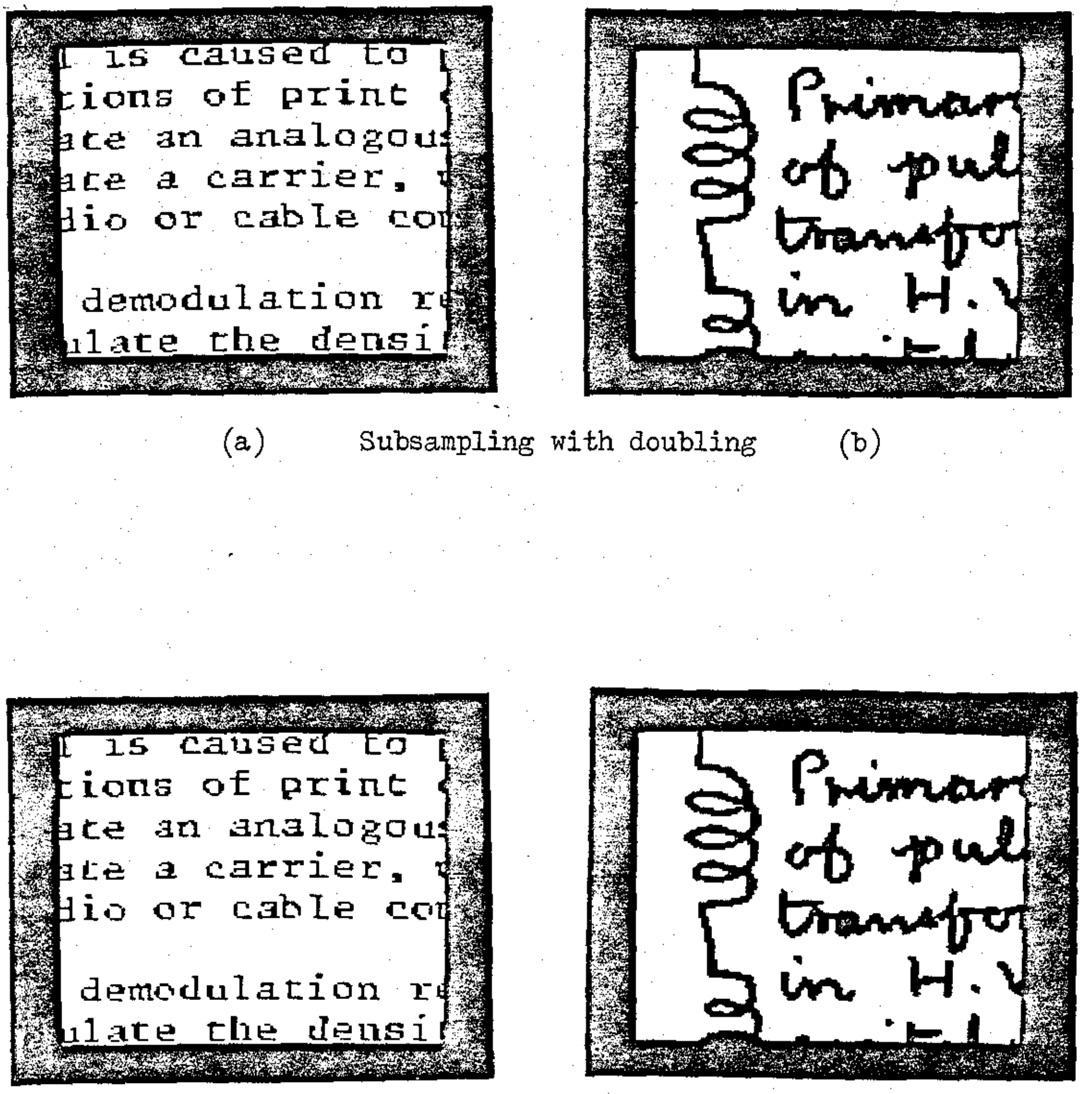


Fig. 5.14 - Subsampling by taking alternate pels

The 4:1 sample rate reduction results in the achievement of high compression and, with all the single element runs doubled, the loss of information is kept to a minimum. Figs. 5.15(a) and 5.15(b) show the effect of subsampling with doubling for parts of documents Nos. 1 and 2 respectively. Comparing them with Figs. 5.15(c) and 5.15(d) which show the results without doubling, it is noticed that the doubling process preserves connectivity whereas without doubling some discontinuities are apparent. Such discontinuities are more noticeable in document No. 1 than document No. 2. It is also observed that notches and pinholes are not recreated. Due to its simplicity and its effectiveness, subsampling by taking alternate pels horizontally and vertically has been adopted as the subsampling technique used in further simulations.



(a) Subsampling with doubling (b)

(c) Subsampling without doubling (d)

Fig. 5.15 - Subsampling by taking alternate pels horizontally and vertically

5.3.2 Post-processing

It is evident from the previous sub-section (5.3.1) that subsampling reduces the data rate by a factor of 4:1 and inevitably causes some loss of information in the original image. Since preprocessing by subsampling is irreversible, it is impossible to restore fully the subsampled image to its original form. In any case, it is not desirable to reintroduce notches and pinholes, which were previously removed, into the image. A visually pleasing effect which represents the original as closely as possible can be obtained by employing post-processing at the receiver. In this sub-section, the investigations carried out to achieve that effect are described. The various stages of post-processing are shown in Fig. 5.16.

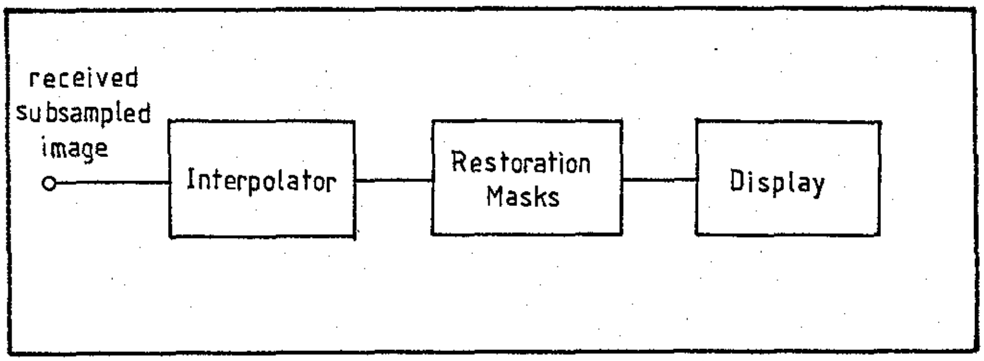


Fig. 5.16 - Post-processing stages

The subsampled image is first enlarged to its original size using three mathematical interpolation functions namely replication, bilinear, and Cubic B-spline interpolation. The interpolation functions are mainly used to create an artificial (not actual) improvement in resolution and comparisons between these three schemes have been made. Naturally, since two level images are considered, some form of thresholding is required. The application of restoration masks to 'cosmetically' improve the resulting image is also described. In all case, pictorial illustrations are presented for viewer interpretation. To confirm informal preferences between the restored images, a subjective experiment was also conducted.

(A) Interpolator

For interpolation, the use of spline functions of degree lower than four is of particular interest. Spline functions are a class of piecewise polynomial functions satisfying continuity properties only slightly less stringent than those of polynomials. Polynomials have long been the functions most widely used to approximate other functions, mainly because they have simple mathematical properties. However, polynomials of moderately high degree fitted to a fairly large number of given data points tend to exhibit more numerous and more severe undulations than a curve drawn using a spline function(94). Since the work of Schoenberg(95) in the mid 1940's, the use of spline functions especially for interpolation, has gained widespread attention in various branches of electrical engineering(96-100). In the field of image processing, Andrews and Peterson(101) and Hou and Andrews(102-103) have adopted spline interpolation in their contributions to image processing.

The choice of spline functions rather than of the sinc function or other classical polynomial approaches is motivated by the advantages that the former possess. Firstly, unlike the latter, the spline function spans a finite set of data points, i.e. its support is local. The behaviour of polynomials, on the other hand, is totally determined by the behaviour of samples over other intervals. Thus, polynomial interpolation is in no sense a local procedure. That is, if the function to be interpolated varies rapidly in some part of the region of interest, the effect of this on interpolation is apparent everywhere. The sinc function, for example, spans an infinite number of data points and if truncated, oscillations(Gibb's phenomenon) will show up in the resultant image. Secondly, the spline function is relatively easy to implement and in both hardware and software form has been employed by Hou and Andrews(103). Lastly, the spline function is non-negative and is thus obviously attractive for image processing applications. The non-negative property is a practical consideration, as interpolating picture elements, and therefore energy (intensity) values, must result in non-negative images. This precludes, in a practical way, the use of the sinc function which has negative lobes, as an interpolant.

Mathematically, the interpolated continuous function in one dimension is :-

$$\hat{f}(\theta) = \sum_{k=1}^{K} C_k \, S_k(\theta) \tag{5.4}$$

where C_k are the coefficients to be determined from the input data, $S_k(\theta)$ are the chosen basis functions and K is the number of data points. Assuming separability, the one dimensional case can be extended to two dimensions as follows:-

$$\hat{f}(\theta,\eta) = \sum_{k=1}^{K} \sum_{\ell=1}^{L} C_{k\ell}\, S_k(\theta)\, S_\ell(\eta) \qquad (5.5)$$

Let $\theta_0, \theta_1, \theta_2, \ldots\ldots, \theta_{n+1}$ be a partition of interval $[\theta_0, \theta_{n+1}]$ on a real axis. The basis function for a one-dimensional B-spline of degree n can be written as:-

$$B_n(\theta_0, \theta_1, \ldots, \theta_{n+1}) = (n+1) \sum_{k=0}^{n+1} \frac{(\theta-\theta_k)^n\, U(\theta-\theta_k)}{\omega(\theta_k)} \qquad (5.6)$$

where $\omega(\theta_k) = \prod_{\substack{j=0 \\ j\neq k}}^{n+1} (\theta_k - \theta_j)$

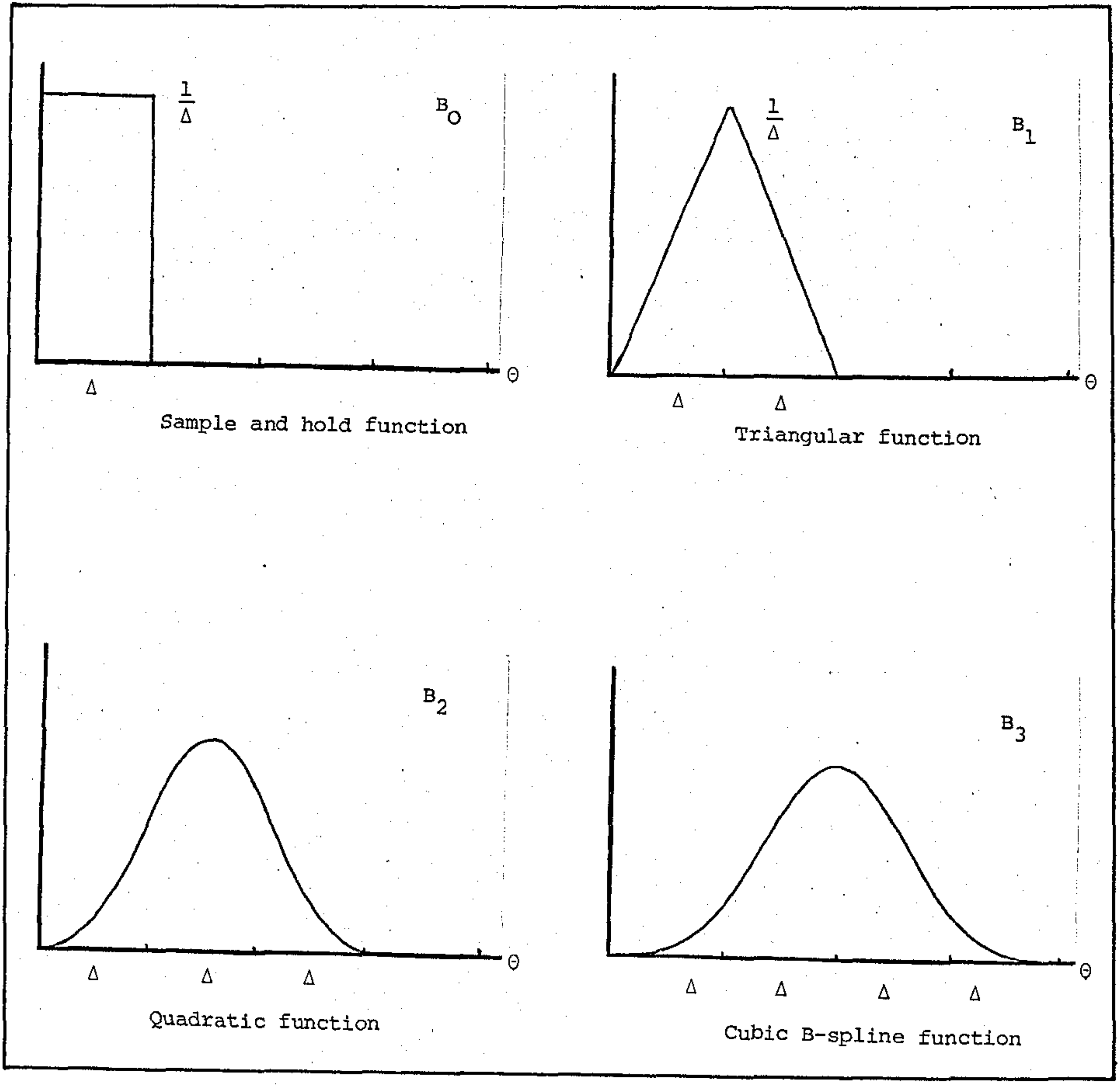
$$U(\theta-\theta_k) = \begin{bmatrix} (\theta-\theta_k)^0 & \text{for} & \theta > \theta_k \\ 0 & \text{for} & \theta \leqslant \theta_k \end{bmatrix}$$

i.e. a unit step function

and n = 0, 1, 2, ...

Fig 5.17 shows the first four lower order spline functions for uniformly spaced data points (knots) where $\Delta = \theta_k - \theta_{k-1}$. Notice that B_n is shift invariant, strictly positive and spans a finite interval on the real axis, i.e. B_n has the property of local support. It can be seen that, for a uniformly spaced data point $B_1 = B_0 * B_0$, $B_2 = B_0 * B_0 * B_0$ and $B_3 = B_0 * B_0 * B_0 * B_0$ where $*$ denotes convolution.

The simplest interpolation waveform is the sample and hold function,



Sample and hold function

Triangular function

Quadratic function

Cubic B-spline function

Fig. 5.17 - B-spline functions

B_0 (zero-order interpolator) which will perform replication. Convolution of the sample and hold function with itself yields the triangular function, B_1, which will perform first order linear interpolation. The convolution of the triangular function with the sample and hold function produces the quadratic function, B_2, and lastly, convolving B_2 and B_0 results in the Cubic B-spline function, B_3. The interpolated result, using the basis function, B_3, is composed of a sequence of third degree polynomials which joins continuously at the knots in amplitude and slope. Here, replication, linear (bilinear in two dimensions) and Cubic B-spline interpolation functions are of interest for the reconstruction of subsampled binary images.

From Eqn. (5.6), the Cubic B-spline basis function for uniformly spaced knots is given by:-

$$\begin{aligned} S(\theta-\theta_k) &= B_3(\theta_{k-2}, \theta_{k-1}, \theta_k, \theta_{k+1}, \theta_{k+2}) \\ &= \Big[(\theta-\theta_{k-2})^3\, U(\theta-\theta_{k-2}) - 4(\theta-\theta_{k-1})^3\, U(\theta-\theta_{k-1}) \\ &\quad + 6(\theta-\theta_k)^3\, U(\theta-\theta_k) - 4(\theta-\theta_{k+1})^3\, U(\theta-\theta_{k+1}) \\ &\quad + (\theta-\theta_{k+2})^3\, U(\theta-\theta_{k+2})\Big]\, \frac{1}{6\Delta^4} \end{aligned} \qquad (5.7)$$

Substitution of the basis function of Eqn.(5.7) into Eqn. (5.4) and Eqn. (.5) will determine the interpolated values between knots. Consider $\hat{f}(\theta)$ (one-dimensional case) at:-

$$\theta = \theta_k + x\Delta \qquad 0 \leqslant x \leqslant 1 \tag{5.8}$$

From the definition of the Cubic B-spline basis function of Eqn. (5.7), $\hat{f}(\theta)$ in Eqn. (5.4) becomes:-

$$\begin{aligned}\hat{f}(\theta) = \frac{1}{6\Delta^4} \Big\{ & C_{k-1} \left[(\theta-\theta_{k-3})^3 - 4(\theta-\theta_{k-2})^3 + 6(\theta-\theta_{k-1})^3 - 4(\theta-\theta_k)^3\right] \\ & + C_k \left[(\theta-\theta_{k-2})^3 - 4(\theta-\theta_{k-1})^3 + 6(\theta-\theta_k)^3\right] \\ & + C_{k+1} \left[(\theta-\theta_{k-1})^3 - 4(\theta-\theta_k)^3\right] \\ & + C_{k+2} \left[(\theta-\theta_k)^3\right] \Big\}\end{aligned} \tag{5.9}$$

Substituting Eqn. (5.8) into the above, we have:-

$$\begin{aligned}\hat{f}(\theta_k + x\Delta) = \frac{1}{6\Delta} \Big\{ & C_{k-1} \left[(3+x)^3 - 4(2+x)^3 + 6(1+x)^3 - 4x^3\right] \\ & + C_k \left[(2+x)^3 - 4(1+x)^3 + 6x^3\right] \\ & + C_{k+1} \left[(1+x)^3 - 4x^3\right] \\ & + C_{k+2} \; x^3 \Big\}\end{aligned} \tag{5.10}$$

Eqn. (5.8) and Eqn. (5.9) determine the interpolation at any point between knots. In particular for $\theta = \theta_k$, i.e. $x = 0$, Eqn. (5.10) gives

$$\hat{f}(\theta_k) = \frac{1}{6\Delta} \left(C_{k-1} + 4C_k + C_{k+1}\right) \tag{5.11}$$

In vector form, for $\hat{f}(\theta)$ at the node points, we have, from Eqn. (5.10) and Eqn. (5.11),

$$\underline{\hat{f}} = E\,\underline{C} \qquad (5.12)$$

where

$$E = \frac{1}{6\Delta}\left[\begin{array}{ccccccc} 4 & 1 & & & & & \\ 1 & 4 & 1 & & & 0 & \\ & 1 & 4 & 1 & & & \\ & & 1 & 4 & 1 & & \\ & & & 1 & \ddots & 1 & \\ & 0 & & & \ddots & 4 & 1 \\ & & & & & 1 & 4 \end{array}\right] \qquad (5.13)$$

and $\underline{\hat{f}} = \left[\hat{f}(\theta_1), \hat{f}(\theta_2), \ldots\ldots\hat{f}(\theta_k)\right]^t$

$\underline{C} = \left[C_1, C_2, \ldots\ldots\ldots C_k\right]^t$

$\underline{\hat{f}}$ and $\underline{C}$ denote the vector representation of $\hat{f}$ and C and $[\;]^t$ denotes the transpose of a matrix.

Similar derivations can be applied to the two-dimensional case. Consider $\hat{f}(\theta, \eta)$ at the point:-

$$\theta = \theta_k + x\Delta \qquad \text{and} \qquad \eta = \eta_\ell + y\Delta$$

where $0 \leq x \leq 1$ and $0 \leq y \leq 1$

Using the same argument as for the one-dimensional case, the interpolated value at the node point (θ_k, η_ℓ), i.e. $x = 0$ and $y = 0$ is:-

$$f(\theta_k,\eta_\ell) = \frac{1}{36\Delta^2}\Big[(C_{k-1,\ell-1} + 4C_{k,\ell-1} + C_{k+1,\ell-1})$$

$$+ 4(C_{k-1,\ell} + 4C_{k,\ell} + C_{k+1,\ell})$$

$$+ (C_{k-1,\ell+1} + 4C_{k,\ell+1} + C_{k+1,\ell+1})\Big] \qquad (5.14)$$

for all $k = 1, 2, \ldots\ldots K$ and $\ell = 1, 2, \ldots\ldots\ldots L$

Written in matrix form, Eqn. (5.14) becomes:-

$$F = E\ C\ E \qquad (5.15)$$

where F is a matrix composed of input samples at the knots, C is a matrix with elements $C_{k,\ell}$ and E is given by Eqn. (5.13)

Two-dimensional Cubic B-spline interpolation is carried out by determining the C coefficient matrix first from an inverse interpolation operation on the input data F, viz:-

$$C = E^{-1}\ F\ E^{-1} \qquad (5.16)$$

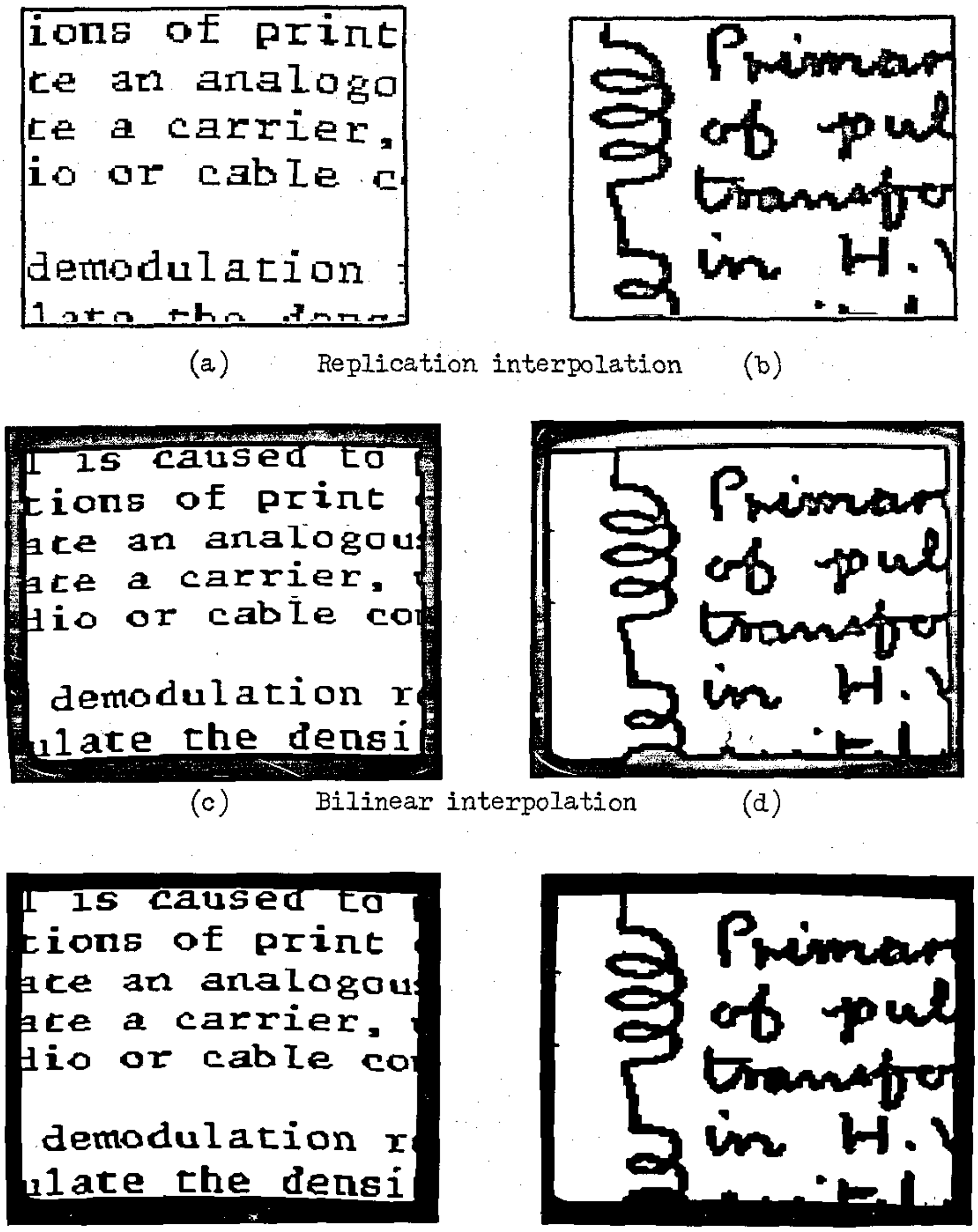
Having found the coefficients C from the input data F, the one-dimensional interpolation formula in Eqn. (5.4) is applied first to every row, and then to every column, of C (assuming separability).

Experimental results for facsimile image reconstruction and enlargement are now presented and comparisons between the different interpolation schemes made. Replication is carried out by repeating each pel inside an m x m square where m is a linear magnification factor. In other words, the interpolation basis function is the sample and hold function, B_0. For bilinear interpolation, the process is carried out on every row of data and then on every column of data using the following interpolation formula:-

$$\hat{f}(k+r) = r\ \hat{f}(k+1) + (1-r)\ \hat{f}(k) \tag{5.17}$$

where $0 \leq r \leq 1$. The interpolation basis function is therefore the triangular function, B_1. In the case of the Cubic B-spline, a subroutine was developed to perform the basis function, B_3.

Fig 5.18 shows the results of enlarging the subsampled image to its full size using the three interpolation schemes described. Figs. 5.18(a) and 5.18(b) are the images obtained from replication, Figs. 5.18(c) and 5.18(d), from bilinear interpolation and Figs. 5.18(e) and 5.18(f) are the images resulting from Cubic B-spline interpolation. These images have not been thresholded, and it is clear that the nearest neighbour replication produces an image with a very 'blocklike' structure which is rather displeasing but however, not too objectionable. Notice that the connectedness of the image, and its intelligibility, are still preserved. The higher order interpolation schemes provide psychovisually more acceptable results which are 'soft' in nature. These methods produce about the same visual reaction with little to choose between them. Since bilinear and Cubic B-spline interpolation

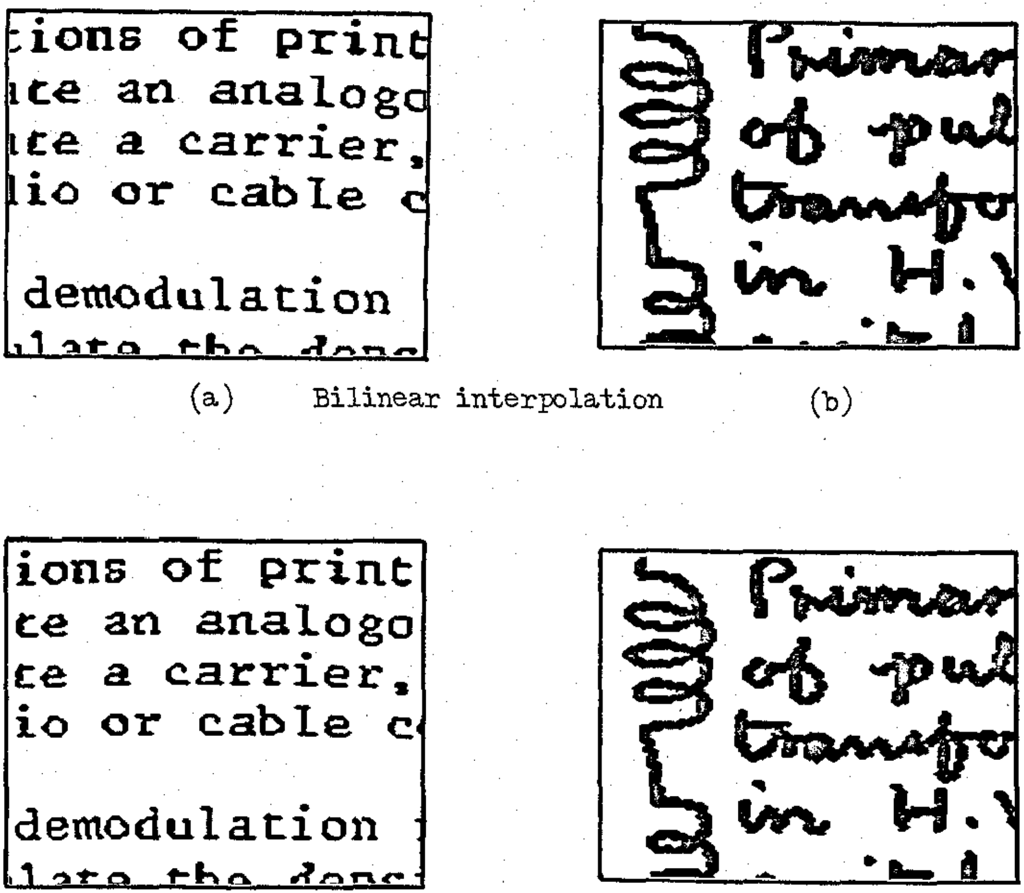


(a) Replication interpolation (b)

(c) Bilinear interpolation (d)

(e) Cubic B-spline interpolation (f)

Fig. 5.18 - Interpolated images.



(a) Bilinear interpolation (b)

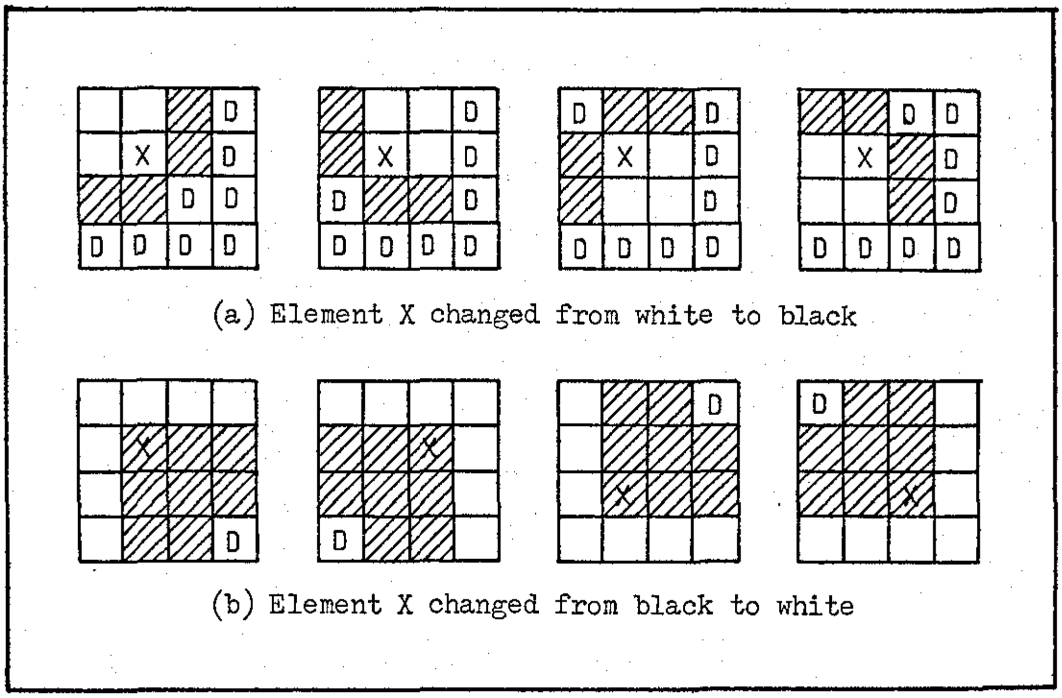
(c) Cubic B-spline interpolation (d)

Fig. 5.19 - Thresholded interpolated images

schemes create intermediate levels between black and white, thresholding is necessary if these methods are to be applied in a two-level facsimile transmission system. Fig. 5.19 shows the images for bilinear and Cubic B-spline interpolation after being suitably thresholded to give a satisfactory output. Compared to replication, bilinear interpolation, as seen in Figs. 5.19(a) and 5.19(b), has effectively produced an image which has smoother corners but is very 'triangular' in form. In some parts of the pictures, adjacent characters are connected and there is a noticeable thickening of the letters. Since in this case, there are no jagged edges, further post-processing will be ineffective. The Cubic B-spline interpolation, as shown in Figs. 5.19(c) and 5.19(d), on the other hand, is more 'rounded' but has distinct notches along the edges of some of the characters. These notches are particularly pronounced in Fig. 5.19(d). In order to improve image quality further, restoration is necessary and is carried out on the Cubic B-spline and replication interpolated images only.

(B) Restoration Masks

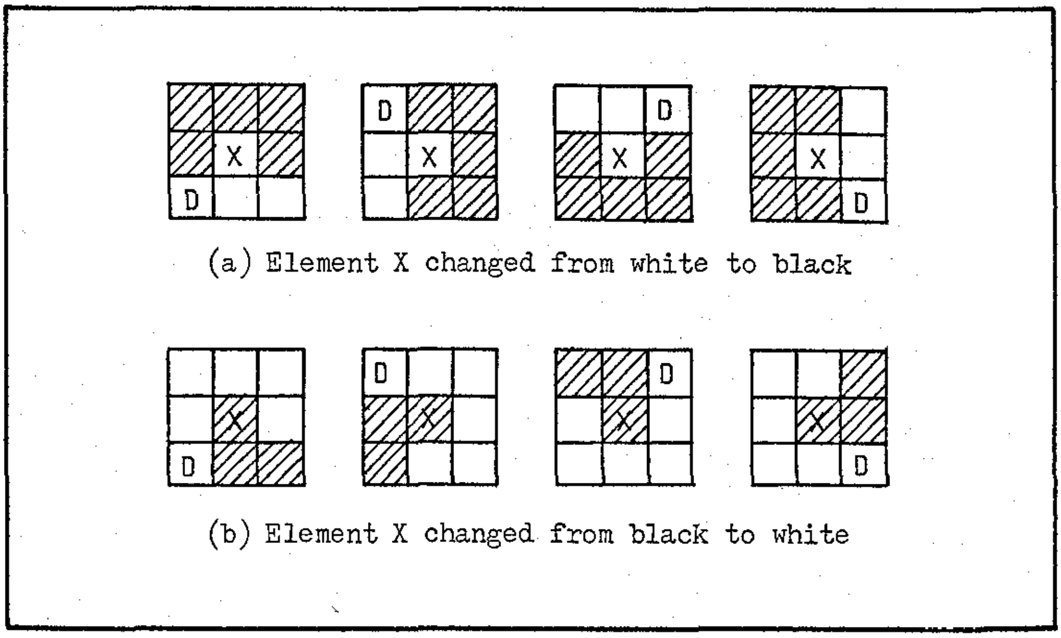
Figs. 5.20 and 5.21, respectively, show the restoration masks for the replication and the Cubic B-spline interpolated images. Fig. 5.20 differs from Fig. 5.21 in that the former requires 4 x 4 masks whereas the latter only uses 3 x 3 masks. The difference is due to the nature of the image to be restored. Basically, the restoration masks for the replication interpolated image have the functions of smoothing the sharp corners and reducing the 'blocklike' structure of the characters, whilst the restoration



(a) Element X changed from white to black

(b) Element X changed from black to white

Fig. 5.20 - Restoration masks for replication interpolation



(a) Element X changed from white to black

(b) Element X changed from black to white

Fig. 5.21 - Restoration masks for Cubic B-spline interpolation

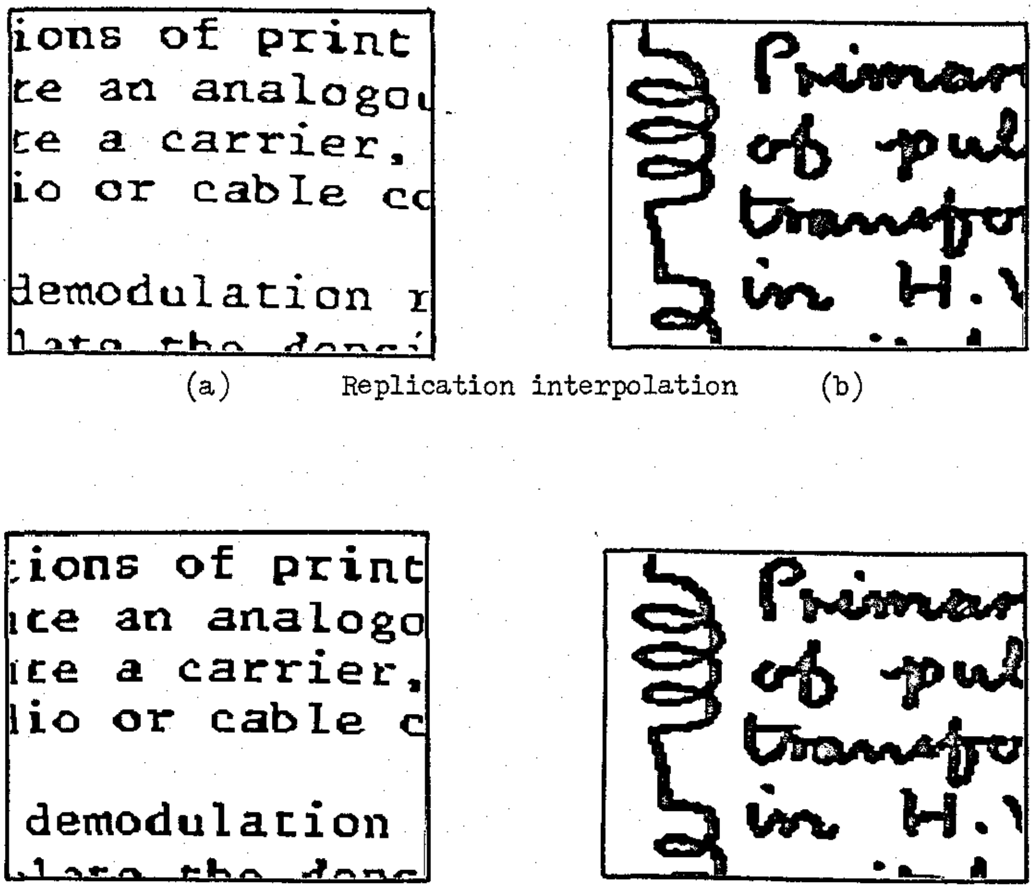
masks of the Cubic B-spline interpolated image remove notches and pinholes.

In Fig. 5.20(a) and Fig. 5.21(a), all the elements marked X are changed from white to black and in Fig. 5.20(b) and Fig. 5.21(b), from black to white. D is considered as a 'don't care' situation , in which either value is acceptable. Each mask is tested sequentially against the interpolated image within a specified window and, if the predetermined pattern is satisfied, element X is changed.

On the one hand, Figs. 5.22(a) and 5.22(b) show that the application of the restoration masks of Fig. 5.20 to the replication interpolated image has reduced the 'blocklike' structure drastically. The characters are clearly legible, more rounded and visually very pleasing. On the other hand, when the restoration masks of Fig. 5.21 are applied to the thresholded Cubic B-spline interpolated image, the effect is to eliminate, as shown in Figs. 5.22(c) and 5.22(d), the notches and pinholes which resulted from the thresholding. Again it produces an image which is 'cosmetically' pleasing but whose characters are slightly thickened. In order to assess the subjective quality and the order of preference of the three interpolation schemes after restoration, a formal subjective test was conducted, as described in the next section.

5.3.3 Subjective Testing

This section examines the subjective quality of the three interpolation



(a) Replication interpolation (b)

(c) Cubic B-spline interpolation (d)

Fig. 5.22 - Restored images

schemes after restoration. The purposes of the test were to determine how subsampling and interpolation affects the quality of the documents and to assess which of the three schemes is preferable. It is also intended to find out what trade-off, if any, exists between the resulting gains in compression ratios and the subjective quality.

(A) Test Procedure

Parts of documents No. 1 and No. 2 (Figs. 5.10(a) and 5.10(b) respectively) were used as the original test data. The images used in the experiment after restoration following replication and Cubic B-spline interpolation are those shown in Figs. 5.22(a) and 5.22(b) and Figs. 5.22(c) and 5.22(d) respectively, whereas Figs. 5.19(a) and 5.19(b) represent the bilinear interpolated images.

The original and the processed images for document No. 1 and document No. 2 were shown independently to 15 subjects of whom 4 were female. The images were arranged in a random fashion to minimise the effects of learning and influences due to the order of the images. The subjects were then instructed to assign a value between 0 and 100 to each test image to denote its subjective quality. The quality ratings for each subject were normalised such that the original had a score of 100 to provide a point of reference. The normalised results for the 15 subjects were then averaged, thereby giving only one quality rating for each test condition.

(B) Test Results

The subjective quality for the processed documents No. 1 and No. 2 is plotted on the graphs shown in Figs. 5.23(a) and 5.23(b) respectively. The ordinate represents the quality scale while the abscissa represents the images generated using the the three interpolation schemes. As it turns out, the subjective quality and the order of preference for text predominant documents are different from those containing mostly graphics. All the subjects agreed, without doubt, that the processed documents were all highly intelligible. This may be due to the doubling process which ensures that connectivity is preserved and that loss of information is kept to a minimum. For parts of documents No. 1 (see Fig. 5.23(a)), the bilinear interpolated image has the highest score next to the original, then comes Cubic B-spline interpolation followed closely by replication. Most subjects favoured the bilinear interpolated image mainly due to the thickness of the characters being more even than in the case of its counterparts. For the Cubic B-spline interpolated image, although the thickness of the characters is more or less uniform, it somewhat lacks the appeal of the bilinear interpolated image, perhaps due to the application of the restoration masks, which causes the characters to be more 'blocklike'.

The subjects found that it was more difficult to assess the subjective quality of the processed images for document No. 2. The choice was between replication and bilinear interpolated images (see Fig. 5.23(b)). Most subjects preferred replication to bilinear interpolation by only a small margin, purely because

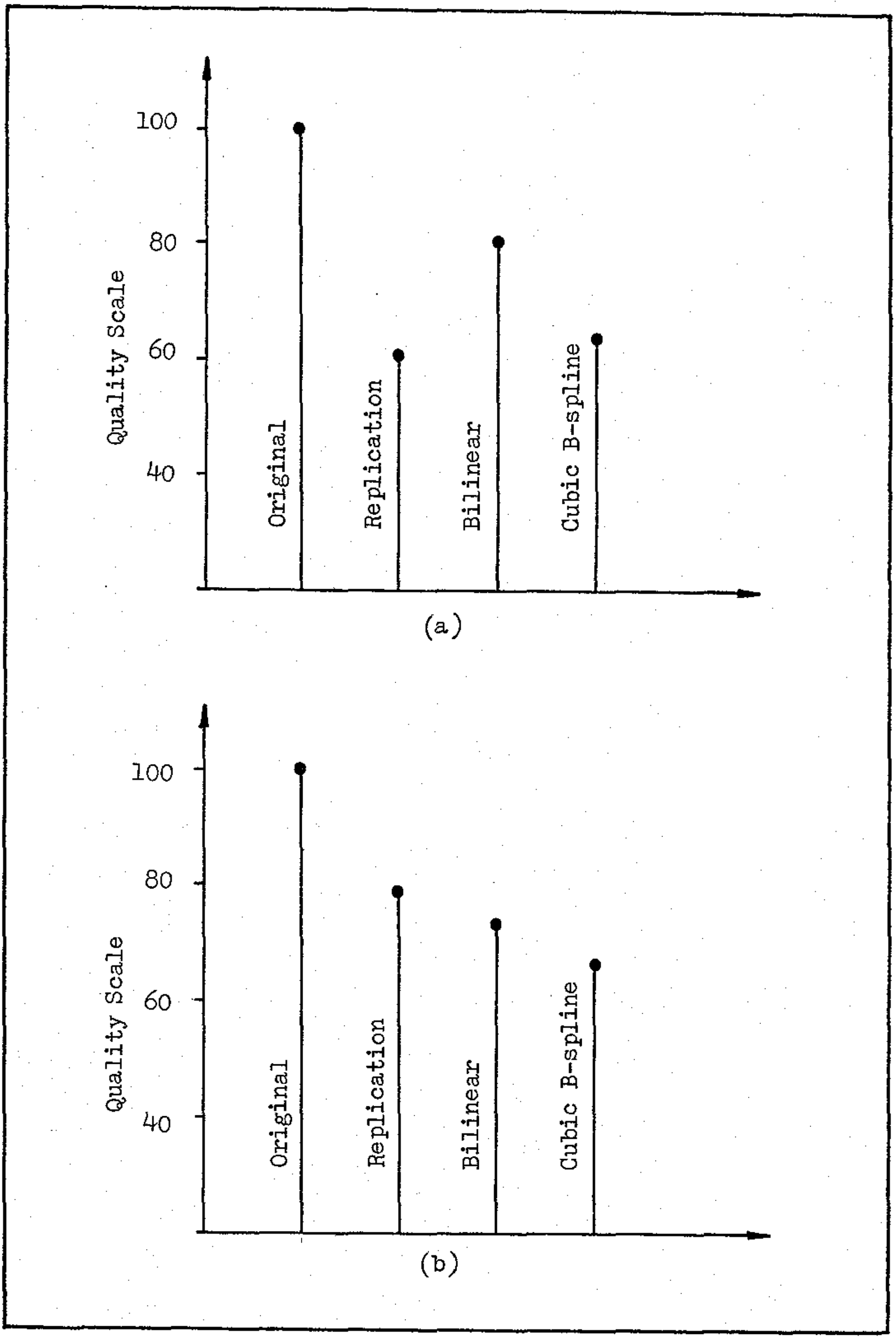


Fig. 5.23 - Subjective Quality Ratings

(a) for part of document No. 1

(b) for part of document No. 2

the former is more pleasing to the eye than the latter. As for the Cubic B-spline interpolated image, most subjects concluded that it is not appealing enough to be regarded as favourable.

For both documents, the subjective test revealed that there is a decrease in quality of about 20 percent between the most preferred image and the original. Due to subsampling, however, there is a 4:1 sample rate reduction and the drop in quality will be compensated by the improvement in compression ratios achieved. (See Chapter VII).

5.4. CONCLUSION

In this chapter, two pre- and post-processing techniques, which are prerequisites to two coding schemes to be explained in Chapter VI and Chapter VII, have been described. The first technique, pre- and post-processing using a set of masks, not only greatly reduces the amount of redundant information in the original picture but,to a certain extent, improves the image quality by removing jagged edges. The coding efficiency of the preprocessed picture has also been improved by about 24 percent compared with the original. However, as a result of applying the preprocessing technique, there is a slight 'blocking' effect on the characters, but this can be easily remedied at the receiver by the application of restoration masks, which will then preserve the naturalness of the image. Pre- and post-processing by, respectively, subsampling and interpolation is the second technique. Here, straightforward subsampling by taking alternate pels horizontally and vertically has been shown to be most suitable for facsimile images. A 4:1 sample rate reduction results in the achievement of high compression ratios. Reconstruction of the subsampled image at the receiver using three different interpolation schemes was investigated, and they were all found to give visually pleasing results. However, the application of restoration masks to the interpolated images improved the subjective quality even further. Formal subjective tests revealed that the order of preference for text predominant documents is different from that for those containing mainly graphics. For text predominant documents, enlargement of the subsampled image using bilinear interpolation is most favoured, while for graphics-type documents, replication with proper restoration produces the most 'cosmetically' pleasing effect.

5.5. NOTE ON PUBLICATIONS

A paper entitled "New Preprocessing Technique for Digital Facsimile Transmission", by M.G.B. Ismail, in co-authorship with R.J. Clarke has been published in IEE Electronics Letters, 9th May 1980, Vol. 16, No. 10, pp. 355-356. This paper is an abridged version of Section 5.2.

A brief version of Section 5.3 has been presented as part of a paper entitled "Adaptive Block/Location Coding of Facsimile Signal using Subsampling and Interpolation for Pre- and Post-processing" at the Conference of Digital Processing of Signals in Communications held at Loughborough University, Leicestershire, 7-10 April 1981.

A full paper describing Section 5.3 has appeared as part of a paper with the same title in the IEEE Transaction on Communications (Special Issue on Picture Communications System), Vol. COM-29, No. 12, December 1981, pp. 1925-1934. The papers were jointly authored by M.G.B. Ismail and R.J. Clarke.

CHAPTER VI

DATA COMPRESSION I

6.1. INTRODUCTION

The transmission and storage of two-tone (black and white) pictures, such as weather maps, circuit diagrams, printed texts, etc, has received much attention in recent years, and its practical importance is evident from the number of facsimile communication systems that are now readily available.(1,114) As the cost of electronic hardware decreases more rapidly than that of transmission, it is becoming advantageous to use sophisticated terminals to reduce transmission costs and time. Indeed, many of the recently developed facsimile communication systems have resorted to various source encoding techniques which utilise the statistical spatial redundancy between picture elements to reduce the bit rates required for transmission.

A scan-line of facsimile image consists of runs of white elements separated by runs of black elements, and picture elements close together are significantly correlated. Source encoding techniques which utilise statistical redundancy along a single scan-line are known as one-dimensional coding schemes while those using many scan-lines are considered as two-dimensional coding schemes. Research into two-dimensional coding schemes, which exploits the line-to-line correlation, together with performance comparisons of many different schemes have been the subject of numerous papers.(34,54,55,63)

In this chapter, a two-dimensional line-by-line sequential coding scheme, where each scan-line can be effectively processed using the vertical correlation between the current and the preceding line, is described. From the results of Chapter V, preprocessing prior to actual coding has the effect of improving the efficiency of facsimile coders. It is therefore natural to use a preprocessor as a prerequisite before the actual coding is carried out. The scheme to be described is called Classified Adaptive Block/Run-Length Coding, and the preprocessor used is the one already outlined in Section 5.2, in which a set of masks is employed. Classified Adaptive Block/Run-Length Coding is based on the observation that a facsimile signal can be regarded as the output of an Nth order Markov source, and that the statistics of the run-lengths conditioned on each of the states of the Markov source are quite different. By assigning different sets of codes which are matched to the run-length distributions corresponding to each state (or group of states), higher compression efficiency can be achieved.

The block diagram of the system to be described is shown in Fig. 6.1. At the transmitter, the original scanned image is first preprocessed to remove unwanted notches, basically caused by uncertainty in the decision process of present day electronic scanners. Based on its statistical properties, the preprocessed image is then coded and transmitted. At the receiver, the received signal is first decoded and a post-processing technique applied to restore the quality of the received image prior to printing.

The pre- and post-processing techniques have been described[115]

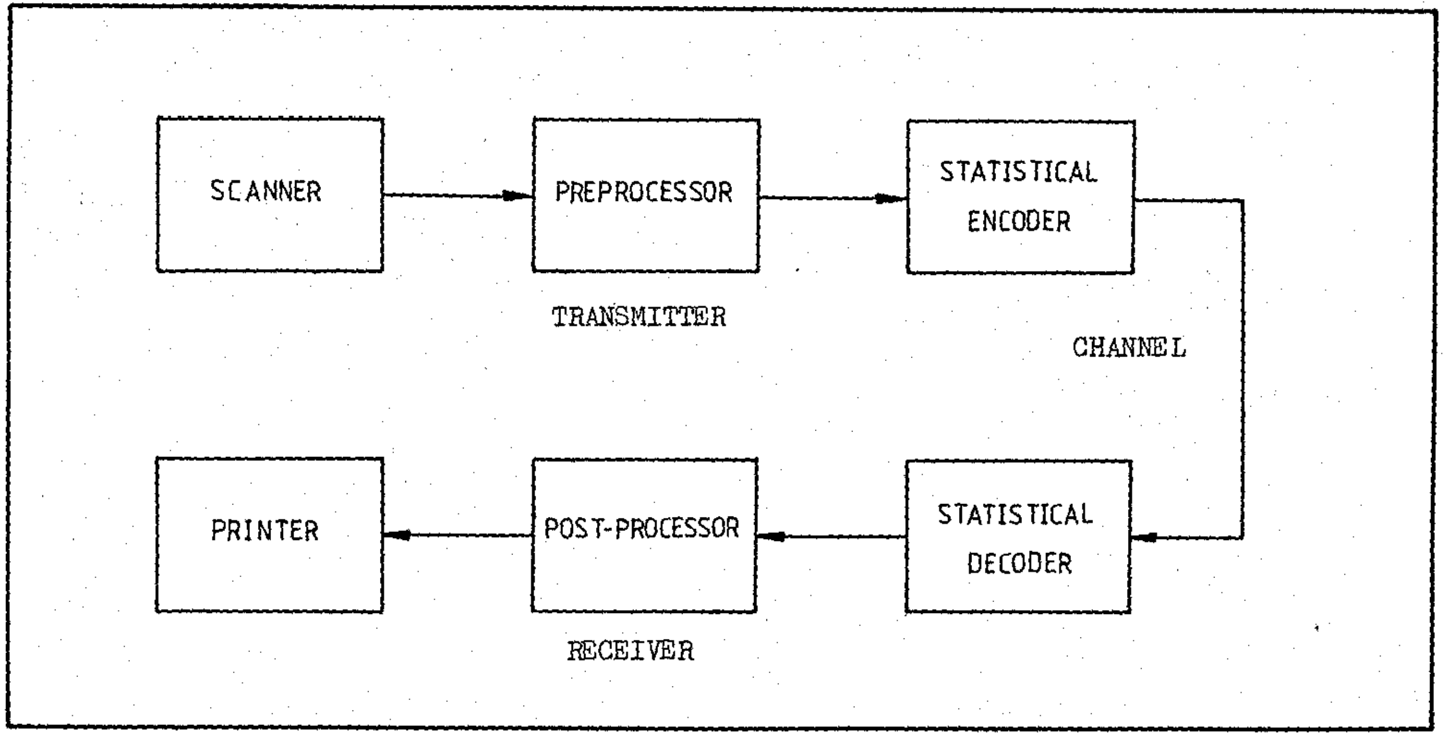


Fig. 6.1 - Block diagram of the CABC system

in Section 5.2 and it has been shown that the technique improves both image quality and coding efficiency. The subject of this chapter is the coding strategy, which forms a logical extension of the preprocessing technique.

The preprocessed image still contains a significant amount of redundancy and, assuming that the image is the output of a Markov source, it may be reduced by employing the 7th-order Markov model predictor.(60) This uses seven previous pels, both in the same and previous lines, to predict the value of the present pel, which is assumed to be statistically dependent on the values of those seven pels. The coding strategy, called Classified Adaptive Block/ Run-Length Coding (CABC), assumes the prediction errors generated by the model to be memoryless, i.e. that they are statistically independent. Basically this scheme classifies each scan-line containing prediction errors into two states called the Greater Error State and the Lesser Error State. These states are then divided adaptively into smaller blocks of size M, where M is found using, as a criterion, the probability of correct prediction. The location of the prediction errors within the blocks are coded and later transmitted. CABC can be regarded as an improvement of the Classified Pel Pattern Method.(65-66,115)

In Section 6.2, the theory of Markov information sources will be described briefly with the intention of providing some background knowledge on information theory applicable to this chapter. Further details are to be found in Abramson(116), Thomas(117), Gallager(118) and Hamming(119). The data compression technique employed and the entropy measurements used are explained in Section 6.3. The coding strategy and the determination of an optimal block size are both outlined in Section 6.4. The CABC scheme has been evaluated by means of computer simulations using five CCITT SG. XIV test documents. Entropies were calculated for both original and preprocessed images to estimate the minimum theoretical bit-rates. The actual bit-rates using the coding scheme were measured and compared with those for run-length coding using the Modified Huffman Code(42) and the Classified Pel Pattern Method(65). These results are shown in Section 6.5. In Section 6.6, the effects of introducing channel errors into the system is studied. The inclusion of pictorial illustrations provides an indication of the system susceptibility to channel errors. Conclusions are appended in Section 6.7.

6.2. THEORY OF MARKOV INFORMATION SOURCES

The principal problem in many communication systems is the transmission of information in the form of messages or data from some originating information source S to some destination D via a communication channel C. Let us assume that the source is emitting a sequence of symbols from the source alphabet denoted by the set $\{X\}$ with symbols $x_1, x_2, \ldots\ldots x_q$ where q is the size of the alphabet. The notation $p(x_i)$, i=1, 2, ... q, will be used for the probability of occurrence of the ith symbol x_i. In general, the set of numbers $\{p(x_i)\}$ may be arbitrarily assigned as long as

$$p(x_i) \geqslant 0 \quad , \quad i = 1, 2, \ldots\ldots q \qquad (6.1)$$

and

$$\sum_{i=1}^{q} p(x_i) = 1 \qquad (6.2)$$

The simplest kind of source is the one where symbols successively emitted are statistically independent. Such an information source is termed a <u>zero memory source</u> completely defined by the source alphabet $\{X\}$ and the probabilities $p(x_i)$ with which the symbols occur. If the successive symbols obey the same fixed probability law so that the one distribution $p(x_i)$ determines the appearance of each symbol, then the source is called <u>stationary</u>.

If the ith symbol x_i occurs, the amount of information for this particular symbol is equal to:-

$$I(x_i) = \log_2 \frac{1}{p(x_i)} = -\log_2 p(x_i) \quad \text{bits} \qquad (6.3)$$

where $I(x_i)$ is defined as the <u>self-information</u> of x_i.

The average amount of information obtained per symbol from the source is given by:-

$$\begin{aligned} H(X) &= \sum_{i=1}^{q} p(x_i)\, I(x_i) \quad \text{bits} \\ &= -\sum_{i=1}^{q} p(x_i) \log_2 p(x_i) \quad \text{bits} \qquad (6.4) \end{aligned}$$

The quantity denoted by $H(X)$ is called the entropy of the zero memory source. If a zero memory binary source is considered, where the source alphabet is just $\{0,1\}$, then the entropy is given by:-

$$H(X) = - p \log_2 p - (1-p) \log_2 (1-p) \qquad (6.5)$$

where p is the probability of a '0'.

The zero memory source considered so far is too restrictive for many applications. A constructive way to generalise this model is to assume that the occurrence of a given symbol depends on a finite number m of immediately preceding symbols. Such an information source is called an mth order Markov source.

For such a source, the m symbols preceding a given symbol x_i will be denoted as

$$x_{i_1} \; x_{i_2} \; x_{i_3} \; x_{i_4} \; \ldots\ldots \; x_{i_m} \qquad (6.6)$$

At a given time, the source will be in a given state. If there are q possible symbols x_i, then there are q^m possible states. For each state there exists a set of conditional probabilities $p(x_i/x_{i_1}, x_{i_2}, \ldots x_{i_m})$ that x_i will occur given that the letters $x_{i_1}, x_{i_2}, \ldots\ldots x_{i_m}$ have occurred. For most purposes, only Markov sources which are ergodic are of interest. An ergodic source is a source which, if observed for a very long period of time, will emit a sequence of symbols which is 'typical'. In other words, if we select an initial state of a Markov source (according to some set of initial probabilities over the state) and wait a large number of state transitions, the states will settle to some definite probability, independent of the initial state.

The self-information of x_i given that x_{i_1}, x_{i_2}, ... x_{i_m} have occurred is given as:-

$$I(x_i/x_{i_1}, x_{i_2}, \ldots\ldots x_{i_m}) = -\log_2 \left[p(x_i/x_{i_1}, x_{i_2}, \ldots x_{i_m}) \right] \qquad (6.7)$$

The conditional entropy over the alphabet X of symbols x_i is naturally given by the expression:-

$$H(X/x_{i_1}, x_{i_2}, \ldots\ldots x_{i_m}) = -\sum_{X} p(x_i/x_{i_1}, x_{i_2}, \ldots\ldots x_{i_m}) \log_2 \left[p(x_i/x_{i_1}, x_{i_2}, \ldots\ldots x_{i_m}) \right] \qquad (6.8)$$

This is the conditional probability of the source alphabet X given that we have the sequence of symbols x_{i_1}, x_{i_2}, ... x_{i_m}.

We next consider the larger system and allow for the individual probabilities of the states of the Markov process. Let $p(x_{i_1}, x_{i_2}, \ldots\ldots x_{i_m})$ be the probability of being in state $x_{i_1}, x_{i_2}, \ldots\ldots x_{i_m}$. Then it is natural to define the entropy of the Markov system as the probabilities of being in a given state times the conditional entropies of the state by the expression:-

$$H(X) = \sum_{X^m} p(x_{i_1}; x_{i_2}, \ldots x_{i_m})\, H(X/x_{i_1}, x_{i_2}, \ldots x_{i_m}) \qquad (6.9)$$

From Eqn. (6.8), we get:-

$$H(X) = -\sum_{X^m} \sum_{X} p(x_{i_1}, x_{i_2}, \ldots x_{i_m})\, p(x_i / x_{i_1}, x_{i_2}, \ldots x_{i_m}) \log_2 \left[p(x_i / x_{i_1}, x_{i_2}, \ldots x_{i_m}) \right] \quad (6.10)$$

But since

$$p(x_{i_1}, x_{i_2}, \ldots x_{i_m})\, p(x_i / x_{i_1}, x_{i_2}, \ldots x_{i_m}) = p(x_i, x_{i_1}, \ldots x_{i_m}) \quad (6.11)$$

then the entropy of a Markov source is:-

$$H(X) = -\sum_{X^{m+1}} p(x_i, x_{i_1}, x_{i_2}, \ldots x_{i_m}) \log_2 p(x_i / x_{i_1}, x_{i_2}, \ldots x_{i_m}) \quad (6.12)$$

6.3. DATA COMPRESSION AND ENTROPY MEASUREMENTS

The preprocessing technique described in Section 5.2 basically removes redundant information such as notches, pinholes, and random isolated black or white pels present in the original image. To supplement the preprocessing technique, a 7th-order Markov model predictor[60] is used as a means of further redundancy reduction by exploiting the fact that most facsimile images exhibit high line-to-line correlation. The use of the 7th-order Markov model predictor is clearly based on the assumption that facsimile signals can be regarded as the output of a 7th-order Markov source. The arrangement of the reference pels is shown in Fig. 6.2.

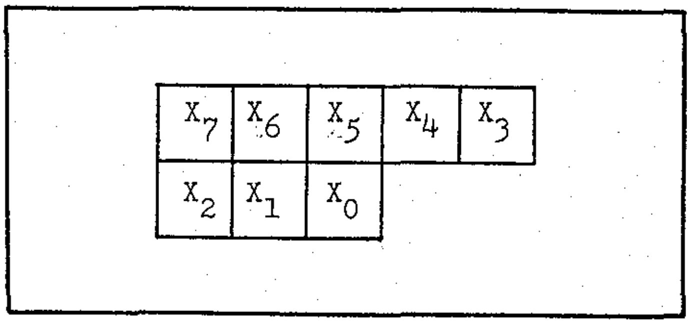


Fig. 6.2 - Reference pels for the 7th-order Markov model predictor.

The transmitter stores these reference pels and from them predicts the value of the present pel X_0 (X_0 being assumed statistically dependent on the previously scanned pels). Denoting the predicted value as $\hat{X}_0$, the present pel is then replaced by the prediction error $E_0 = X_0 \oplus \hat{X}_0$, where the symbol $\oplus$ means modulo-2 addition. This conversion is carried out on a pel-to-pel and line-by-line basis. Thus, the original sequence $\{X_0\}$ is converted into an error sequence $\{E_0\}$. In other words, the resulting picture of the prediction errors is a one-to-one transformation of the original. The prediction errors generated are then coded and later transmitted.

Since the facsimile signal is regarded as deriving from a 7th-order Markov source, it can be expressed by the conditional probability given by:-

$$P(X_0 / X_1, X_2, \ldots\ldots\ldots X_7) \qquad (6.13)$$

and the [redacted] entropy of the Markov source is then given by:-

$$H_{pel} = -\sum P(X_0,X_1,X_2, \ldots X_7) \log_2 P(X_0 / X_1,X_2 \ldots X_7) \quad (6.14)$$

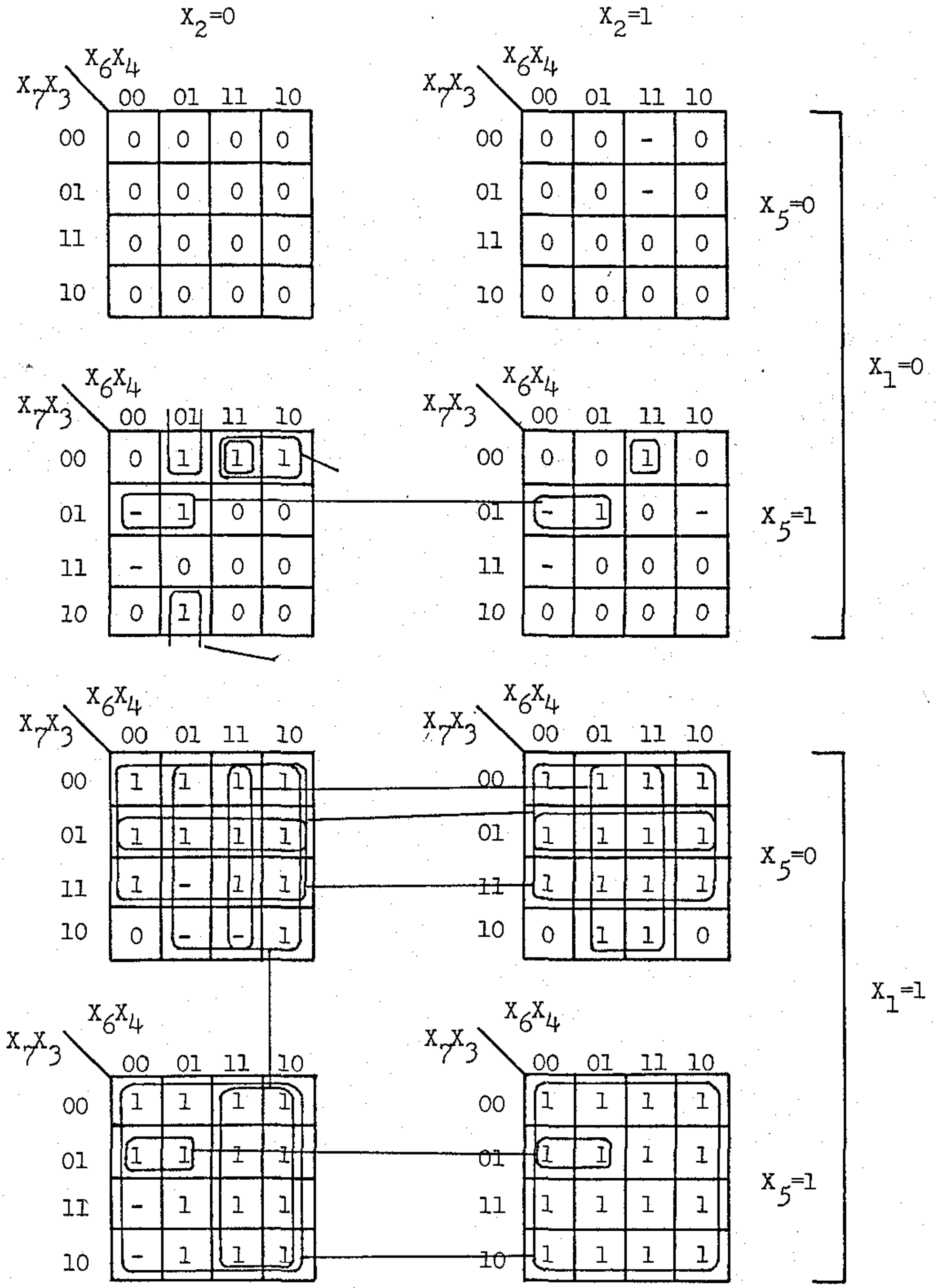
Based on the conditional probabilities $P(X_0=0/S_j)$ and $P(X_0=1/S_j)$, where S_j represents all the 2^7 possible states of the reference pels, a maximum likelihood prediction function can be derived. The optimum prediction function is one which minimises the probability of making an error, given that a particular state has occurred. The explicit expression for the optimum prediction function can be easily derived with the aid of a Karnaugh map, if the conditional probabilities expressed by Eqn. (6.13) are known. The conditional probabilities $P(X_0=0/S_j)$ and $P(X_0=1/S_j)$ and the predicted values $\hat{X}_0$, which were obtained based on the maximum likelihood condition for each source state, for the original and preprocessed CCITT document No. 1 are shown in Table 6.1(a) and Table 6.1(b) respectively. Since there are seven reference pels, a seven-variable Karnaugh map is constructed as shown in Fig. 6.3(a) and Fig. 6.3(b) for the original and preprocessed documents. For each source state, the Karnaugh map is filled with the corresponding predicted values, $\hat{X}_0$. The dash is considered as a 'don't care' state, and, by combining the '1s', a Boolean logical expression indicating the optimum prediction function is obtained. The prediction function obtained from the map will satisfy all the different possible states present. Since we are only concerned with binary images, the derivation of the prediction function using the Karnaugh map method not only economises on the

						$X_1=0$			$X_1=1$		
X_2	X_5	X_6	X_4	X_7	X_3	$P(X_0=0/S)$	$P(X_0=1/S)$	$\hat{X}_0$	$P(X_0=0/S)$	$P(X_0=1/S)$	$\hat{X}_0$
0	0	0	0	0	0	0.9995	0.0005	0	0.1374	0.8626	1
0	0	0	0	0	1	0.9653	0.0347	0	0.0231	0.9769	1
0	0	0	0	1	0	0.9984	0.0016	0	0.5192	0.4808	0
0	0	0	0	1	1	0.9816	0.0184	0	0.0000	1.0000	1
0	0	0	1	0	0	0.6872	0.3128	0	0.0000	1.0000	1
0	0	0	1	0	1	0.8006	0.1994	0	0.0045	0.9955	1
0	0	0	1	1	0	0.8125	0.1875	0	0.5000	0.5000	-
0	0	0	1	1	1	0.9416	0.0584	0	0.5000	0.5000	-
0	0	1	0	0	0	0.8657	0.1343	0	0.2500	0.7500	1
0	0	1	0	0	1	1.0000	0.0000	0	0.2857	0.7143	1
0	0	1	0	1	0	0.9892	0.0108	0	0.1056	0.8944	1
0	0	1	0	1	1	0.9861	0.0139	0	0.2105	0.7895	1
0	0	1	1	0	0	1.0000	0.0000	0	0.3333	0.6667	1
0	0	1	1	0	1	0.8889	0.1111	0	0.0000	1.0000	1
0	0	1	1	1	0	1.0000	0.0000	0	0.5000	0.5000	-
0	0	1	1	1	1	0.9907	0.0093	0	0.2857	0.7143	1
0	1	0	0	0	0	0.5438	0.4562	0	0.0578	0.9422	1
0	1	0	0	0	1	0.5000	0.5000	-	0.0000	1.0000	1
0	1	0	0	1	0	1.0000	0.0000	0	0.5000	0.5000	-
0	1	0	0	1	1	0.5000	0.5000	-	0.5000	0.5000	-
0	1	0	1	0	0	0.2097	0.7913	1	0.0006	0.9994	1
0	1	0	1	0	1	0.4511	0.5489	1	0.0008	0.9992	1
0	1	0	1	1	0	0.3684	0.6316	1	0.0000	1.0000	1
0	1	0	1	1	1	0.9281	0.0719	0	0.0000	1.0000	1
0	1	1	0	0	0	0.3106	0.6984	1	0.0262	0.9738	1
0	1	1	0	0	1	0.8125	0.1875	0	0.1429	0.8571	1
0	1	1	0	1	0	0.8604	0.1396	0	0.0284	0.9716	1
0	1	1	0	1	1	0.9444	0.5556	0	0.0000	1.0000	1
0	1	1	1	0	0	0.1528	0.8472	1	0.0023	0.9977	1
0	1	1	1	0	1	0.5946	0.4054	0	0.0026	0.9974	1
0	1	1	1	1	0	0.7313	0.2687	0	0.0158	0.9842	1
0	1	1	1	1	1	0.8136	0.1864	0	0.0309	0.9691	1
1	0	0	0	0	0	0.9902	0.0098	0	0.3263	0.6737	1
1	0	0	0	0	1	0.8967	0.1033	0	0.0601	0.9399	1
1	0	0	0	1	0	0.9993	0.0007	0	0.7047	0.2953	0
1	0	0	0	1	1	0.9872	0.0128	0	0.3362	0.6638	1
1	0	0	1	0	0	0.7143	0.2857	0	0.0244	0.9756	1
1	0	0	1	0	1	0.7143	0.2857	0	0.0126	0.9874	1
1	0	0	1	1	0	1.0000	0.0000	0	0.4000	0.6000	1
1	0	0	1	1	1	0.8647	0.1353	0	0.2038	0.7962	1
1	0	1	0	0	0	1.0000	0.0000	0	0.4149	0.5861	1
1	0	1	0	0	1	1.0000	0.0000	0	0.4000	0.6000	1
1	0	1	0	1	0	0.9994	0.0006	0	0.6698	0.3302	0
1	0	1	0	1	1	1.0000	0.0000	0	0.4828	0.5172	1
1	0	1	1	0	0	0.5000	0.5000	-	0.0000	1.0000	1
1	0	1	1	0	1	0.5000	0.5000	-	0.0000	1.0000	1
1	0	1	1	1	0	1.0000	0.0000	0	0.0714	0.9286	1
1	0	1	1	1	1	0.9773	0.0227	0	0.2026	0.7974	1
1	1	0	0	0	0	0.6000	0.4000	0	0.0389	0.9611	1
1	1	0	0	0	1	0.5000	0.5000	-	0.0000	1.0000	1
1	1	0	0	1	0	1.0000	0.0000	0	0.0667	0.9333	1
1	1	0	0	1	1	0.5000	0.5000	-	0.0000	1.0000	1
1	1	0	1	0	0	0.5714	0.4286	0	0.0161	0.9839	1
1	1	0	1	0	1	0.3500	0.6500	1	0.0068	0.9932	1
1	1	0	1	1	0	0.6000	0.4000	0	0.0500	0.9500	1
1	1	0	1	1	1	0.7826	0.2174	0	0.0161	0.9839	1
1	1	1	0	0	0	0.7143	0.2857	0	0.1884	0.8116	1
1	1	1	0	0	1	0.5000	0.5000	-	0.2143	0.7857	1
1	1	1	0	1	0	0.9784	0.0216	0	0.3229	0.6771	1
1	1	1	0	1	1	1.0000	0.0000	0	0.2671	0.7329	1
1	1	1	1	0	0	0.3000	0.7000	1	0.0259	0.9741	1
1	1	1	1	0	1	1.0000	0.0000	0	0.0330	0.9670	1
1	1	1	1	1	0	0.8603	0.1397	0	0.1186	0.8814	1
1	1	1	1	1	1	0.9010	0.0990	0	0.0588	0.9412	1

Table 6.1(a)

						$X_1=0$			$X_1=1$		
X_2	X_5	X_6	X_4	X_7	X_3	$P(X_0=0/S)$	$P(X_0=1/S)$	$\hat{X}_0$	$P(X_0=0/S)$	$P(X_0=1/S)$	$\hat{X}_0$
0	0	0	0	0	0	0.9994	0.0006	0	0.0138	0.9862	1
0	0	0	0	0	1	0.9561	0.0409	0	0.0072	0.9928	1
0	0	0	0	1	0	0.9968	0.0032	0	0.4423	0.5577	1
0	0	0	0	1	1	0.9781	0.0219	0	0.2500	0.7500	1
0	0	0	1	0	0	0.8847	0.1153	0	0.0000	1.0000	1
0	0	0	1	0	1	0.9623	0.0377	0	0.0000	1.0000	1
0	0	0	1	1	0	0.5000	0.5000	-	0.5000	0.5000	-
0	0	0	1	1	1	1.0000	0.0000	0	0.0000	1.0000	1
0	0	1	0	0	0	0.9386	0.0614	0	0.4727	0.5273	1
0	0	1	0	0	1	1.0000	0.0000	0	0.2777	0.7223	1
0	0	1	0	1	0	0.9902	0.0098	0	0.2784	0.7216	1
0	0	1	0	1	1	0.9929	0.0071	0	0.4839	0.5161	1
0	0	1	1	0	0	0.5000	0.5000	-	0.5000	0.5000	-
0	0	1	1	0	1	0.5000	0.5000	-	0.5000	0.5000	-
0	0	1	1	1	0	0.5000	0.5000	-	0.5000	0.5000	-
0	0	1	1	1	1	0.9524	0.0476	0	0.0000	1.0000	1
0	1	0	0	0	0	0.1951	0.8049	1	0.1957	0.8043	1
0	1	0	0	0	1	0.5000	0.5000	-	0.5000	0.5000	-
0	1	0	0	1	0	0.5000	0.5000	-	0.5000	0.5000	-
0	1	0	0	1	1	0.5000	0.5000	-	0.5000	0.5000	-
0	1	0	1	0	0	0.0826	0.9174	1	0.0070	0.9930	1
0	1	0	1	0	1	0.2399	0.7601	1	0.0024	0.9976	1
0	1	0	1	1	0	0.3333	0.6667	1	0.0000	1.0000	1
0	1	0	1	1	1	0.7143	0.2857	0	0.0000	1.0000	1
0	1	1	0	0	0	0.5579	0.4421	0	0.0050	0.9950	1
0	1	1	0	0	1	0.6667	0.3333	0	0.0000	1.0000	1
0	1	1	0	1	0	0.8219	0.1781	0	0.0025	0.9975	1
0	1	1	0	1	1	0.9678	0.0322	0	0.0000	1.0000	1
0	1	1	1	0	0	0.4647	0.5353	1	0.0088	0.9912	1
0	1	1	1	0	1	0.8889	0.1111	0	0.0097	0.9903	1
0	1	1	1	1	0	0.7861	0.2139	0	0.0000	1.0000	1
0	1	1	1	1	1	0.8425	0.1575	0	0.0027	0.9973	1
1	0	0	0	0	0	0.9956	0.0044	0	0.2756	0.7244	1
1	0	0	0	0	1	0.9259	0.0741	0	0.0575	0.9425	1
1	0	0	0	1	0	0.9994	0.0006	0	0.3904	0.6096	1
1	0	0	0	1	1	0.9871	0.0129	0	0.1790	0.8210	1
1	0	0	1	0	0	0.5000	0.5000	-	0.0208	0.9792	1
1	0	0	1	0	1	0.9889	0.0111	0	0.0084	0.9916	1
1	0	0	1	1	0	1.0000	0.0000	0	0.2500	0.7500	1
1	0	0	1	1	1	0.9960	0.0040	0	0.0723	0.9277	1
1	0	1	0	0	0	1.0000	0.0000	0	0.9588	0.0412	0
1	0	1	0	0	1	1.0000	0.0000	0	0.7500	0.2500	0
1	0	1	0	1	0	0.9986	0.0014	0	0.8563	0.1437	0
1	0	1	0	1	1	1.0000	0.0000	0	0.6342	0.3658	0
1	0	1	1	0	0	0.5000	0.5000	-	0.5000	0.5000	-
1	0	1	1	0	1	0.5000	0.5000	-	0.5000	0.5000	-
1	0	1	1	1	0	0.5000	0.5000	-	0.0000	1.0000	1
1	0	1	1	1	1	1.0000	0.0000	0	0.7231	0.2769	0
1	1	0	0	0	0	1.0000	0.0000	0	0.0435	0.9565	1
1	1	0	0	0	1	0.5000	0.5000	-	0.5000	0.5000	-
1	1	0	0	1	0	0.5000	0.5000	-	0.3333	0.6667	1
1	1	0	0	1	1	0.5000	0.5000	-	0.5000	0.5000	-
1	1	0	1	0	0	0.3333	0.6667	1	0.0091	0.9909	1
1	1	0	1	0	1	0.2353	0.7647	1	0.0064	0.9936	1
1	1	0	1	1	0	0.3000	0.7000	1	0.2000	0.8000	1
1	1	0	1	1	1	0.1316	0.8684	1	0.1875	0.8125	1
1	1	1	0	0	0	0.7778	0.2222	0	0.0445	0.9555	1
1	1	1	0	0	1	0.5000	0.5000	-	0.0000	1.0000	1
1	1	1	0	1	0	0.9965	0.0035	0	0.0619	0.9381	1
1	1	1	0	1	1	1.0000	0.0000	0	0.0656	0.9381	1
1	1	1	1	0	0	0.5000	0.5000	-	0.1357	0.8643	1
1	1	1	1	0	1	0.8334	0.1666	0	0.1111	0.8889	1
1	1	1	1	1	0	0.9874	0.0126	0	0.1280	0.8720	1
1	1	1	1	1	1	0.9979	0.0021	0	0.0582	0.9418	1

Table 6.1(b)



X_7X_3 \ X_6X_4	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	0	0	0	0
10	0	0	0	0

X_7X_3 \ X_6X_4	00	01	11	10
00	0	0	-	0
01	0	0	-	0
11	0	0	0	0
10	0	0	0	0

X_7X_3 \ X_6X_4	00	01	11	10
00	0	1	1	1
01	-	1	0	0
11	-	0	0	0
10	0	1	0	0

X_7X_3 \ X_6X_4	00	01	11	10
00	0	0	1	0
01	-	1	0	-
11	-	0	0	0
10	0	0	0	0

X_7X_3 \ X_6X_4	00	01	11	10
00	1	1	1	1
01	1	1	1	1
11	1	-	1	1
10	0	-	-	1

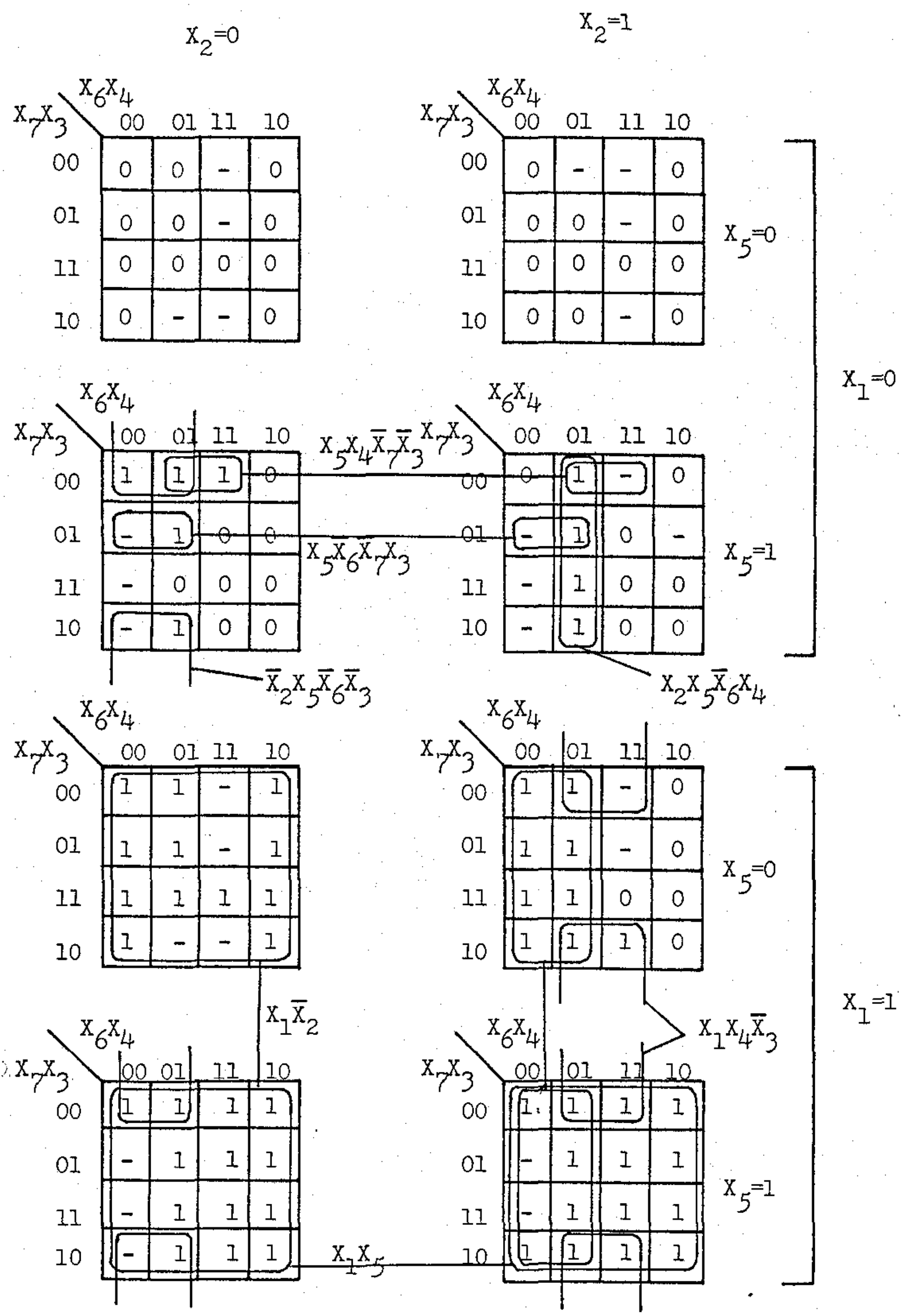
X_7X_3 \ X_6X_4	00	01	11	10
00	1	1	1	1
01	1	1	1	1
11	1	1	1	1
10	0	1	1	0

X_7X_3 \ X_6X_4	00	01	11	10
00	1	1	1	1
01	1	1	1	1
11	-	1	1	1
10	-	1	1	1

X_7X_3 \ X_6X_4	00	01	11	10
00	1	1	1	1
01	1	1	1	1
11	1	1	1	1
10	1	1	1	1

$$\hat{X}_0 = X_1(X_5+X_4+\bar{X}_7+X_3)+X_1\bar{X}_2X_6+X_5X_6\bar{X}_7\bar{X}_3(X_4+\bar{X}_2)+X_5\bar{X}_6(\bar{X}_7X_3+\bar{X}_2X_4\bar{X}_3)$$

Fig. 6.3(a) - Karnaugh map for original image



$X_1=0$, $X_5=0$, $X_2=0$

X_7X_3 \ X_6X_4	00	01	11	10
00	0	0	-	0
01	0	0	-	0
11	0	0	0	0
10	0	-	-	0

$X_1=0$, $X_5=0$, $X_2=1$

X_7X_3 \ X_6X_4	00	01	11	10
00	0	-	-	0
01	0	0	-	0
11	0	0	0	0
10	0	0	-	0

$X_1=0$, $X_5=1$, $X_2=0$

X_7X_3 \ X_6X_4	00	01	11	10
00	1	1	1	0
01	-	1	0	0
11	-	0	0	0
10	-	1	0	0

$X_1=0$, $X_5=1$, $X_2=1$

X_7X_3 \ X_6X_4	00	01	11	10
00	0	1	-	0
01	-	1	0	-
11	-	1	0	0
10	-	1	0	0

$X_1=1$, $X_5=0$, $X_2=0$

X_7X_3 \ X_6X_4	00	01	11	10
00	1	1	-	1
01	1	1	-	1
11	1	1	1	1
10	1	-	-	1

$X_1=1$, $X_5=0$, $X_2=1$

X_7X_3 \ X_6X_4	00	01	11	10
00	1	1	-	0
01	1	1	-	0
11	1	1	0	0
10	1	1	1	0

$X_1=1$, $X_5=1$, $X_2=0$

X_7X_3 \ X_6X_4	00	01	11	10
00	1	1	1	1
01	-	1	1	1
11	-	1	1	1
10	-	1	1	1

$X_1=1$, $X_5=1$, $X_2=1$

X_7X_3 \ X_6X_4	00	01	11	10
00	1	1	1	1
01	-	1	1	1
11	-	1	1	1
10	1	1	1	1

$$X_0 = X_1(X_5+\bar{X}_2+\bar{X}_6)+X_5\bar{X}_6(X_2X_4+\bar{X}_2\bar{X}_3)+X_5\bar{X}_7(X_4\bar{X}_3+\bar{X}_6X_3)+X_1X_4\bar{X}_3$$

Fig. 6.3(b) - Karnaugh map for preprocessed image amount of memory needed to store all the 128 possible states but also reduces the complexity of implementation.

Due to the different statistical properties of the original and preprocessed CCITT documents (digitised with 1728 pels by 2376 lines, corresponding to a resolution of 8 pels/mm horizontally and vertically), the prediction function is optimised for each document as shown in Table 6.2. These prediction functions are easily implementable using simple logic gates. (The use of a common prediction function for all the documents is estimated to increase the entropy per pel by a factor of about four per-cent(66).)

Fig. 6.4(a) and Fig. 6.4(b) show, respectively, parts of documents Nos. 1 and 2 with their corresponding prediction errors in Figs. 6.4(c) and 6.4(d). Notice there is a one-to-one transformation and that the prediction errors appear to be less correlated. From this lack of correlation, it is justifiable to assume that the prediction errors generated by the 7th-order Markov model predictor could approximate a memoryless binary information source (zero-memory binary source) and this property will be used later in the coding strategy. The entropy of such a source, given by:-

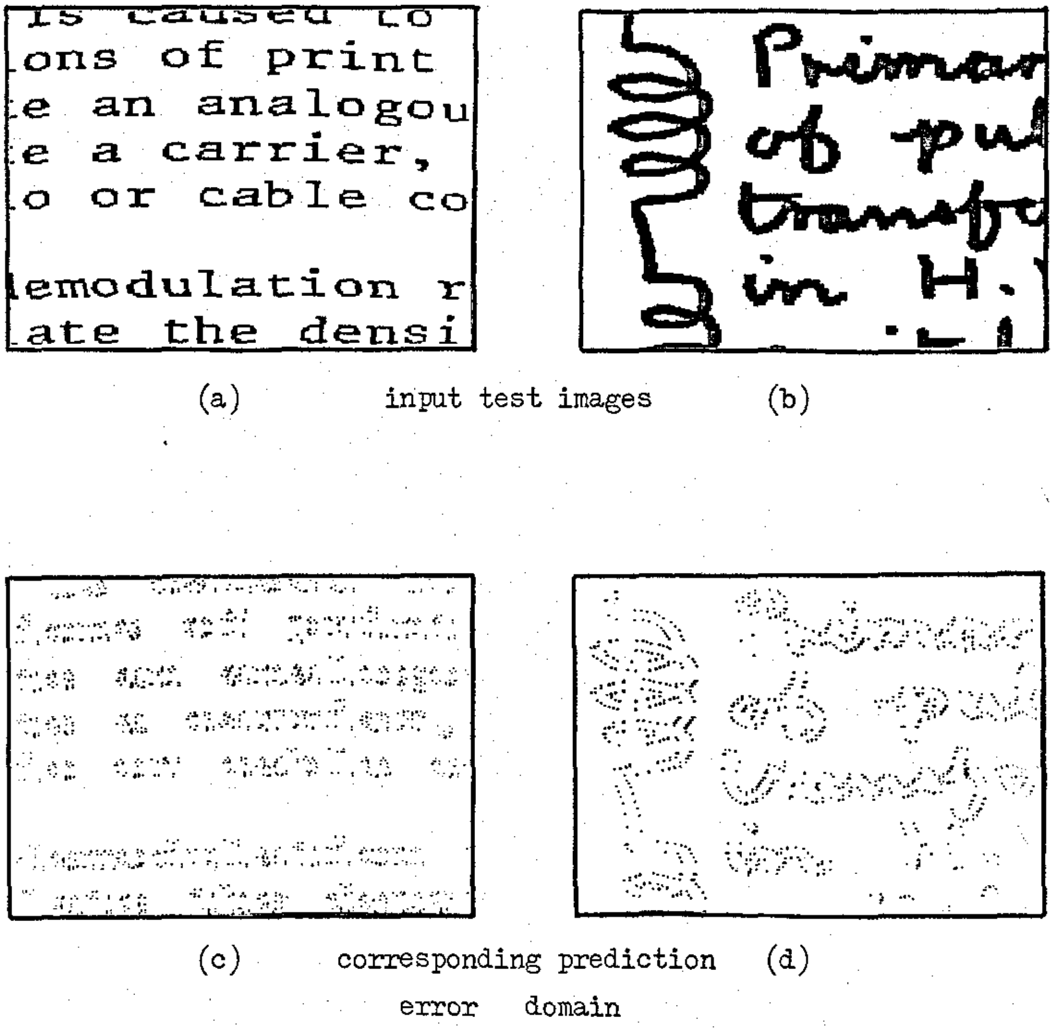
$$H = - p \log_2 p - (1-p) \log_2 (1-p) \qquad (6.15)$$

is used as a measure of performance, where p is the probability of correct prediction.

Since the statistics of each source state of the Markov source are

	ORIGINAL	PREPROCESSED
DOCUMENT NO. 1	$\hat{X}_0 = X_1(X_5+X_4+\bar{X}_7+X_3)+X_1\bar{X}_2X_6+$ $X_5X_6\bar{X}_7\bar{X}_3(X_4+\bar{X}_2)+X_5\bar{X}_6(\bar{X}_7X_3+\bar{X}_2X_4\bar{X}_3)$	$\hat{X}_0 = X_1(X_5+\bar{X}_2+\bar{X}_6)+X_5\bar{X}_6(X_2X_4+\bar{X}_2\bar{X}_3)+$ $X_5\bar{X}_7(X_4\bar{X}+\bar{X}_6X_3)+X_1X_4\bar{X}_3$
DOCUMENT NO. 2	$\hat{X}_0 = X_1X_5+X_1X_2(\bar{X}_7+X_3+X_4)+X_1\bar{X}_7(X_3+\bar{X}_6)+$ $X_2X_5\bar{X}_3(X_6\bar{X}_7+\bar{X}_6X_4)+X_4\bar{X}_7(\bar{X}_2X_5X_3+X_2X_6)+$ $X_1\bar{X}_2X_6X_7\bar{X}_3$	$\hat{X}_0 = X_1X_5+X_1\bar{X}_6(X_2+\bar{X}_7)+X_2X_4\bar{X}_7(X_6+X_5\bar{X}_3)+$ $X_5\bar{X}_6(X_7+\bar{X}_2X_4X_3)+X_1\bar{X}_2X_7\bar{X}_3+$ $\bar{X}_1\bar{X}_5X_6X_4\bar{X}_7$
DOCUMENT NO. 4	$\hat{X}_0 = X_1(\bar{X}_2+X_5+X_4+\bar{X}_7+X_3)+X_4\bar{X}_7\bar{X}_3(\bar{X}_2X_5+X_2\bar{X}_5)+$ $X_5\bar{X}_7(\bar{X}_6X_4+X_6\bar{X}_4X_3)$	$\hat{X}_0 = X_1(\bar{X}_2+X_5)+\bar{X}_6(X_1+X_5)$
DOCUMENT NO. 5	$\hat{X}_0 = X_1(\bar{X}_2+X_5+X_4+\bar{X}_7)+X_1\bar{X}_6X_3+X_5\bar{X}_7X_3(\bar{X}_2+X_4)+$ $X_5\bar{X}_6(\bar{X}_2X_4\bar{X}_7+X_2X_7\bar{X}_3)$	$\hat{X}_0 = X_1X_5+X_1\bar{X}_2(\bar{X}_6+X_4+X_7+X_3)+X_1\bar{X}_6(X_3+\bar{X}_7+X_3)+$ $\bar{X}_2X_5\bar{X}_6(\bar{X}_3+X_4)+(X_5\bar{X}_6X_7)$
DOCUMENT NO. 7	$\hat{X}_0 = X_1X_5(\bar{X}_2+X_4+\bar{X}_3)+X_1X_4(\bar{X}_2+\bar{X}_6)+X_1\bar{X}_6(\bar{X}_5\bar{X}_7+$ $X_5X_7)+\bar{X}_7\bar{X}_3(X_1X_4+\bar{X}_2X_5)+X_1\bar{X}_2X_6(X_3+X_7)+$ $X_5\bar{X}_6X_4X_7X_3+X_1X_2X_5\bar{X}_6\bar{X}_4\bar{X}_7X_3$	$\hat{X}_0 = X_1X_5(X_6+X_7+\bar{X}_3)+X_1\bar{X}_6X_4+X_1\bar{X}_5\bar{X}_6+X_1X_4(\bar{X}_7\bar{X}_3+$ $\bar{X}_2X_7)+\bar{X}_6X_7(X_5\bar{X}_3+X_1X_2)+X_5\bar{X}_6\bar{X}_7(\bar{X}_1\bar{X}_2+$ $X_4\bar{X}_3)+\bar{X}_1\bar{X}_5X_6\bar{X}_3(\bar{X}_2\bar{X}_7+X_2X_4X_7)$

Table 6.2 - Prediction functions for original and preprocessed CCITT documents



(a) input test images (b)

(c) corresponding prediction error domain (d)

Fig. 6.4 - Input test images and their corresponding prediction error domains

different, higher compression efficiency can be achieved by coding the run-lengths corresponding to each state or groups of states independently using a different code set. Preuss(60) developed this technique by exploiting the two-dimensional redundancy in the signal to the fullest extent. Netravali et. al.(63) who also used the mth-order Markov model predictor, classified the prediction errors into two states called the "good" and "bad" state according to some "goodness" threshold in order to increase the average run-lengths of black and white pels. An ordering technique is then implemented by reordering elements in a line based on previously transmitted surrounding elements. The run-lengths of such reordered lines were then coded using one or more code sets, as desired.

Ideally, each state should be coded differently as in Preuss's scheme but the choice adopted here is a simpler version similar to that proposed by Ueno et. al.(65), where the prediction domain is categorised into two states. The predicted values generated by the predictor are classified into two general states called the Lesser Error State (LES) and the Greater Error State (GES). The LES and GES so obtained produce good results if the logical expression given by:-

$$\begin{aligned} LES &= X_1X_4X_5 + \bar{X}_1\bar{X}_4\bar{X}_5 \\ GES &= \overline{LES} \end{aligned} \qquad (6.16)$$

are satisfied.

An example to illustrate how the prediction domain is classified into IES and GES is shown in Fig. 6.5. Consider two adjacent lines, L and L-1, of a typical facsimile image as shown in Fig. 6.5(a). Using the prediction function for the preprocessed document No. 1 listed in Table 6.2, the predicted values of the present pel X_0 for line L is shown in Fig. 6.5(b). The predicted values $\hat{X}_0$, are then compared with the real values X_0, and if they do not correspond, an error, denoted by a '1', is inserted for X_0 as seen in Fig. 6.5(c). To categorise the prediction domain into the Lesser Error State and the Greater Error State, Eqns. (6.16) are applied. If elements $X_1X_4X_5$ are all black OR all white, the corresponding value of the prediction domain goes into the IES, if NOT, they go into the GES. This is clearly shown in Fig. 6.5(d) and Fig. 6.5(e). Notice that in the example used, IES has fewer prediction errors than GES. This is generally true for all scan-lines.

As a measure of performance, entropy is again used and this time is given by:-

$$\begin{aligned} H = & - P(\text{IES}) \left[p_{\text{IES}} \log_2 p_{\text{IES}} + (1-p_{\text{IES}}) \log_2 (1-p_{\text{IES}}) \right] \\ & - P(\text{GES}) \left[p_{\text{GES}} \log_2 p_{\text{GES}} + (1-p_{\text{GES}}) \log_2 (1-p_{\text{GES}}) \right] \qquad (6.17) \end{aligned}$$

where P(IES) is the probability of the prediction domain being in the Lesser Error State and p_{IES} is the probability of prediction errors in the Lesser Error State. P(GES) and p_{GES} are similarly defined. (See Appendix A for the derivation of Eqn. (6.17))

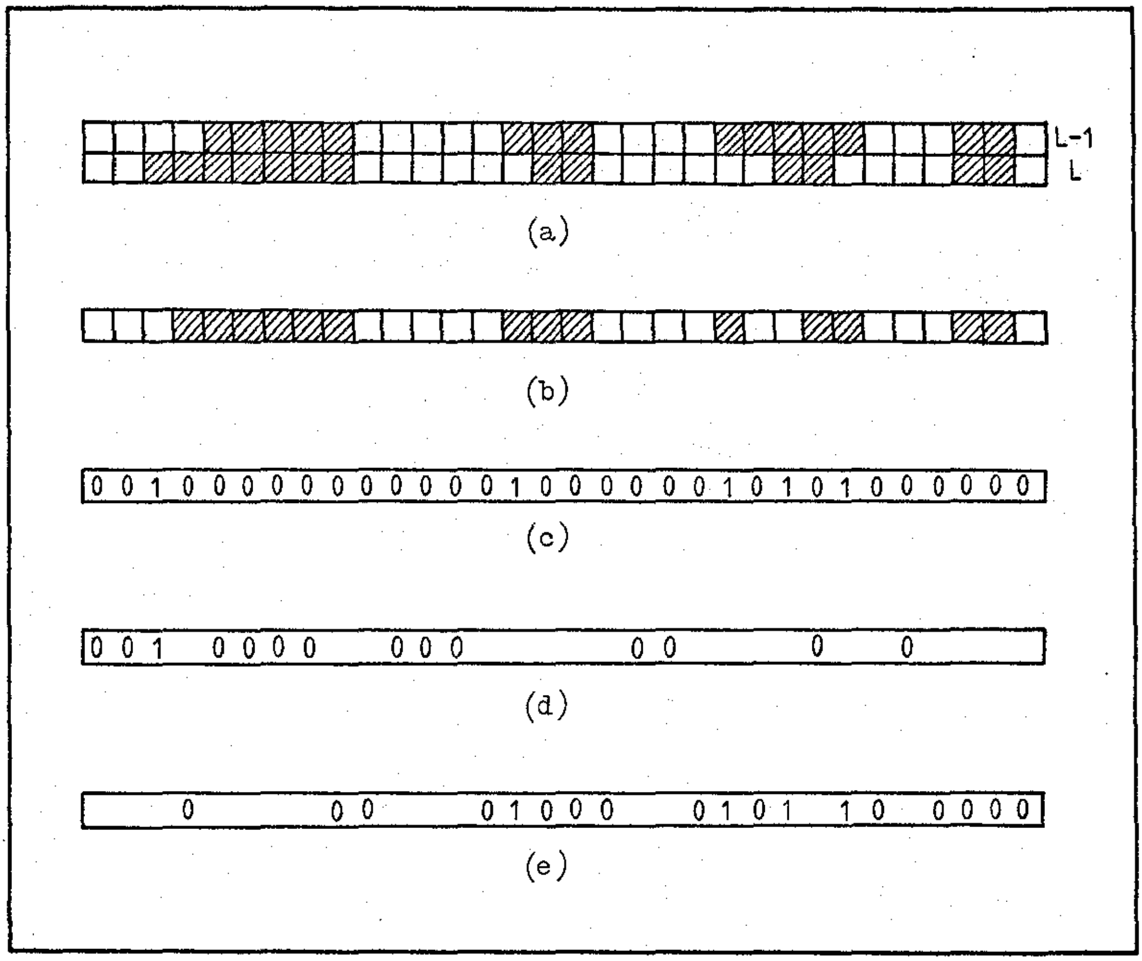


Fig. 6.5 - Lesser Error State and Greater Error State

(a) Original

(b) Predicted values

(c) Prediction errors

(d) LES

(e) GES

6.4. CODING STRATEGY

The Classified Adaptive Block/Run-Length Coding (CABC) scheme for coding the position of prediction errors in the two general states is now described. Taking each scan-line independently, the sequence of errors generated by the 7th-order Markov model predictor is assumed to constitute a memoryless binary information source. According to Shannon's Noiseless Coding Theorem(3), such a source can be encoded using on average $H_1(x)$ bits per source digit, where $H_1(x)$ is the entropy per scan-line given by:-

$$H_1(x) = p_1 \log_2 p_1 - (1-p_1) \log_2 (1-p_1) \qquad (6.18)$$

and p_1 is the probability of correct prediction on a scan-line.

Consider first the case before classification into the two states where each scan-line is made up of N pels. The measured probability distribution of the number of prediction errors per scan-line is shown in Fig. 6.6. It can be approximated by a decaying exponential characteristic, for which the occurrence of a scan-line with few or no prediction errors is most probable.

Optimal coding of scan-lines containing prediction errors, knowing their probabilities, into a sequence of bits, can be obtained using the Huffman procedure.(36) In Huffman coding, the longest sequence of bits is assigned to the least probable message while shorter sequences are used for more probable messages, thus minimising the average number of bits per source symbol. If

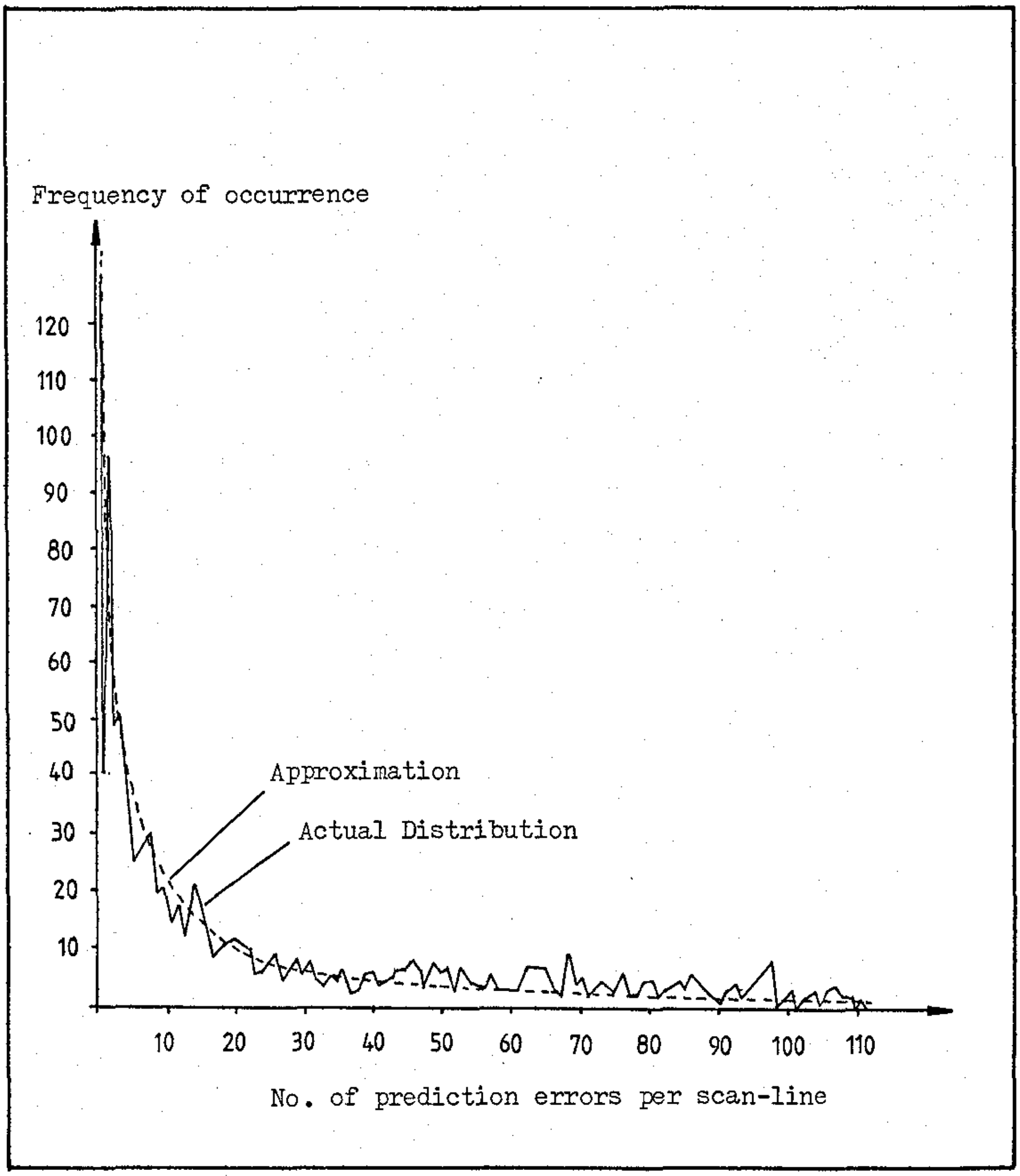


Fig. 6.6 - Distribution of prediction errors per scan-line

the probabilities of the source messages are known, the Huffman algorithm gives the code best 'matched' to the source in the sense of minimising code redundancy. But if the actual statistics differ from those used to design the Huffman code, then there will be a degradation in performance due to 'mismatch' between the code and the source. Another problem with Huffman coding is that although it is optimum for a particular distribution, it requires a relatively large amount of computation and memory to store all the possible codewords. For a typical facsimile document having 1728 pels per line, the number of possible messages is 2^{1728} — certainly an excessive requirement. It is therefore desirable to produce a code which is 'matched', in a certain way, to the probability distribution of Fig. 6.6 but which does not use a large codebook memory. 'Matching' to this probability distribution is achieved by adaptive division of the scan-line of N pels into smaller blocks of size M pels based on a certain probability criterion. Having done so, the blocks are then coded such that the bit-rate is as close to the entropy as possible.

The criterion is taken to be the probability of correct prediction p_1, and the adaptivity of block size M corresponds to the approximate version of the distribution of prediction errors per scan-line shown in Fig. 6.6. The justification for determining M by 'matching' it to the probability distribution of errors per scan-line is that shorter codewords for the more probable, and larger codewords for the less probable, events are required.

For most scan-lines, the probability of correct prediction p_1 in a scan-line is always very much greater than the probability of prediction error $(1-p_1)$. In other words, p_1 is very nearly unity, and so any decrease in p_1 will cause an increase in entropy. (See Eqn. (6.18)). The higher the entropy, i.e. the greater the number of prediction errors, the more blocks of size M pels are required to code the source. It is assumed that if $p_1 \rightarrow 1$, i.e. when there are no prediction errors on a scan-line, then $M \rightarrow N$ where N is the number of pels per scan-line.

One equation which 'matches' the approximation of Fig. 6.6 and satisfies the above assumption is:-

$$M = -\left[\frac{\log_2 p_1 + Z}{\log_2 p_1}\right] \tag{6.19}$$

where Z is an arbitrary adaptation constant.

Eqn. (6.19) shows how each scan-line of N pels may be sub-divided into smaller blocks of M pels based on the probability criterion mentioned above. It can be simplified to:-

$$M = -\left[1 + \frac{Z}{\log_2 p_1}\right] \tag{6.20}$$

From Eqn. (6.20), if the magnitude of $\frac{Z}{\log_2 p_1} \gg 1$, then

$$M \simeq -\frac{Z}{\log_2 p_1} \tag{6.21}$$

As p_1 decreases, M becomes smaller. The scan-line of length N is therefore divided into $\frac{N}{M}$ blocks each of size M pels. The greater the number of prediction errors in a scan-line, (i.e. the smaller the value of p_1), the more blocks are required to code the source efficiently.

By taking the natural logarithm Eqn. (6.21) may alternatively be written as:-

$$M \simeq - \frac{Z \ln 2}{\ln p_1} \tag{6.22}$$

Since $p_1 = \frac{c}{N}$ where c is the number of correct predictions per scan-line, we obtain:-

$$M \simeq - \frac{Z \ln 2}{\ln (\frac{c}{N})}$$

$$\simeq \frac{Z \ln 2}{\ln (\frac{N}{c})} \tag{6.23}$$

Letting $N = w + c$ where w is the number of prediction errors per scan-line, then

$$\ln (\frac{N}{c}) = \ln (1 + \frac{w}{c}) \tag{6.24}$$

Let $X = \frac{w}{c}$, and using the Maclaurin Series Expansion we can further simplify Eqn. (6.24). The Maclaurin Series Expansion is given by:-

$$f(X) = f(0) + Xf'(0) + \frac{X^2\ f''(0)}{2!} + \frac{X^3\ f'''(0)}{3!} + \ldots$$

where $f'(0)$, $f''(0)$ and $f'''(0)$ are first, second and third derivatives respectively.

Let $f(X) = \ln\ (1 + X)$, then

$$f(X) = X - \frac{X^2}{2} + \frac{X^3}{3} - \frac{X^4}{4} + \ldots.$$

therefore,

$$\ln\ (1 + \frac{w}{c}) = \frac{w}{c} - \frac{(\frac{w}{c})^2}{2} + \frac{(\frac{w}{c})^3}{3} - \frac{(\frac{w}{c})^4}{4} + \ldots\ldots.$$

and if $c \gg w$, then

$$\ln\ (1 + \frac{w}{c}) \simeq \frac{w}{c} \qquad (6.25)$$

Substituting Eqn. (6.25) into Eqn. (6.23), we obtain:-

$$M \simeq \frac{c}{w}\ Z\ \ln 2 \qquad (6.26)$$

Eqn. (6.19) has now been simplified to Eqn. (6.26) which demonstrates that the block size M is dependent on the number of correct predictions in a scan-line. If $p_1 \rightarrow 1$ ($w \rightarrow 0$, i.e. when there are no prediction errors), then from Eqn (6.26), $M \rightarrow \infty$. Since it is not possible to have blocks of infinite size, if this condition occurs, M is taken to equal N, satisfying the assumptions made earlier.

The adaptation of block size M with respect to the number of prediction errors per scan-line, obtained from Eqn. (6.26) for $N = 1728$ pels, is shown in Fig. 6.7 where it can be seen that the distribution of M approximates to the probability distribution of the number of prediction errors per scan-line of Fig. 6.6. Z, for simplicity, is taken to be 1. As can be seen from Fig. 6.7, the most probable event is assigned the biggest block size, thereby requiring fewer bits, while the less probable events use smaller block sizes, needing a greater number of bits for coding.

Since the values of M obtained from Eqn. (6.26) are not normally multiples of powers of 2, the following relationships are introduced:-

$$\left.\begin{array}{l} \text{If } \left| M - 2^{m} \right| > \left| M - 2^{m+1} \right| \\ \text{then the value of } M \text{ is taken to be } 2^{m+1} \\ \\ \text{If } \left| M - 2^{m} \right| < \left| M - 2^{m+1} \right| \end{array}\right\} \qquad (6.27)$$

then the value of M is taken to be 2^{m}, where $m = 1,2,\ldots 11$.

Having divided the scan-lines, according to Eqn. (6.26) and Eqn. (6.27), into blocks of maximum size M pels, bit assignment now takes place as follows:-

(a) The coder examines the first block of M pels in a scan-line. If it contains no prediction errors, a one bit prefix code '0' is assigned.

(b) If a prediction error is encountered, its location

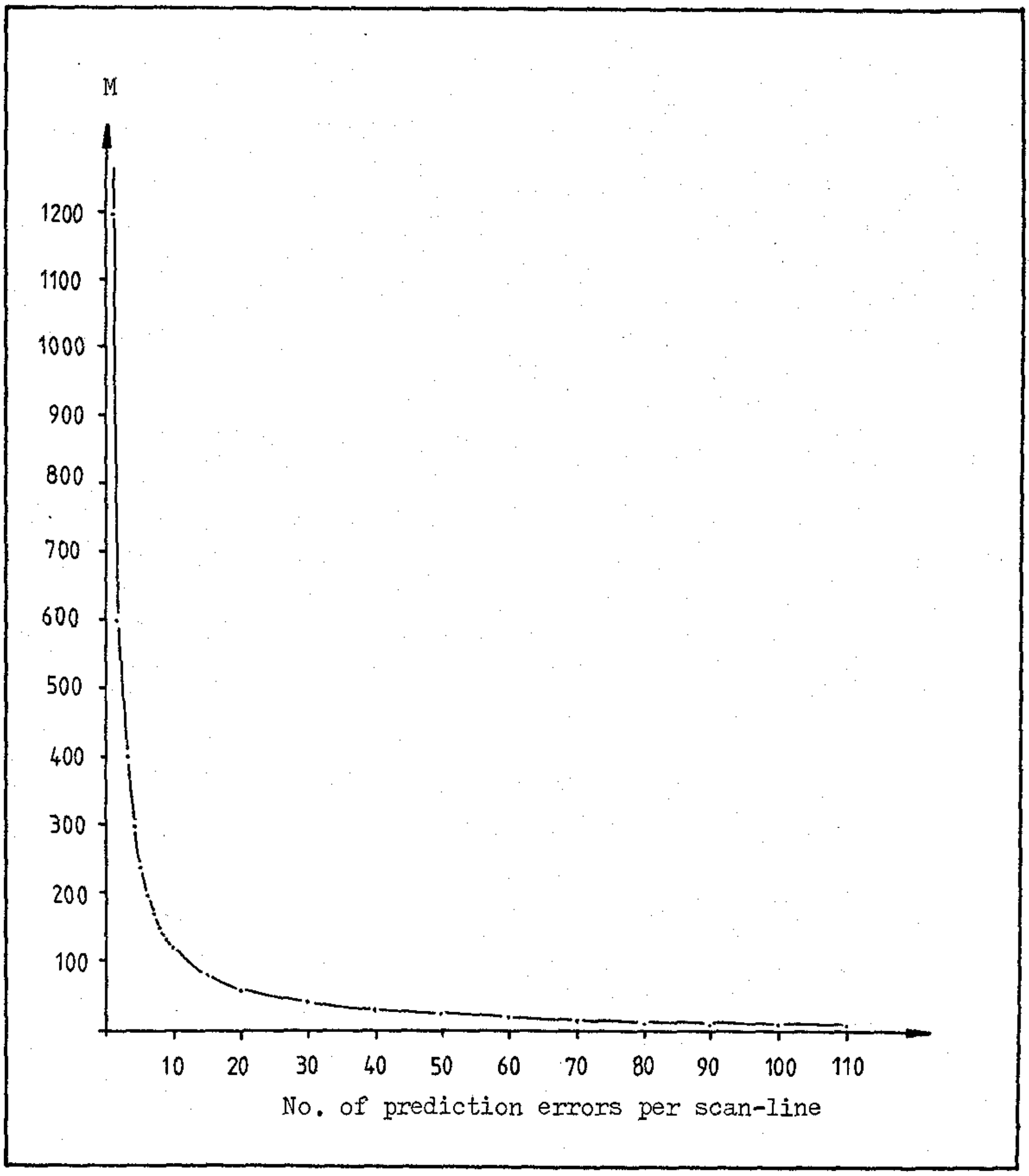


Fig. 6.7 - Adaptation of block size M with respect to number of prediction errors.

is registered by coding its distance from the beginning of the block. This information is then transmitted with a prefix '1'.

(c) The block terminates at that point. The next block starts from the last coded prediction error. The procedure is then repeated.

This bit assignment procedure gives rise to the name Adaptive Block/Run-Length Coding, and since the predicted values are classified into the Lesser Error State and the Greater Error State, the name Classified Adaptive Block/Run-Length Coding is appropriate, abbreviated to CABC. CABC is illustrated in Fig. 6.8, where Fig. 6.8(a) shows the location of the prediction errors and the corresponding codewords, and Fig. 6.8(b) exemplifies the coding technique where $M = 8$.

6.4.1 Optimality of Block Size M

In this sub-section, the optimality of Eqn. (6.26) is demonstrated and the validity of the earlier analysis tested under the assumptions made.

Consider once again the case before classification into the two general states, where each scan-line containing the prediction errors is made up of N pels. If we divide each scan-line into smaller blocks, where the maximum block size is M pels, according to the coding strategy and the bit assignment procedure described earlier, and let P_M be the probability that the maximum block size

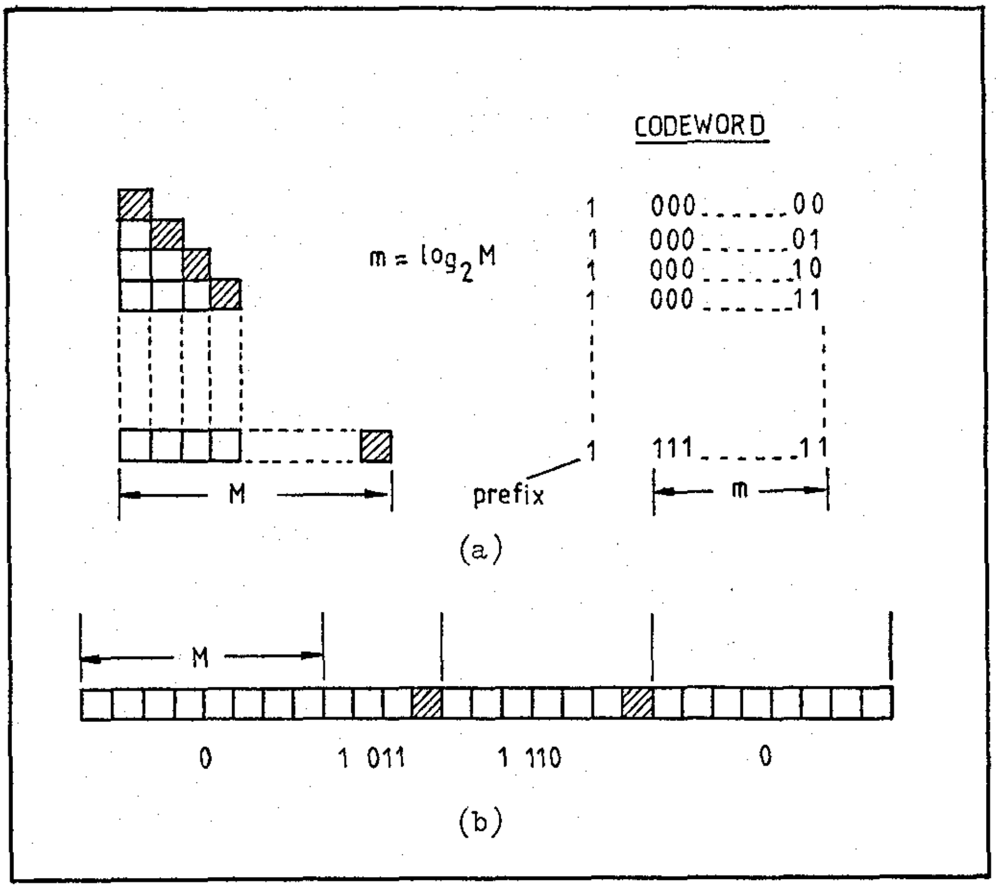


Fig. 6.8 - Adaptive Block/Run-Length Coding

(a) Bit Assignment

(b) Example of coding technique where M = 8

M contains all white pels, then the generalised equation for the average number of bits per pel can be written as:-

$$B_N = \frac{1}{\bar{M}} \left[1.P_M + (1-P_M)(1 + \log_2 M) \right] + \frac{K_c}{N} \qquad (6.28)$$

$1 + \log_2 M$ represents the prefix bit, and the number of bits required to code the positions of prediction errors, respectively,, $\bar{M}$ is the average block size, defined as the total number of pels divided by the number of blocks in a scan-line, and K_c represents flag bits indicating the maximum block size M to the receiver. Eqn. (6.28) can be further simplified to:-

$$B_N = \frac{1}{\bar{M}} \left[1 + (1-P_M) \log_2 M \right] + \frac{K_c}{N} \qquad (6.29)$$

To demonstrate the optimality of Eqn. (6.26), the "ideal" case is first considered in which the prediction errors are situated in the last locations of the blocks as shown in Fig. 6.9(a). Assuming that the relationship between M and w is not known, the average number of bits per pel, can be given by inspection as:-

$$B_I = \frac{1}{N} \left(\frac{N}{M} + w \log_2 M + K_c \right) \qquad (6.30)$$

where N is the number of pels per scan-line and w is the number of prediction errors. The term $w \log_2 M$ denotes the number of bits required to code the position of prediction errors (in Fig. 6.9(a), for example, $w = 2$ and $M = 8$; the number of bits required is therefore 6), and $\frac{N}{M}$ represents the total number of prefix bits required. Simplifying Eqn. (6.30) gives:-

$$B_I = \frac{1}{M} + \frac{w}{N}\log_2 M + \frac{K_c}{N} \qquad (6.31)$$

If Eqn. (6.29) is applied to the "ideal" case condition where the average block size ($\overline{M}$) is equal to M , then

$$B_N = \frac{1}{M} + \frac{(1-P_M)\log_2 M}{M} + \frac{K_c}{N} \qquad (6.32)$$

Comparing Eqn. (6.31) with Eqn. (6.32) (the generalised equation), we obtain:-

$$\frac{(1-P_M)}{M} = \frac{w}{N} \qquad (6.33)$$

This relationship of Eqn. (6.33) will be required when considering the "arbitrary" case.

In order to find the maximum block size M which minimises B_I , we need to equate the partial derivative $\frac{\partial B_I}{\partial M}$ to zero, given that w is fixed for a given scan-line.
From Eqn. (6.31) ,

$$B_I = \frac{1}{M} + \frac{w \ln M}{N \ln 2} + \frac{K_c}{N}$$

Hence $\frac{\partial B_I}{\partial M} = -\frac{1}{M^2} + \frac{w}{MN \ln 2} \qquad = 0$

and so $M = \frac{N \ln 2}{w}$ (6.34)

(Note the similarity between Eqn. (6.34) and Eqn. (6.26))

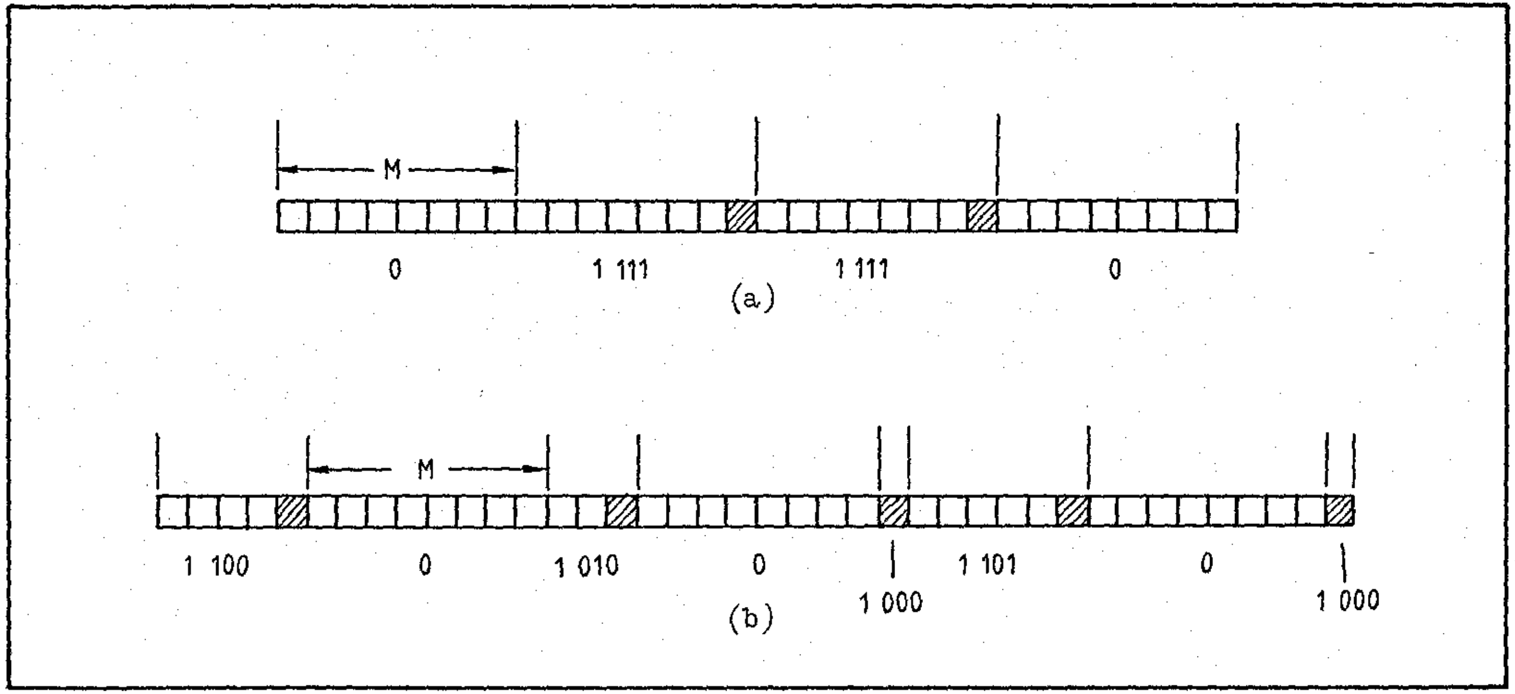


Fig. 6.9 - The optimality of block size M

(a) "Ideal" case

(b) "Arbitrary" case

Consider now an "arbitrary" case in which the prediction errors are situated anywhere on a scan-line as shown in Fig. 6.9(b). The average number of bits per pel can be represented by Eqn. (6.29) viz:-

$$B_N = \frac{1}{\bar{M}} + \frac{(1-P_M)}{\bar{M}} \log_2 M + \frac{K_c}{N} \qquad (6.35)$$

From Eqn. (6.33), and by inspection, it was found that:-

$$\frac{(1-P_M)}{\bar{M}} = \frac{w}{N} \qquad (6.36)$$

The relationship of Eqn. (6.36) is true for all cases.

Eqn. (6.35) therefore becomes:-

$$B_N = \frac{1}{\bar{M}} + \frac{w}{N} \log_2 M + \frac{K_c}{N} \qquad (6.37)$$

In order to find the optimum block size M which minimises B_N, we have to solve $\frac{\partial B_N}{\partial M} = 0$, but we need to know the dependence of $\bar{M}$ on M. Instead of solving Eqn. (6.37) numerically for the optimum M, we choose to evaluate, using computer simulation, the bit-rate B_N using typical facsimile data where each scan-line contains different numbers of prediction errors. By doing so, a better idea of how B_N depends on w and M can be obtained.

Fig. 6.10 shows B_N plotted as a function of block size M for $N = 256$ and $w = 3,6,9,16,21,25,$ and 30. The values of M are chosen to be powers of two, consistent with the earlier analysis. From Fig. 6.10, as w increases, B_N also increases. However, one major conclusion that can be drawn from the graph is that near the minima (circled on graph), B_N is relatively insensitive to slight variation in M . If the next higher value of M from the minimum is taken, B_N only changes by a maximum of about six percent. If Eqn. (6.34) and Eqn. (6.27) are used to determine the optimum M for an "arbitrary" case using different values of w (as shown on Fig. 6.10), it is found that the minimum B_N obtained are the same as those circled. From this evidence, it can be concluded that the value of $\overline{M}$ (the average block size) for typical facsimile data is nearly unity and that the optimum M can be safely calculated for all cases using Eqn. (6.34).

Using the above theoretical analysis, the optimality of Eqn. (6.26) has been demonstrated. The difference between Eqn. (6.34) and Eqn. (6.26) is only marginal if the adaptation constant Z (see Eqn. (6.26)) is taken to be 1 and Z can, of course be changed to account for this difference. It is shown that adaptive division of the scan-line according to Eqn. (6.34) agrees very well with the earlier analysis where M was obtained from the probability distribution of prediction errors per scan-line. It can be seen, therefore, that the value of M obtained from Eqn. (6.26) is nearly optimum.

Fig. 6.11 shows the variation of bit-rates with Z for CCITT document Nos. 1 and 7. It can be clearly seen that the minimum bit-rate is obtained when the value of Z is approximately 1. For the sake of

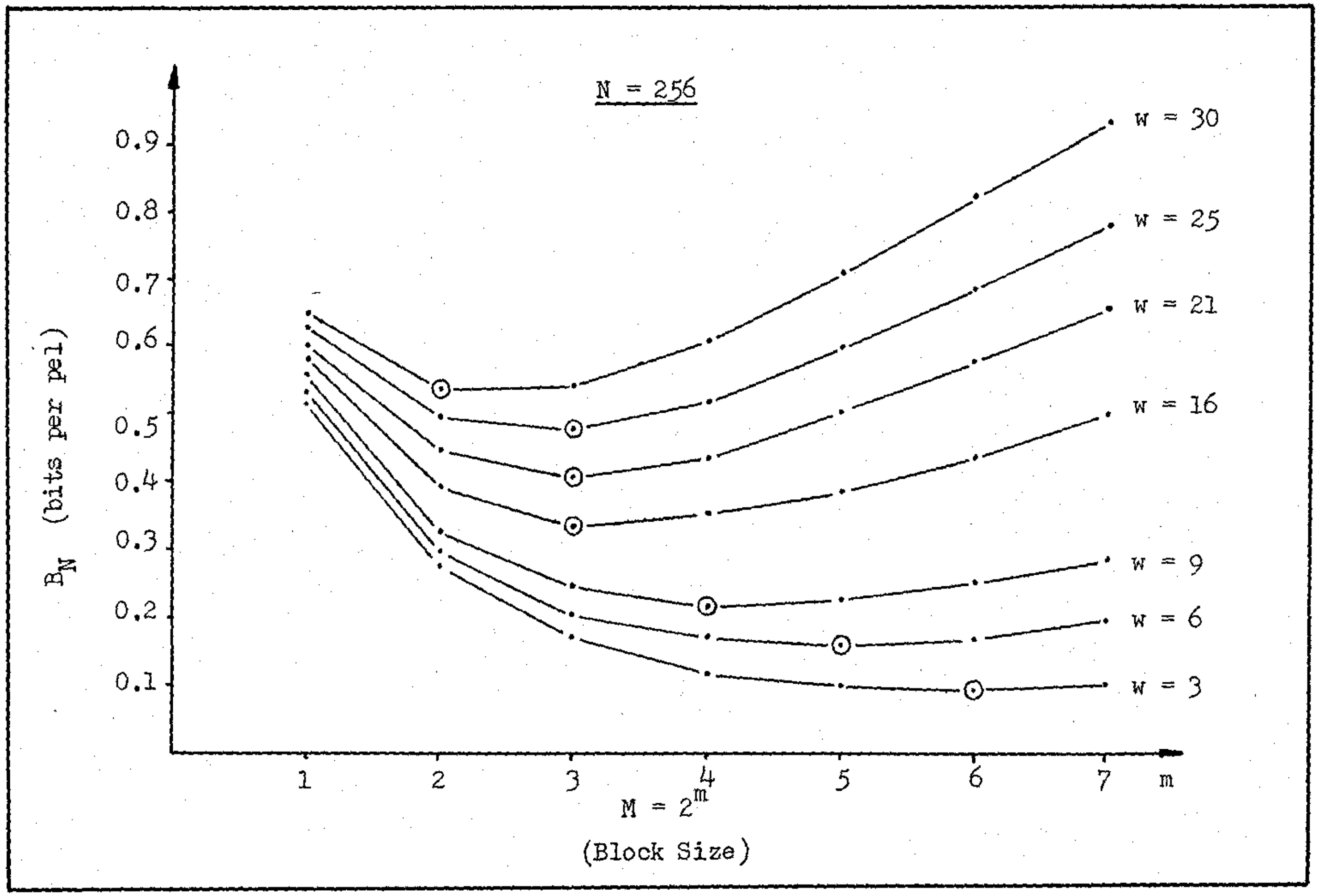


Fig. 6.10 - Bit-rate (B_N) as a function of block size (M) for various values of w

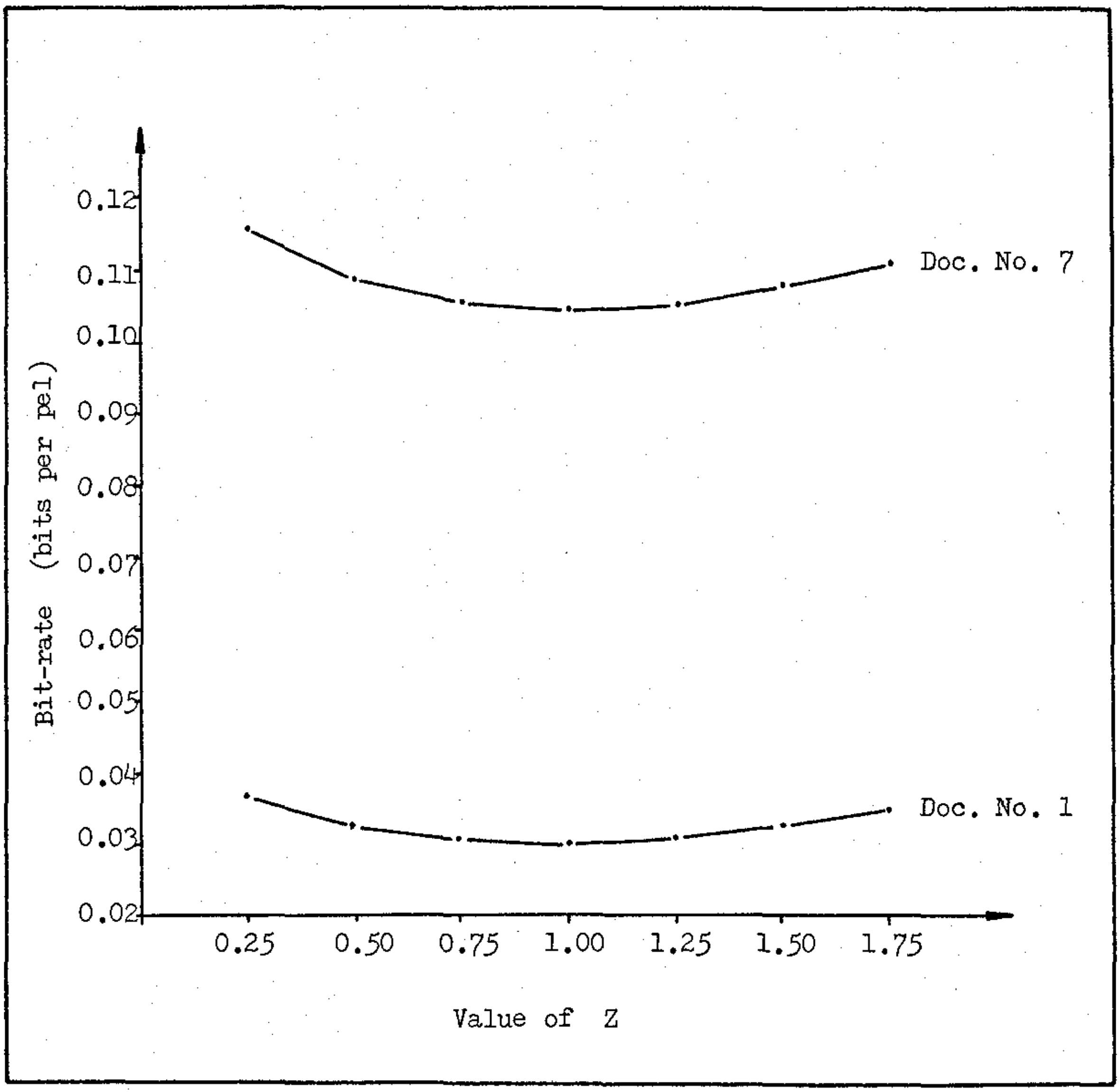


Fig. 6.11 - Variation of bit-rate with Z

simplicity, the results of Section 6.5 are computed with Z = 1.

6.4.2 Coding Strategy Using A Huffman Code

The use of the Huffman algorithm(36) to represent the location of prediction errors is now examined. The usual Huffman procedure for source encoding requires a codebook memory for translation of the source messages into variable length codewords. When some of the messages are highly improbable, the size of the codebook becomes very large. To avoid this, the combination of the bit assignment procedure of Fig. 6.8 and the Huffman procedure is employed to define the location of the prediction errors. For block sizes of M = 8,16,32, and 64, the Huffman code is used to define the location of prediction errors within the block, where the codewords are as shown in Fig. 6.12. These codewords were constructed using probabilities averaged over five CCITT documents. P(r) is defined as the probability of a prediction error located at distance r from the beginning of the block. For other block sizes, the bit assignment of Fig. 6.8 is used.

6.5. SIMULATION RESULTS

In order to evaluate the performance of the CABC algorithm, it was compared, in terms of entropy, with the Classified Pel Pattern Method (CLAP)(66). Five CCITT SG.XIV test documents digitised at 8 pels/mm horizontally and vertically were used, and the prediction function for the unprocessed and preprocessed images were optimised

Block Size M = 8 (2^3)

Location	P(r)	Codeword
1	0.2199	01
2	0.1781	111
3	0.1256	100
4	0.1266	101
5	0.0790	1100
6	0.1024	001
7	0.0812	1101
8	0.0872	000

Block Size M = 16 (2^4)

Location	P(r)	Codeword
1	0.119	011
2	0.0975	001
3	0.0837	000
4	0.0807	1110
5	0.0753	1100
6	0.0832	1111
7	0.0660	1000
8	0.0696	1001
9	0.0739	1011
10	0.0518	0100
11	0.0359	10101
12	0.0383	11011
13	0.0317	01011
14	0.0383	11010
15	0.0274	01010
16	0.0348	10100

Fig. 6.12 - CABC using a Huffman Code

Block Size M = 32 (2^5)

Location	P(r)	Codeword
1	0.0101	1111100
2	0.0724	1011
3	0.0629	0111
4	0.0751	1110
5	0.0730	1100
6	0.0681	1010
7	0.0675	1001
8	0.0582	0101
9	0.0491	0010
10	0.0376	11011
11	0.0336	10001
12	0.0294	01100
13	0.0276	01001
14	0.0255	00111
15	0.0216	00000
16	0.0235	00011
17	0.0235	00010
18	0.0239	00110
19	0.0189	111100
20	0.0186	110101
21	0.0206	111111
22	0.0193	111101
23	0.0179	110100
24	0.0164	100001
25	0.0153	011011
26	0.0152	011010
27	0.0121	000011
28	0.0155	100000
29	0.0136	010001
30	0.0124	010000
31	0.0105	1111101
32	0.0111	000010

Fig. 6.12 - (cont'd)

Block Size M = 64 (2^6)

Location	P(r)	Codeword
1	0.0117	001100
2	0.0503	0100
3	0.0489	0010
4	0.0481	0000
5	0.0462	11111
6	0.0489	0001
7	0.0454	11110
8	0.0422	11100
9	0.0371	11001
10	0.0308	10001
11	0.0269	01100
12	0.0251	00111
13	0.0257	01010
14	0.0183	110000
15	0.0192	110101
16	0.0177	101101
17	0.0191	110100
18	0.0145	011110
19	0.0149	100000
20	0.0166	101001
21	0.0160	101000
22	0.0158	100111
23	0.0178	101110
24	0.0157	100101
25	0.0143	011100
26	0.0185	110001
27	0.0155	100100
28	0.0128	010110
29	0.0106	1110101
30	0.0152	100001
31	0.0108	1110110
32	0.0075	0111111

Location	P(r)	Codeword
33	0.0114	1110111
34	0.0086	1011000
35	0.0098	1101101
36	0.0092	1011111
37	0.0103	1101111
38	0.0089	1011110
39	0.0095	1101100
40	0.0071	0111010
41	0.0080	1001101
42	0.0085	1010110
43	0.0085	1010101
44	0.0086	1010111
45	0.0066	0101111
46	0.0074	0111110
47	0.0068	0110101
48	0.0074	0111011
49	0.0078	1001100
50	0.0060	0011010
51	0.0071	0110111
52	0.0054	11101001
53	0.0065	0101110
54	0.0069	0111010
55	0.0065	0011011
56	0.0029	01101000
57	0.0043	10110010
58	0.0048	11011100
59	0.0046	10110011
60	0.0052	11101000
61	0.0052	11011101
62	0.0040	10101000
63	0.0043	10101001
64	0.0038	01101001

Fig. 6.12 - (cont'd)

for each document.

In CABC, since the prediction errors in the two states are coded on an error by error basis, reducing the number of prediction errors is of prime importance, and this has been achieved by preprocessing the original image prior to coding. The numbers of prediction errors in the original and preprocessed images for the five CCITT documents used are shown in Table 6.3.

	CCITT TEST DOCUMENT NO.				
	1	2	4	5	7
ORIGINAL	32475	19175	131419	56801	115575
PREPROCESSED	20026	13085	78933	33587	81656

Table 6.3 - Reduction in the number of prediction errors by preprocessing.

The table shows that with the introduction of preprocessing using masks, there is a substantial reduction in the number of prediction errors, ranging from about 29 to 41 percent. With this degree of reduction, a considerable improvement in coding efficiency can be expected.

To calculate the entropies before and after classification, Eqn. (6.15) and Eqn. (6.17) respectively were used. The entropies obtained, as shown in Table 6.4, represent the minimum theoretical bit-rates for the different coding schemes. It can be seen that the preprocessed documents, after classification for the proposed CABC scheme, have

CODING SCHEME	CCITT TEST DOCUMENT NUMBER				
	1	2	4	5	7
CLAP (Before classification)	0.070	0.044	0.211	0.109	0.199
CLAP (After classification)	0.035	0.021	0.128	0.060	0.120
CABC (original) (Before classification)	0.067	0.042	0.204	0.105	0.186
CABC (original) (After classification)	0.034	0.020	0.126	0.059	0.116
CABC (preprocessed) (Before classification)	0.044	0.031	0.137	0.068	0.141
CABC (preprocessed) (After classification)	0.029	0.017	0.101	0.048	0.102

Table 6.4 - Theoretical entropy measurements for different coding schemes

entropies ranging from 0.017 to 0.102 bits per pel. This represents possible compression factors of 58.82:1 and 9.8:1 respectively. When compared with CLAP, CABC reduces the entropy by 13 to 21 percent. This indicates that the maximum achievable compression of CABC is on average 1.2 times greater than that of CLAP. When the preprocessed documents are compared with the original, unprocessed documents after classification, for the CABC scheme, an improvement between 12 and 19 percent is observed. The reduction in entropy indicates that preprocessing prior to coding has the effect of improving the coding efficiency substantially. Categorising the prediction errors into Lesser and Greater Error States also improves the coding efficiency by an amount ranging from 34 to 49 percent for the processed documents. The reduction in entropy here may be due to the fact that the numbers of prediction errors in the two states are different, most of the prediction errors being packed in the GES leaving few prediction errors in the IES.

Although entropy evaluation represents the theoretical lower bound, the performance of CABC using the bit assignment technique of Fig. 6.8 and the Huffman code were also investigated. The results, shown in Table 6.5, for CABC and CLAP schemes represent the bit rates, after classification only, for both the original and preprocessed documents. The flag bit is taken to be 4 for each general state (IES and GES) to inform what the maximum block size M is to the receiver. Hence, if M = 16, the flag bit is 0100, i.e.decimal 4 ($2^4 = 16$). However, the results of Table

CODING SCHEME	CCITT TEST DOCUMENT NUMBER				
	1	2	4	5	7
CLAP	0.0370	0.0234	0.1313	0.0614	0.1225
CABC (original)	0.0357	0.0232	0.1293	0.0604	0.1183
CABC (preprocessed)	0.0299	0.0202	0.1067	0.0485	0.1041
CABC (original) (Huffman Code Used)	0.0350	0.0229	0.1290	0.0603	0.1175
CABC (preprocessed) (Huffman Code Used)	0.0295	0.0200	0.1061	0.0481	0.1033

Table 6.5 - Comparison of bit-rates after classification

6.5 do not include end-of-line codes, etc, and the parameter K (See Chapter III) is taken to be infinity.

The performance of the CABC scheme was compared with conventional run-length coding using the Modified Huffman code(42,43) and also with the CLAP(66) method. It is apparent that CABC performs better than its counterparts for both the original and preprocessed documents. There is an improvement of about 41 to 61 percent, for the preprocessed documents, when the CABC scheme is compared with run-length coding using the Modified Huffman code. The reduction in bit-rate is particularly high when simple, uncomplicated documents are used, for example, document No. 2. When the preprocessed documents are compared with the originals, for the CABC scheme, there is a reduction in average bit-rate of about 13 to 20 percent. There is a further, though marginal, reduction in bit-rate when the combination of the Huffman code and the bit assignment procedure of Fig. 6.8 is employed. Comparison of theoretical entropies of Table 6.4 with the practical bit-rates of Table 6.5 demonstrates that the coding strategy for CABC, especially with regard to the determination of block size M is nearly optimum.

6.6. EFFECTS OF, AND SENSITIVITY TO, TRANSMISSION ERRORS

One penalty for efficient source coding is the increased sensitivity to channel errors. One single bit error can ruin the rest of a scan-line due to loss of synchronisation between codewords. As with most two-dimensional coding algorithms, the effect of transmission errors on the CABC scheme is to cause error propagation in the decoded image. Unlike in the one-dimensional case, where error propagation is limited to a single scan-line due to the unique EOL codeword, in the CABC scheme, errors tend to propagate vertically from one scan-line to the next. The effect is quite intolerable in most cases, but such propagation can be limited by transmitting one line every Kth line using a one-dimensional coding scheme. Values of the K-factor are typically 2 for documents scanned at normal resolution, and 4 for those scanned at high resolution.

The effects of increasing numbers of channel errors on the CABC scheme were studied and in each case, the Error Sensitivity Factor (ESF)(106) calculated. ESF is defined as the number of incorrect pels in the received image divided by the number of transmitted bits in error. Transmission errors were introduced randomly and simulation was carried out using bit by bit inversion of the transmitted bitstream. To simulate short burst errors, double errors are introduced, where two consecutive bits are inverted. These simulations are by no means typical of a "real-life" telephone channel but they do give an indication of the effect of channel errors on the coded image. The introduction of these errors causes a spatial shift in the received picture. Furthermore, if the decoded run-length between two consecutive EOLs is not equal to the nominal scan-line length, transmission errors can be safely assumed to have occurred.

6.6.1 Experimental Procedures

The basic configuration illustrating the overall simulation process for the study of channel error effects on the CABC scheme is shown in Fig. 6.13.

The input test image, stored on magnetic tape, is part of preprocessed CCITT document No. 1, whose resolution is 8 pels/mm horizontally and vertically. The CABC encoder compresses and transforms the input image into a digital bitstream, after which random errors are introduced. The corrupted bit stream is then decoded producing images as shown in Fig. 6.14. The difference between the output and input images, on a pel-by-pel basis, is also computed in order to calculate the ESF.

The source of transmission errors is a Random Error Generator subroutine which introduces errors into the channel in a random fashion. Two kinds of error configuration were simulated — random single errors and random double errors, which simulate, respectively, the occurrence of isolated spikes and short spurious bursts. In both cases, the corrupted bits in question are inverted.

The introduction of channel errors using the CABC algorithm both before and after classification was studied. The effect of increasing the bit error rate (BER) on the received picture was observed and the sensitivity of the system to channel errors measured. These measurements were taken ignoring such factors such as FILL bits, where a certain minimum scan-line time (MSLT) is required. To limit vertical propagation, the K-factor is taken to be four. It was decided not to transmit every Kth line by the one-dimensional Modified Huffman code in order to ensure that the effect of channel errors are not felt on these scan-lines. In

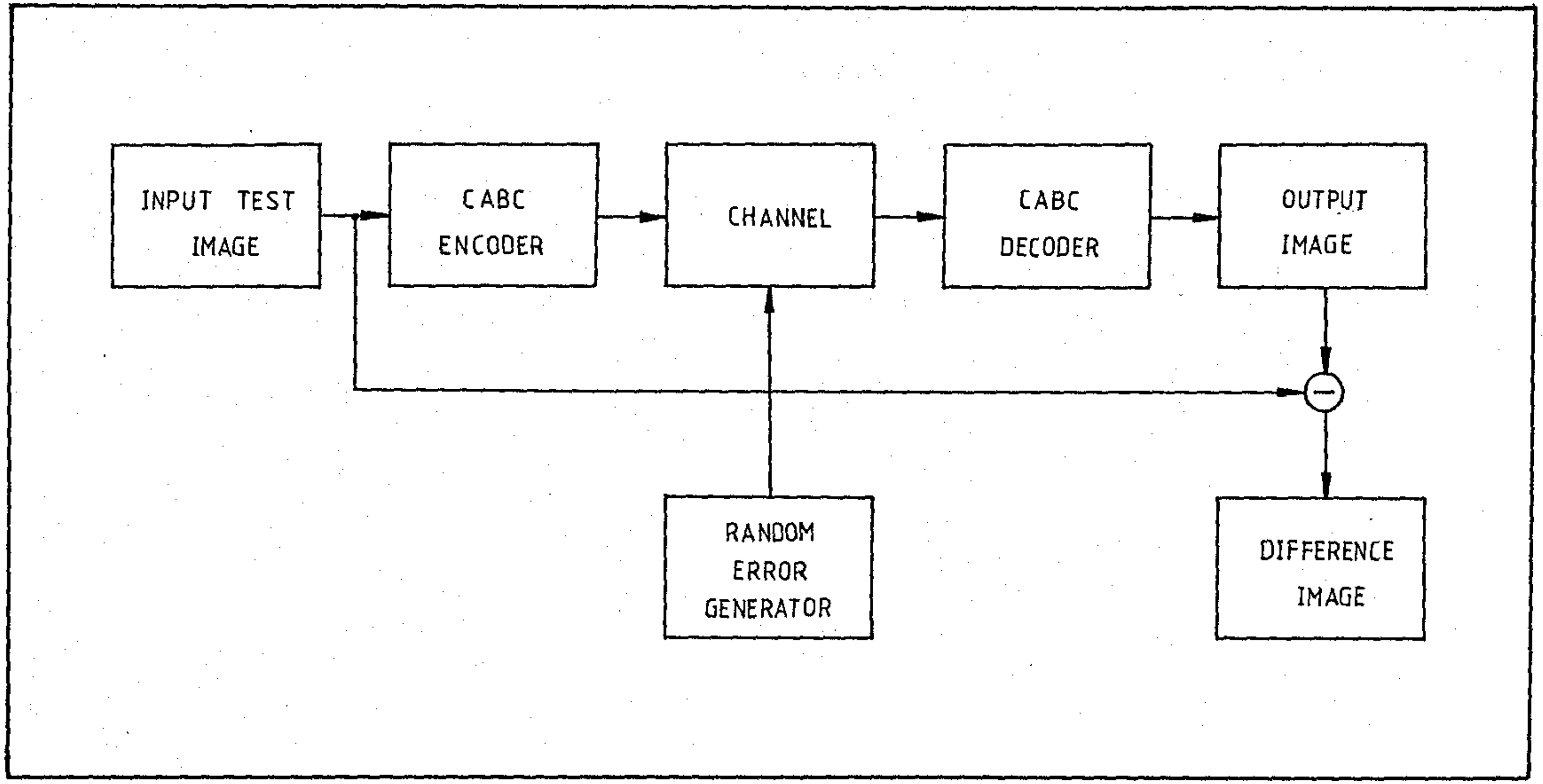


Fig. 6.13 - Block diagram of simulation process

other words, these scan-lines are left uncoded. This is done to limit vertical propagation to within K-1 lines. Furthermore, only the effects of channel errors on the two-dimensional CABC scheme are of interest. For all cases, the EOL codewords are included in the simulation process.

6.6.2 Test Results

Eight test runs were performed using the CABC algorithm, with and without classification, for both single and double errors. The bit error-rate (BER) is increased for each run where BER is defined as the ratio of the number of bits transmitted in error to the total number of coded bits. Since every Kth line is transmitted uncoded, the total number of bits is taken as that generated by the CABC encoder and the EOL codewords only. FILL bits are not included.

Table 6.6(a) and Table 6.6(b) show the results obtained for single errors before and after classification, whilst Table 6.6(c) and Table 6.6(d) show the results obtained for double errors, again before and after classification, respectively. Notice that the number of coded bits for CABC after classification is less than that for the case before classification. For each BER, the corresponding ESF is measured, and it can be observed that there is no drastic fluctuation in this quantity. The average ESF of the CABC system before and after classification for single and double errors is listed in Table 6.7. It is observed that for the introduction of double errors, the CABC system after classification has a superior

No. of bits in error transmitted	No. of coded bits	BER 10^{-3}	No. of incorrect pels	ESF
10	13124	0.76	734	73.40
13	13124	0.99	963	74.07
20	13124	1.52	1480	74.00
37	13124	2.82	3095	83.65
47	13124	3.58	3203	68.15
55	13124	4.19	4681	85.11
124	13124	9.45	7039	56.77
159	13124	12.11	9490	59.68

Table 6.6(a) - Before Classification
(Single Errors)

No. of bits in error transmitted	No. of coded bits	BER 10^{-3}	No. of incorrect pels	ESF
10	12651	0.79	880	88.00
13	12651	1.03	796	61.23
20	12651	1.58	1745	87.25
37	12651	2.92	2774	74.97
47	12651	3.71	4570	97.23
55	12651	4.35	3803	69.14
124	12651	9.80	6669	53.78
159	12651	12.56	8354	52.54

Table 6.6(b) - After Classification
(Single Errors)

No. of bits in error transmitted	No. of coded bits	BER 10^{-3}	No. of incorrect pels	ESF
20	13124	1.52	621	31.05
26	13124	1.98	910	35.00
40	13124	3.05	1150	28.75
74	13124	5.64	2131	28.79
94	13124	7.16	3126	33.26
110	13124	8.38	3485	31.68
248	13124	18.89	6947	28.01
318	13124	24.23	8461	26.61

Table 6.6(c) - Before Classification
(Double Errors)

No. of bits in error transmitted	No. of coded bits	BER 10^{-3}	No. of incorrect pels	ESF
20	12651	1.58	678	33.90
26	12651	2.06	351	13.50
40	12651	3.16	1360	34.00
74	12651	5.85	2497	33.74
94	12651	7.43	3242	34.49
110	12651	8.69	3038	27.62
248	12651	19.60	6707	27.04
318	12651	25.14	11004	34.60

Table 6.6(d) - After Classification
(Double Errors)

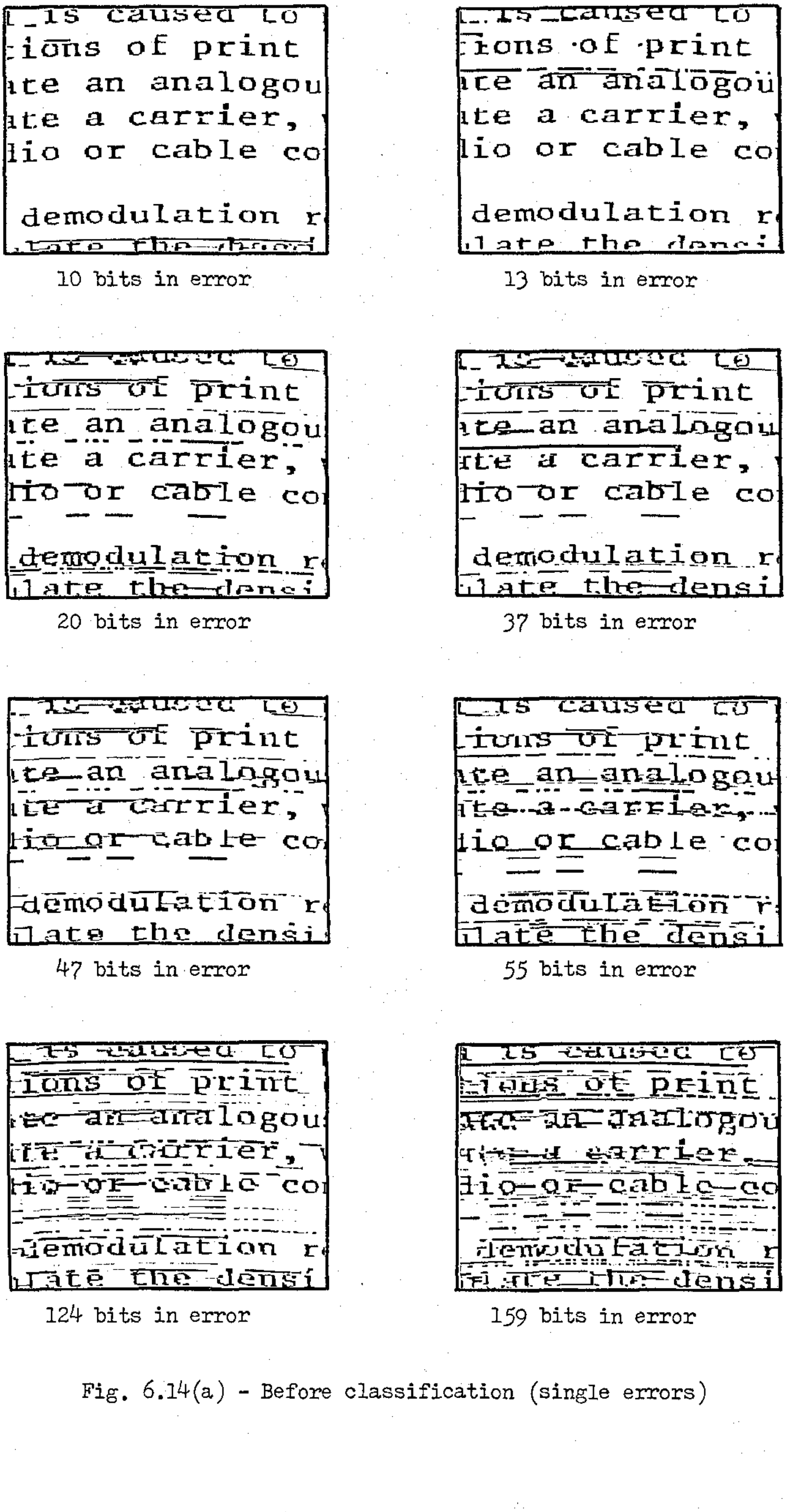
CODING SCHEME	ESF_{AVG} FOR EIGHT RUNS
BEFORE CLASSIFICATION - SINGLE ERRORS	71.85
AFTER CLASSIFICATION - SINGLE ERRORS	73.02
BEFORE CLASSIFICATION - DOUBLE ERRORS	30.39
AFTER CLASSIFICATION - DOUBLE ERRORS	29.86

Table 6.7 - Average ESF

ESF, indicating that it is less vulnerable to transmission errors when compared with its counterparts. With regard to single errors, the CABC system after classification is the most susceptible to errors. From these results, it is obvious that the CABC system is more resilient to double errors and that double errors (burst errors) are easier to deal with than single errors. In a real life situation, for a given BER, burst errors would pose fewer problems than single errors because errors coming in bursts will obliterate a single or up to K-1 lines only, whereas single errors occurring at random would cause greater degradation. Fig. 6.14 contains pictures of part of CCITT document No. 1 which have been subjected to single and double errors before and after classification using the CABC scheme. The degradation becomes increasingly worse as BER is increased. At higher BER, the destruction of characters results in totally unacceptable pictures. At lower rates, however, the errors can be tolerated, and characters are still legible. The extent of picture degradation depends very much on where the errors occur. If on a white background, the reduction in overall quality will be less than if they occurred on the characters themselves.

6.7. CONCLUSION

This chapter has been devoted to the description of the CABC algorithm for efficient coding of black and white facsimile pictures. The scheme employs a preprocessing technique which reduces the amount of redundant information in the original image. The coding strategy, which uses a 7th-order Markov model predictor and divides


is caused to
ions of print
ite an analogou
ite a carrier,
lio or cable co
demodulation r
late the densi

10 bits in error

13 bits in error

20 bits in error

37 bits in error

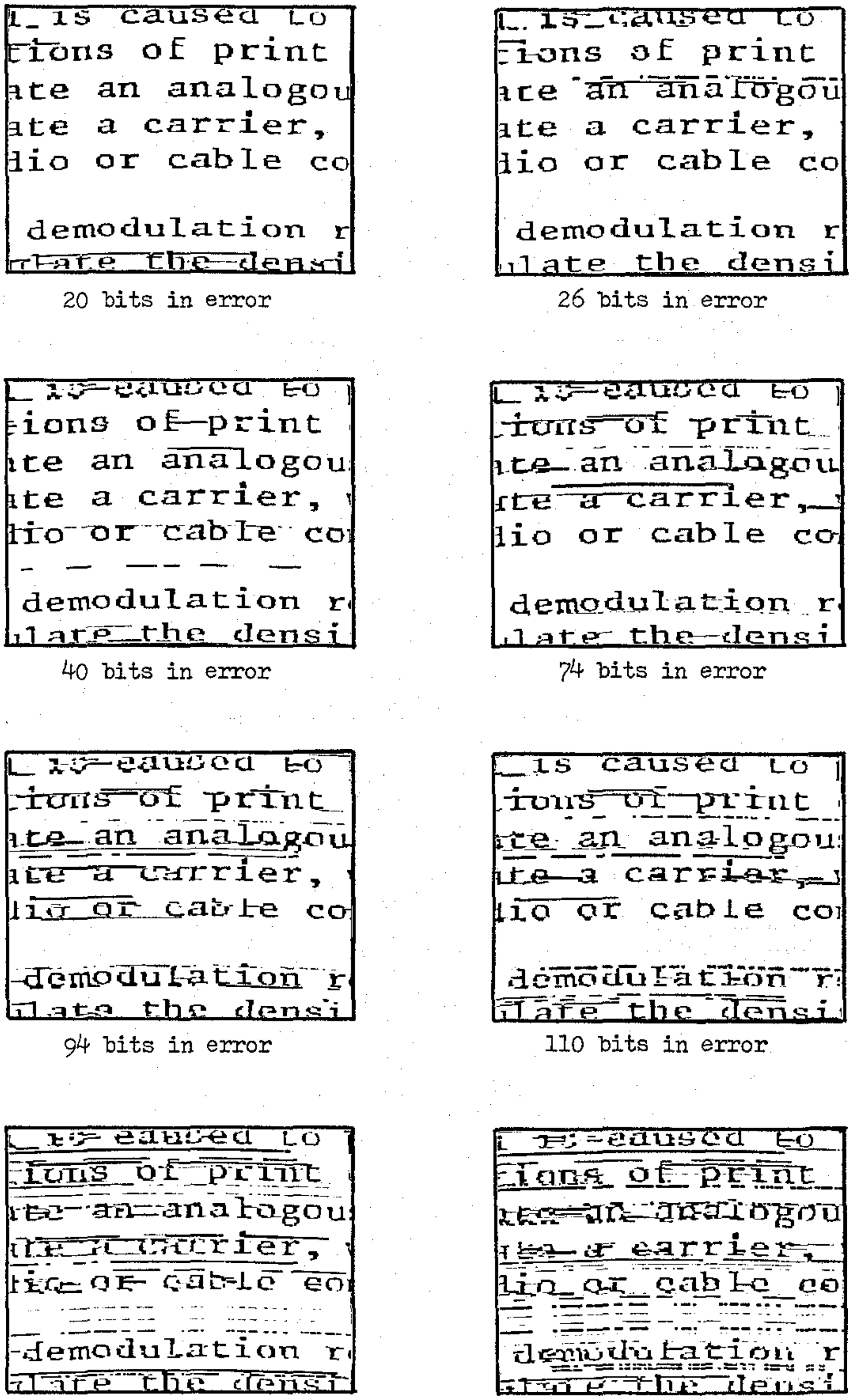
47 bits in error

55 bits in error

124 bits in error

159 bits in error

Fig. 6.14(a) - Before classification (single errors)



20 bits in error

26 bits in error

40 bits in error

74 bits in error

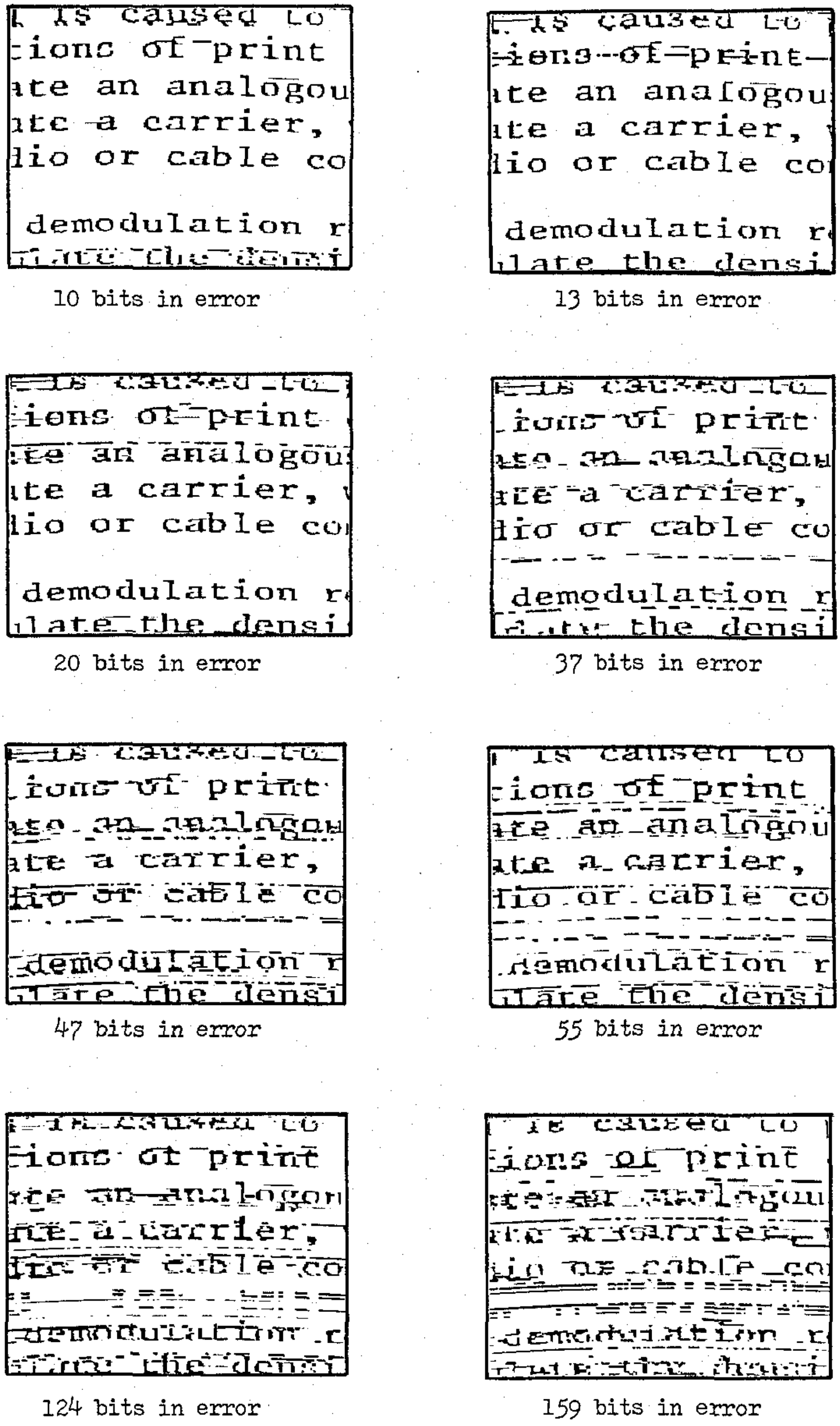
94 bits in error

110 bits in error

248 bits in error

318 bits in error

Fig. 6.14(b) - Before classification (double errors)



10 bits in error

13 bits in error

20 bits in error

37 bits in error

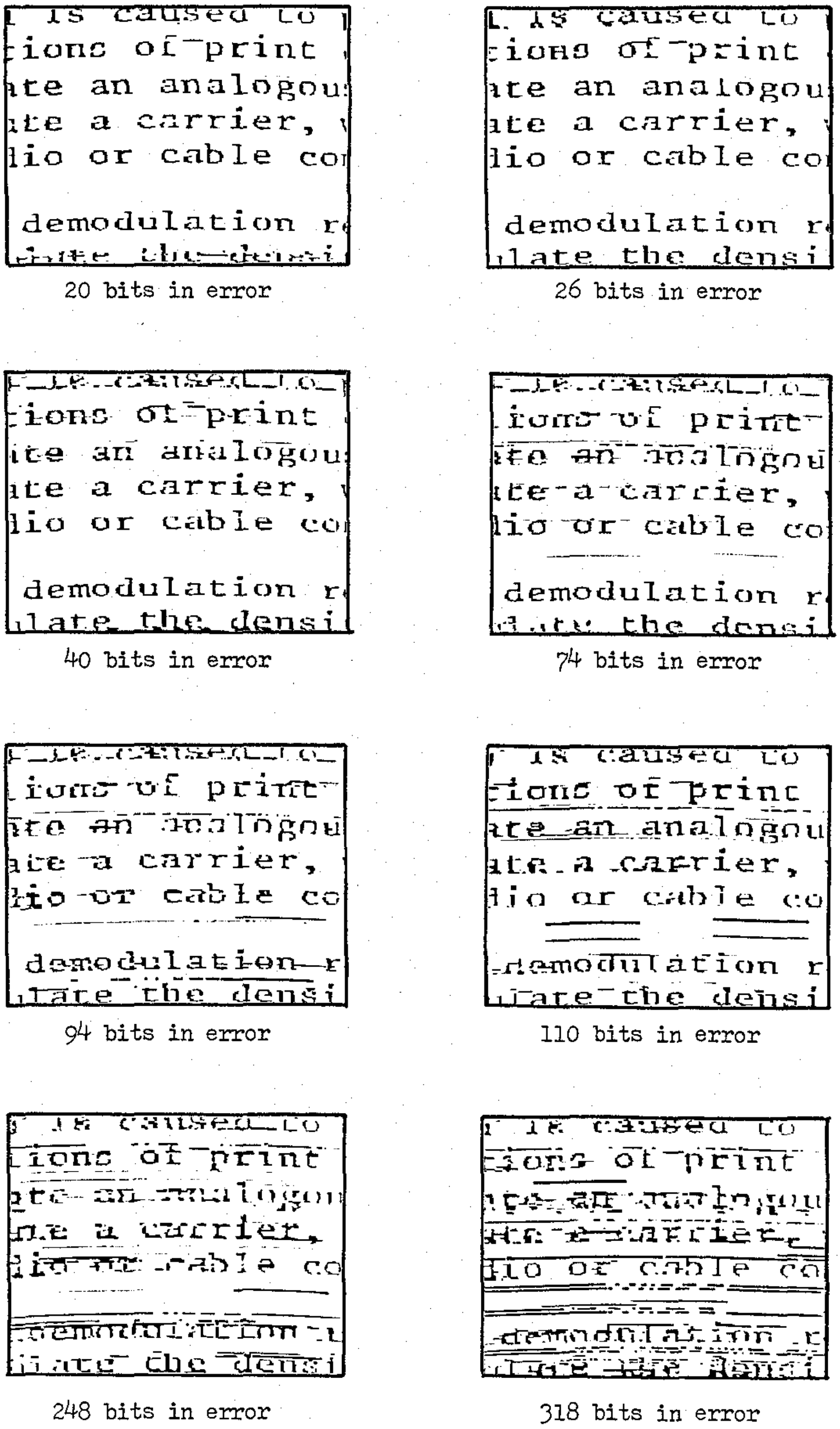
47 bits in error

55 bits in error

124 bits in error

159 bits in error

Fig. 6.14(c) - After classification (single errors)



20 bits in error

26 bits in error

40 bits in error

74 bits in error

94 bits in error

110 bits in error

248 bits in error

318 bits in error

Fig. 6.14(d) - After classification (double errors) the Lesser and Greater Error States adaptively into smaller blocks, is nearly optimum. Developed by obtaining an equation matched to the probability distribution of the number of prediction errors per scan-line, its optimality is later confirmed. The scheme reduces bit-rates by 13 to 20 percent when compared with the CLAP method and by 41 to 61 percent when compared with run-length coding using the Modified Huffman code. Further reduction in bit-rates is achieved when a Huffman code is used to represent the location of prediction errors in the block.

In addition, the sensitivity of the system to channel errors is also studied. The performance of the system in the presence of double errors is as good, if not better, than when single errors are introduced. Finally, compression ratios ranging from 9.4 to 49.4 were obtained for preprocessed pictures corresponding to bit rates ranging from 0.104 to 0.0202 bits per pel.

6.8. NOTE ON PUBLICATION

A paper entitled 'Facsimile Compression Using a Classified Adaptive Block/Run-Length Coding Scheme' was presented at the 1980 National Telecommunication Conference held in Houston, Texas, U.S.A. in December 1980. This paper, in co-authorship with R.J. Clarke, was an abridged description of the CABC scheme described in this chapter.

CHAPTER VII

DATA COMPRESSION II

7.1. INTRODUCTION

The growing need to transmit black and white documents quickly has prompted researchers to resort to an information lossy approach to facsimile coding. Preprocessing[17,120], prior to coding, is one such approach which enables the compression ratios (CRs) of existing coding methods to be significantly enhanced. Signal modification[26], a thinning process[27] and the notchless bilevel quantizer with logical feedback[23] are some of the more important techniques. Preprocessing to clean up noisy originals, to ensure that the CRs remain high, has been extensively studied by Ting and Prasada[16]. The Combined Symbol Matching (CSM) technique[32] is another approach where extremely high CRs are obtained for alphanumeric documents which are efficiently coded by symbol recognition techniques. However, CSM has the disadvantage of not maintaining its superior performance for graphics-type documents and it is relatively complex to implement.

This chapter describes another information lossy approach to facsimile coding using the subsampling technique, outlined in Section 5.3, as the preprocessor. The coding scheme, known as Adaptive Block/Location Coding (ABLC) is an area coding algorithm which takes a square block of picture elements at a time and uses some 'complexity' measure as a coding criterion. It is relatively easy to implement and when used with subsampling as a pre-processing step, overcomes the disadvantage of CSM by performing extremely well for graphic-type documents whilst still maintaining significantly high CRs for alphanumeric type documents. Subsampling is carried out by taking alternate pels horizontally and vertically, prior to which notches and pinholes are removed using the set of masks described in Section 5.2. Single element runs are then doubled both horizontally and vertically to preserve connectivity. Since A4 size documents digitised at a resolution of 8x8 pels per mm are dealt with here, the doubling process ensures that the resultant loss of information is minimal and that subsampling poses little difficulty. It was found in Chapter V that the interpolation technique with proper restoration is feasible for binary images and this would therefore be a natural choice for a post-processor for the complete ABLC system. The basic stages of the system are shown in Fig. 7.1.

Adaptive Block/Location Coding (ABLC) with a numerical measure of complexity(69) as a coding criterion is employed to code the subsampled image. ABLC is compared quantitatively with the D.F.+ coding algorithm of Kawaguchi and Endo(69), (which can be regarded as the generalised version of two-dimensional adaptive block coding(22,35)), and the former is shown to be more efficient. Computer simulations carried out on different variations of the ABLC system with the pre- and post-processors included give maximum CRs ranging from 13.35:1 for document No. 4 to 64.06:1 for document

+ D.F. stands for 'Depth-First' as described in (69)

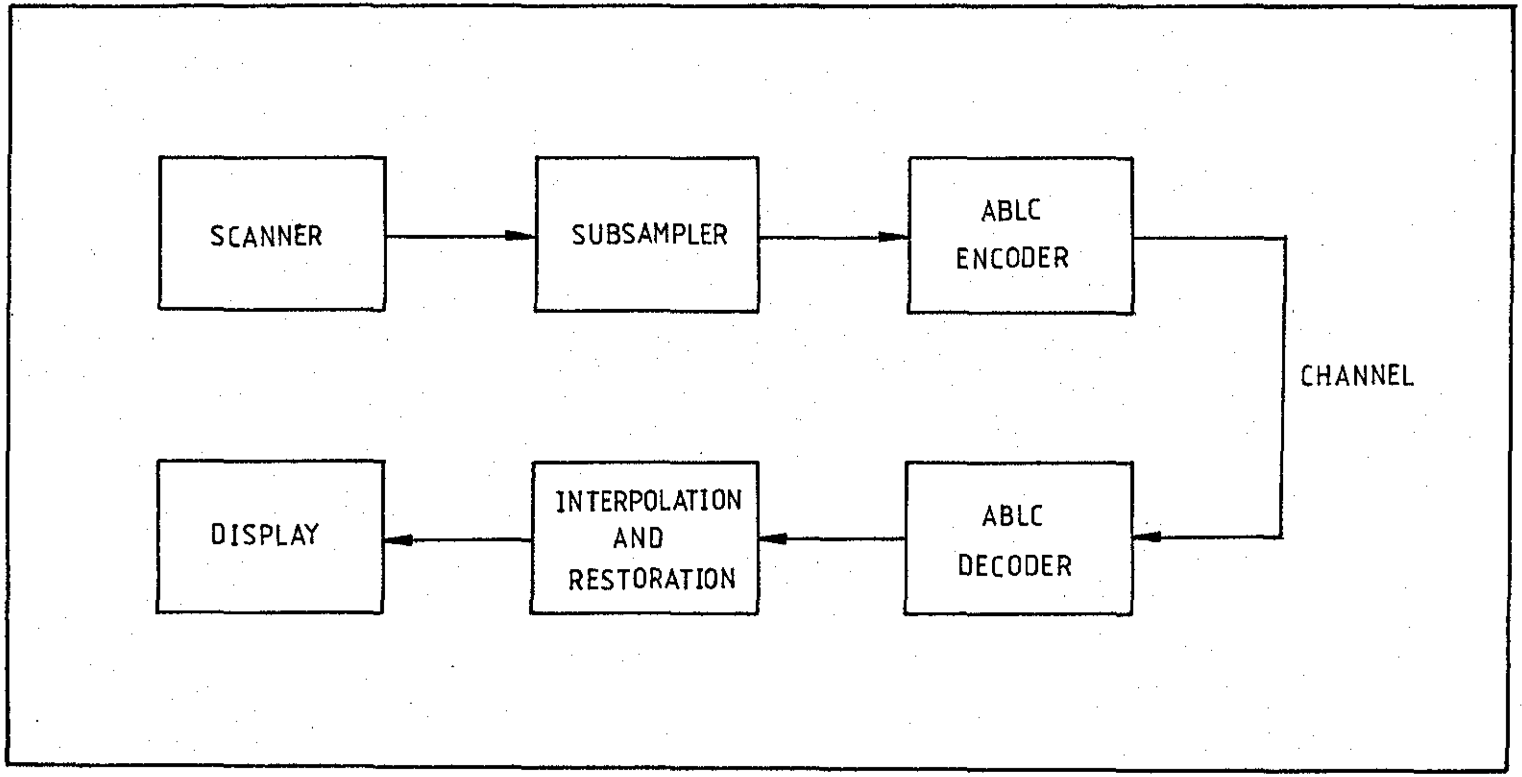


Fig. 7.1 - Block diagram of ABLC system

No. 2.

The description of ABLC begins with preliminary definitions, summarised in Section 7.2, required for the development of the coding algorithm. ABLC is described in Section 7.3 where a quantitative analysis of block and location coding is made. The dependence of bit-rate on image complexity is also examined. Section 7.4 outlines the simulation results obtained for four different configurations of ABLC. The effects of transmission errors on one of the ABLC systems is studied (including pictorial illustrations) in Section 7.5. The possible application of the CCITT coding standards to the subsampled documents is summarised in Section 7.6. The chapter is concluded in Section 7.7 with comparisons between ABLC and the one- and two-dimensional coding standards.

7.2. PRELIMINARY DEFINITIONS

The coding technique to be described makes use of square blocks of size MxM where $M = 2^R$. Every picture block thus consists of $2^R \times 2^R = 4^R$ pels, and R is the number of address bits required in the vertical and horizontal directions for each pel. Fig. 7.2 shows a typical picture block where M = 8.

7.2.1 Block Hierarchy

An initial picture block (IPB) is defined as the initial square area within which all pels in question exist (see Fig. 7.2). A

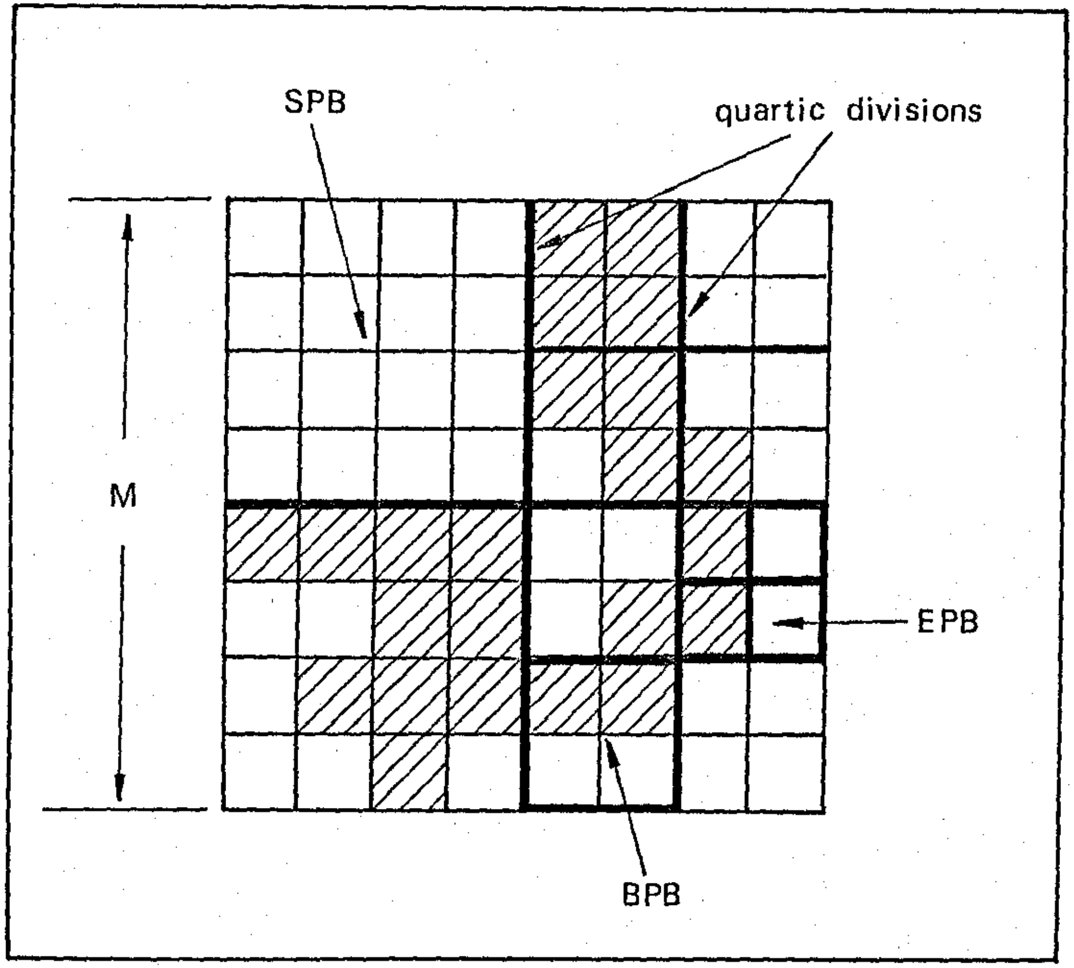


Fig. 7.2 - An Initial Picture Block

subpicture block (SPB) is a subset of IPB which is generated only by quartic divisions of the IPB and the subsequent SPBs until an elementary picture block (EPB) is reached. An EPB is a subpicture block containing one pel. Prior to reaching this state, a basic picture block (BPB) is encountered, which is a SPB containing 4 pels. The quartic division of a BPB results in 4 EPBs.

7.2.2 Picture Primitives

A picture primitive, as defined by Kawaguchi and Endo(69), is a wholly black or wholly white subpicture block which has no lower homogeneous subpictures within it. Fig. 7.2, therefore, contains 37 primitives.

7.2.3 Picture Complexity

Picture Complexity, C_p , is defined as the ratio of the total number of primitives in a picture block to the total number of pels in the same picture block. If we define $m_{\square}$ as the number of white primitives and $n_{\blacksquare}$ as the number of black primitives, then,

$$C_p = \frac{m_{\square} + n_{\blacksquare}}{4^R} \tag{7.1}$$

and $0 < C_p \leqslant 1$.

The complexity of Fig. 7.2 is 37/64.

7.2.4 Relationship between Quartic Division and Complexity

It has been shown(69) that if ℓ_+ is the number of quartic divisions in any picture block, then the total number of primitives (black and white) is equal to $3\ell_+ + 1$.

From Eqn. (7.1),

$$C_p = \frac{3\ell_+ + 1}{4^R}$$

therefore
$$\ell_+ = \frac{4^R C_p - 1}{3} \qquad (7.2)$$

7.3. ADAPTIVE BLOCK/LOCATION CODING (ABLC)

In ABLC, provided an IPB of size M^2 contains lower homogeneous subpictures, it is subdivided into four SPBs (by quartic division) of size $M^2/4$. Each SPB is tested for lower order homogeneous subpictures and if these are present, further quartic divisions are carried out until the BPBs are reached, i.e. successive subdivisions are made on SPBs which are not completely white or completely black. When the BPBs are reached, no further quartic divisions are required and location coding becomes operational. The BPBs of size 2 x 2 pels are coded according to the 14 possible locations of black pels within them (see Fig. 7.3). Bits are then assigned to each quartic division, white primitive greater than 2^2, black primitive greater than 2^2 and the 14 possible locations. ABLC differs from the D.F. coding algorithm in that the former codes the BPBs according to the location of the black pels whereas the latter assigns bits to the black and white primitives in the BPBs.

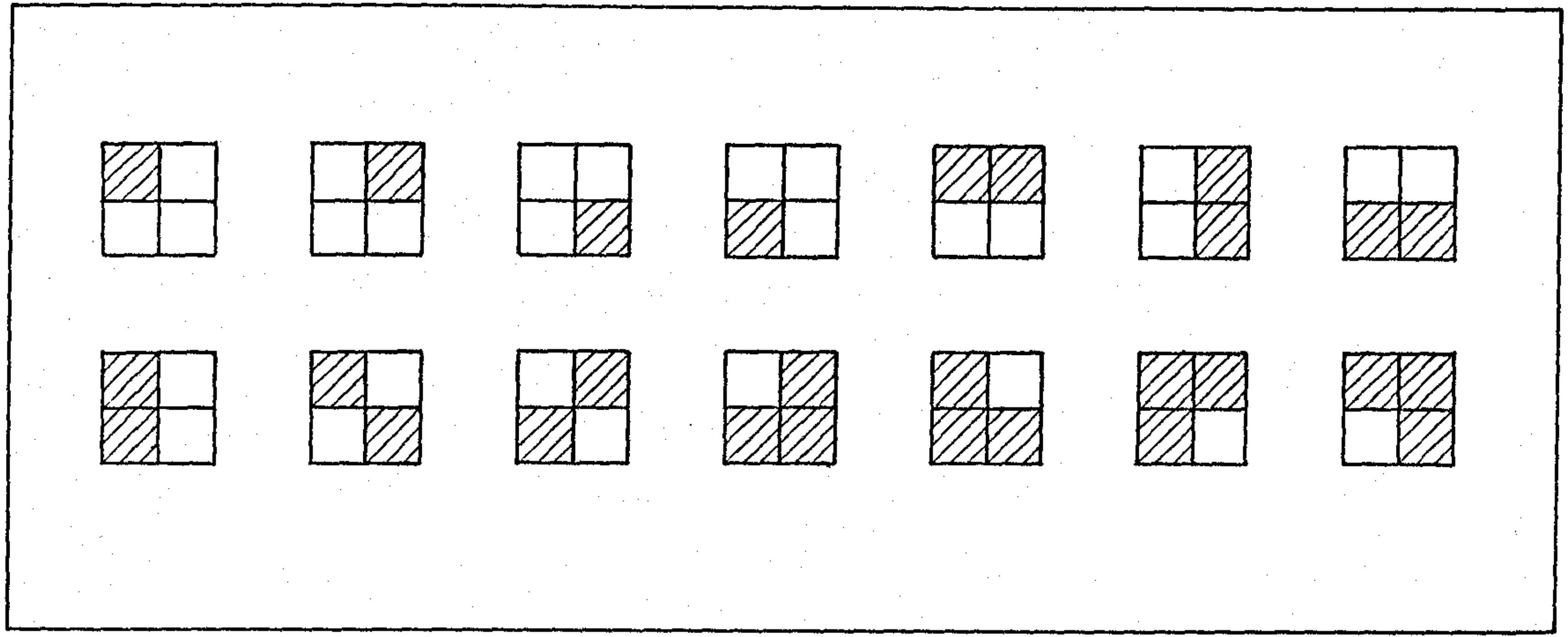


Fig. 7.3 - Possible locations of black pels in BPBs

Quantitative analysis of both schemes, as seen in the next two subsections, reveals that the former is the more efficient scheme.

7.3.1 Quantitative Analysis of Block Coding

Let:- $+$ represent quartic divisions

□ represent white primitives

■ represent black primitives

The number of bits required to code an initial picture block is:-

$$B = x.\ell_{+} + y.m_{\square} + z.n_{\blacksquare} \qquad (7.3)$$

where x, y, and z are the codewords assigned to quartic division, white primitives and black primitives respectively.

Since three symbols are considered, namely quartic divisions, white primitives and black primitives, the average amount of information [redacted] assigned to each x, y, and z respectively is $\log_2 3$ bits. Hence:-

$$H = (\ell_{+} + m_{\square} + n_{\blacksquare}) \log_2 3 \quad \text{bits} \qquad (7.4)$$

But $4^R\, C_p = m_{\square} + n_{\blacksquare}$ (from Eqn. (7.1))

and $\ell_{+} = \dfrac{4^R\, C_p - 1}{3}$ (from Eqn. (7.2))

Therefore,

$$H = \left[\frac{4^R\, C_p - 1}{3} + 4^R\, C_p \right] \log_2 3$$

$$= \left[\frac{4}{3} \cdot 4^R\, C_p - \frac{1}{3} \right] \log_2 3 \quad \text{bits} \qquad (7.5)$$

Eqn. (7.5) is the minimum number of bits required to code any particular IPB (assuming equal probabilities) using the D.F. coding algorithm(69). It is found that if the complexity of an IPB is less than approximately 0.47, data compression can be achieved using the above scheme.

Bits are now assigned to the 3 symbols where:-

(a) + = '10' □ = '00' ■ = '11'

(b) + = '0' □ = '10' ■ = '11'

(c) + = '10' □ = '0' ■ = '11'

Notice that the above codes are comma-free codes.

Code (a) is the simplest with constant length having no problem of synchronisation but obviously inherently redundant. The number of bits required to code an IPB will be:-

$$B_a = (\ell_+ + m_\square + n_\blacksquare) \cdot 2 \quad \text{bits} \tag{7.6}$$

Substituting Eqn. (7.1) and Eqn. (7.2), we get:-

$$B_a = 2 \cdot \left[\frac{4}{3} \cdot 4^R \, C_p - \frac{1}{3} \right] \quad \text{bits} \tag{7.7}$$

Code (b) is used when the probability of quartic division is greater than the probability of white primitives. (For the test documents used, the probability of black primitives is normally small.) Hence, the number of bits required to code an IPB is:-

$$B_b = \ell_+ + 2(m_\square + n_\blacksquare) \quad \text{bits} \tag{7.8}$$

Substituting Eqn. (7.1) and Eqn. (7.2), we get:-

$$B_b = \left[\frac{7}{3} \, . \, 4^R \, C_p - \frac{1}{3} \right] \quad \text{bits} \qquad (7.9)$$

Code (c) is used when the probability of white primitives exceeds that of quartic divisions. The number of bits required to code an IPB using this assignment procedure is:-

$$B_c = 2 \, \ell_+ + m_\square + 2 \, n_\blacksquare \quad \text{bits} \qquad (7.10)$$

$$= 2 \, \ell_+ + m_\square + n_\blacksquare + n_\blacksquare \quad \text{bits}$$

Substituting Eqn. (7.1) and Eqn. (7.2):-

$$B_c = 2 \left[\frac{4^R \, C_p - 1}{3} \right] + 4^R \, C_p + n_\blacksquare \qquad (7.11)$$

but from Eqn. (7.1),

$$C_p = \frac{m_\square}{4^R} + \frac{n_\blacksquare}{4^R}$$

$$\therefore \quad C_p = C_{p\square} + C_{p\blacksquare} \qquad (7.12)$$

where $C_{p\square}$ and $C_{p\blacksquare}$ are complexities due to the white and black primitives respectively.

From Eqn. (7.12), $\quad n_\blacksquare = 4^R \, C_{p\blacksquare} \qquad (7.13)$

Therefore, substituting Eqn. (7.13) into Eqn. (7.11), we get:-

$$B_c = 2 . \left[\frac{4^R \, C_p - 1}{3} \right] + 4^R \, C_p + 4^R \, C_{p\blacksquare}$$

Hence, $B_c = \left[\frac{5}{3} \, . \, 4^R \, C_p + 4^R \, C_{p\blacksquare} - \frac{2}{3} \right] \quad \text{bits} \qquad (7.14)$

7.3.2 Quantitative Analysis of Location Coding

In location coding, after successive quartic divisions, basic picture blocks (BPBs) of size 2 x 2 pels are reached and are coded according to the distribution of black pels within them. There are 14 possible locations as shown in Fig. 7.3.

Since ABLC uses three symbols, the amount of information [redacted] is $\log_2 3$ bits/symbol. Hence, the number of bits required to code SPBs greater than 2^2 is:-

$$H_{>2^2} = (\ell_{+>2^2} + m_{\square>2^2} + n_{\blacksquare>2^2}) \log_2 3 \quad \text{bits} \tag{7.15}$$

where the subscript $>2^2$ denotes SPBs greater than 2^2 and ℓ_+ is defined as the number of quartic divisions.

The minimum number of bits required to code the BPBs using location coding (assuming each location is equally probable) is:-

$$H_{=2^2} = \ell_{+=2^2} \cdot \log_2 3 + \ell_{+=2^2} \cdot \log_2 14 \quad \text{bits} \tag{7.16}$$

where the subscript $=2^2$ denotes SPBs equal to 2^2 and $\ell_{+=2^2} \cdot \log_2 3$ is the number of prefix bits.

Hence, the total minimum number of bits required to code an IPB of size 4^R is:-

$$H_T = H_{=2^2} + H_{>2^2} \tag{7.17}$$

$$= \ell_{+=2^2} (\log_2 3 + \log_2 14) + (\ell_{+>2^2} + m_{\square>2^2} + n_{\blacksquare>2^2}) . \log_2 3 \quad \text{bits}$$

where $\ell_{+=2^2}$ is the number of quartic divisions for subpicture blocks equal to 2^2 (i.e. the number of BPBs).

But $\ell_{+} = \ell_{+=2^2} + \ell_{+>2^2}$

$$\therefore \quad H_T = \ell_{+} . \log_2 3 + (m_{\square > 2^2} + n_{\blacksquare > 2^2}) \log_2 3 + \ell_{+=2^2} . \log_2 14 \qquad (7.18)$$

Substituting Eqn. (7.1) and Eqn. (7.2) into Eqn. (7.18), we get:-

$$H_T = \left(\frac{4^R C_p - 1}{3} + 4^R C_{p>2^2} \right) \log_2 3 + \ell_{+=2^2} . \log_2 14 \qquad (7.19)$$

where $C_{p>2^2}$ is the complexity due to primitives in subpicture blocks greater than 2^2.

The number of primitives in subpicture blocks equal to 2^2 is:-

$$m_{\square = 2^2} + n_{\blacksquare = 2^2} = \ell_{+=2^2} . 4 \qquad (7.20)$$

The complexity due to primitives in subpicture blocks equal to 2^2 is:-

$$C_{p=2^2} = \frac{\ell_{+=2^2} . 4}{4^R}$$

$$\therefore \quad \ell_{+=2^2} = \frac{4^R C_{p=2^2}}{4} \qquad (7.21)$$

Substituting Eqn. (7.21) into Eqn. (7.19), we get:-

$$H_T = \left(\frac{4^R C_p - 1}{3} + 4^R C_{p>2^2} \right) \log_2 3 + \frac{4^R C_{p=2^2}}{4} . \log_2 14$$

but $\quad C_p = C_{p>2^2} + C_{p=2^2}$

$$\therefore \qquad C_{p>2^2} = C_p - C_{p=2^2} \tag{7.22}$$

Hence,

$$H_T = \left(\frac{4^R C_p - 1}{3} + 4^R C_p \right) \log_2 3 + 4^R C_{p=2^2} \left(\frac{\log_2 14}{4} - \log_2 3 \right)$$

$$= \left(\frac{4}{3} . 4^R C_p - \frac{1}{3} \right) \log_2 3 + 4^R C_{p=2^2} (- 0.633)$$

Since $\left(\frac{4}{3} . 4^R C_p - \frac{1}{3} \right) \log_2 3 = H$ (see Eqn. (7.5))

it follows that :-

$$H_T = H - 0.633 \; 4^R C_{p=2^2} \qquad \text{bits} \tag{7.23}$$

where $C_{p=2^2}$ is the complexity due to primitives in the BPBs and $H = \left(\frac{4}{3} . 4^R C_p - \frac{1}{3} \right) \log_2 3$ bits is the minimum number of bits required to code an IPB using the D.F. coding algorithm. It is thus shown, from Eqn. (7.23), that the bit-rate for ABLC is less than for D.F. coding by an amount equal to $0.633 . 4^R C_{p=2^2}$, i.e. the former is the more efficient scheme.

Fig. 7.4 shows the variation of the number of bits with complexity for the two coding algorithms. An IPB of size 8 x 8 is used and $C_{p=2^2}$ is the third variable shown in the shaded area. Although the complexities of some blocks are the same, the number of bits required to code an IPB need not necessarily be the same, due to variations in $C_{p=2^2}$. Clearly, ABLC has a superior performance.

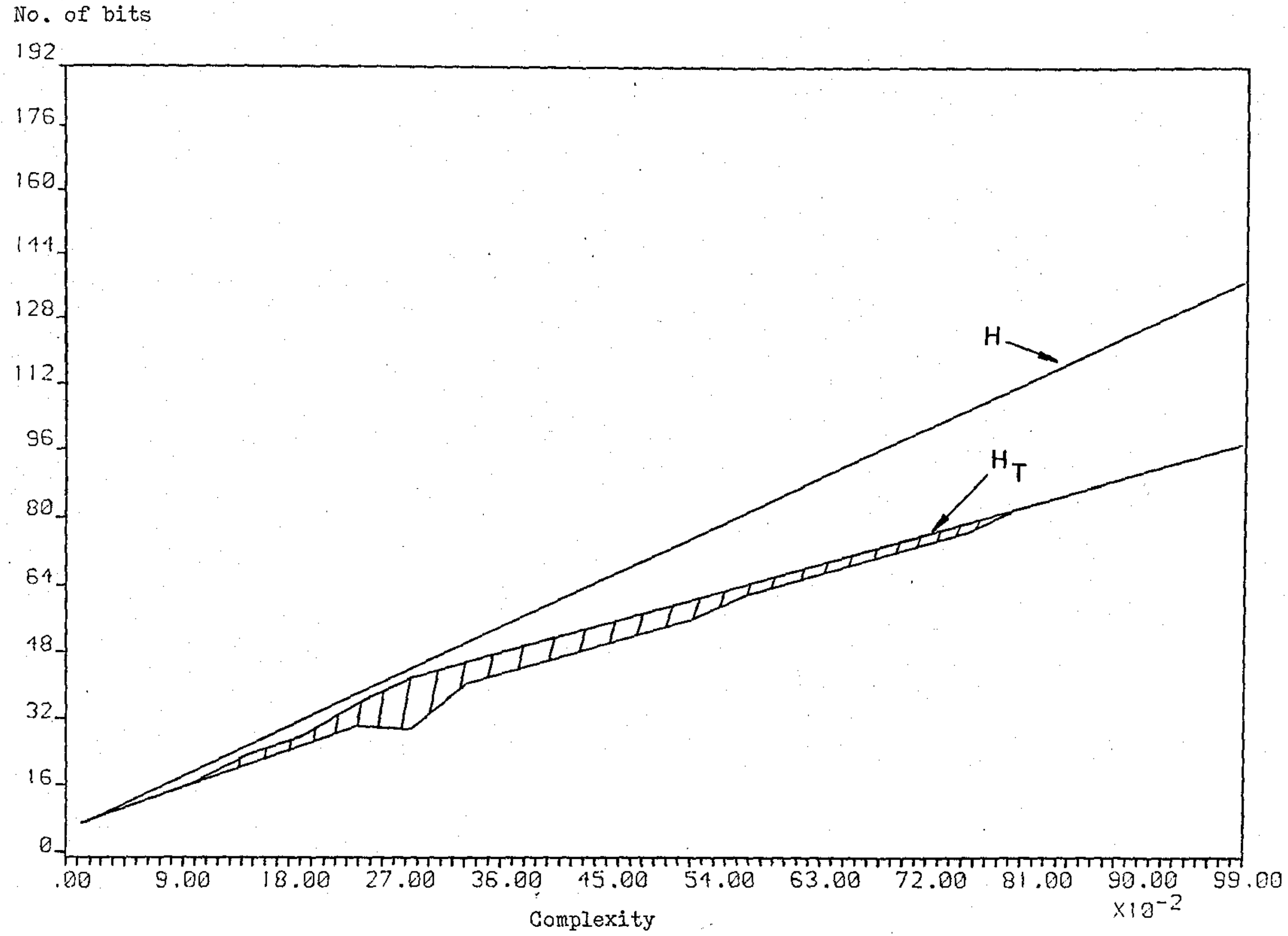


Fig. 7.4 - Variations of number of bits with complexity

Let us now consider that the symbols (+, □, ■) and the location of black pels in the basic picture block have unequal probabilities. Huffman coding can then be used to code the location of black pels in the BPBs.

Let A be a discrete source comprising the Q independent possible locations of the black pels, and let P_k denote the probability of the kth location of the black pel, a_k. The average word length required to code the location of black pels in the BPBs will be:-

$$\bar{c} = \sum_{k=1}^{Q} r_k P_k \tag{7.24}$$

where r_k is the length of binary word that encodes a_k.
The entropy of the Q possible locations of the black pels is:-

$$H_{LC} = - \sum_{k=1}^{Q} P_k \log_2 P_k \tag{7.25}$$

If probabilities are assigned to the 3 symbols where

X_1 = probability of quartic division

X_2 = probability of white primitives

X_3 = probability of black primitives

then the entropy of the symbol is:-

$$H_{sym} = - \sum_{k=1}^{3} X_k \log_2 X_k \tag{7.26}$$

Therefore, the entropy of an IPB from Eqn. (7.19) is:-

$$H_H = \left(\frac{4^R\ C_p - 1}{3} + 4^R\ C_{p>2^2}\right) H_{sym} + \ell_{+=2^2} \cdot H_{LC} \qquad (7.27)$$

Let us now assign bits to the three symbols and use the Huffman code to code the locations of the black pels in the BPBs:-

(a) $+$ = '10' $\square_{>2^2}$ = '00' $\blacksquare_{>2^2}$ = '11' location= Huffman code

From Eqn. (7.24) and Eqn. (7.27), the number of bits required to code an IPB using this coding procedure is:-

$$B_{Ha} = \left(\frac{4^R\ C_p - 1}{3} + 4^R\ C_{p>2^2}\right).2 + \ell_{+=2^2} \cdot \sum_{k=1}^{Q} r_k\ P_k \qquad (7.28)$$

Substituting Eqn. (7.21) and Eqn. (7.22) into Eqn. (7.28), it can be shown that:-

$$B_{Ha} = B_a - 4^R\ C_{p=2^2} \left[2 - \frac{\sum_{k=1}^{Q} r_k\ P_k}{4} \right] \qquad (7.29)$$

where $B_a = \left(\frac{4}{3} \cdot 4^R\ C_p - \frac{1}{3}\right).2$ (see Eqn. (7.7))

(b) $+$ = '0' $\square_{>2^2}$ = '10' $\blacksquare_{>2^2}$ = '11' location= Huffman code

From Eqn. (7.24) and Eqn. (7.27), the number of bits required to code an IPB is:-

$$B_{Hb} = \left(\frac{4^R\ C_p - 1}{3}\right) + 2.4^R\ C_{p>2^2} + \ell_{+=2^2} \cdot \sum_{k=1}^{Q} r_k\ P_k \qquad (7.30)$$

Substitution of Eqn. (7.21) and Eqn. (7.22) into Eqn. (7.30) leads to:-

$$B_{Hb} = \left(\frac{7}{3} \, . \, 4^R \, C_p - \frac{1}{3} \right) + 4^R \, C_{p=2^2} \left[\frac{\sum_{k=1}^{Q} r_k \, P_k}{4} - 2 \right]$$

$$= B_b - 4^R \, C_{p=2^2} \left[2 - \frac{\sum_{k=1}^{Q} r_k \, P_k}{4} \right] \qquad (7.31)$$

where $B_b = \left(\frac{7}{3} \, . \, 4^R \, C_p - \frac{1}{3} \right)$ (see Eqn. (7.9))

(c) $+$ = '10' $\square_{>2^2}$ = '0' $\blacksquare_{>2^2}$ = '11' location= Huffman code

The number of bits required to code an IPB using this procedure is:-

$$B_{Hc} = 2 \, . \, \ell_{+} + m_{\square > 2^2} + 2 \, n_{\blacksquare > 2^2} + \ell_{+=2^2} \, . \sum_{k=1}^{Q} r_k \, P_k \qquad (7.32)$$

$$= 2 \, . \, \ell_{+} + (m_{\square > 2^2} + n_{\blacksquare > 2^2}) + n_{\blacksquare > 2^2} + \ell_{+=2^2} \, . \sum_{k=1}^{Q} r_k \, P_k$$

Substituting Eqn. (7.2), Eqn. (7.12) and Eqn. (7.21), we get:-

$$B_{Hc} = 2 \left(\frac{4^R \, C_p - 1}{3} \right) + 4^R \, C_{p > 2^2} + 4^R \, C_{p\blacksquare > 2^2} + \frac{4^R \, C_{p=2^2}}{4} \sum_{k=1}^{Q} r_k \, P_k \qquad (7.33)$$

where $C_{p > 2^2}$ is the complexity due to primitives greater than 2^2

and $C_{p\blacksquare > 2^2} = \dfrac{n_{\blacksquare > 2^2}}{4^R}$ is the complexity due to black primitives greater than 2^2.

But $$C_{p\blacksquare} = C_{p\blacksquare > 2^2} + C_{p\blacksquare = 2^2} \qquad (7.34)$$

$$\therefore \quad C_{p\blacksquare > 2^2} = C_{p\blacksquare} - C_{p\blacksquare = 2^2}$$

$$\therefore B_{Hc} = 2\left(\frac{4^R C_p - 1}{3}\right) + 4^R \left(C_p - C_{p=2^2}\right) + 4^R \left(C_{p\blacksquare} - C_{p\blacksquare = 2^2}\right) + \frac{4^R C_{p=2^2}}{4} \sum_{k=1}^{Q} r_k P_k \qquad (7.35)$$

Simplifying Eqn.(7.35) further leads to:-

$$B_{Hc} = B_c - 4^R C_{p=2^2} \left[1 - \frac{\sum_{k=1}^{Q} r_k P_k}{4} \right] - 4^R C_{p\blacksquare = 2^2} \qquad (7.36)$$

where $B_c = \left(\frac{5}{3} \cdot 4^R C_p + 4^R C_{p\blacksquare} - \frac{2}{3} \right)$ (see Eqn. (7.14))

and $C_{p\blacksquare = 2^2}$ is the complexity due to black primitives in the BPBs.

Using the bit assignment procedures (a), (b) and (c) to code quartic divisions, white and black primitives greater than 2^2 and employing the Huffman code to code the location of black pels in the BPBs, the variation of the number of bits assigned to an IPB of size 8 x 8 (from Eqn. (7.29), Eqn. (7.31) and Eqn. (7.36)) can be plotted as shown in Fig. 7.5(a), 7.5(b) and 7.5(c) respectively. The average codeword length ($\overline{c}$ = 3.474 bits) is obtained for an ensemble of 7 CCITT documents and their prediction error domains (see next Section, Table 7.1). It is obvious from these graphs that ABLC is more efficient then D.F. coding for all cases.

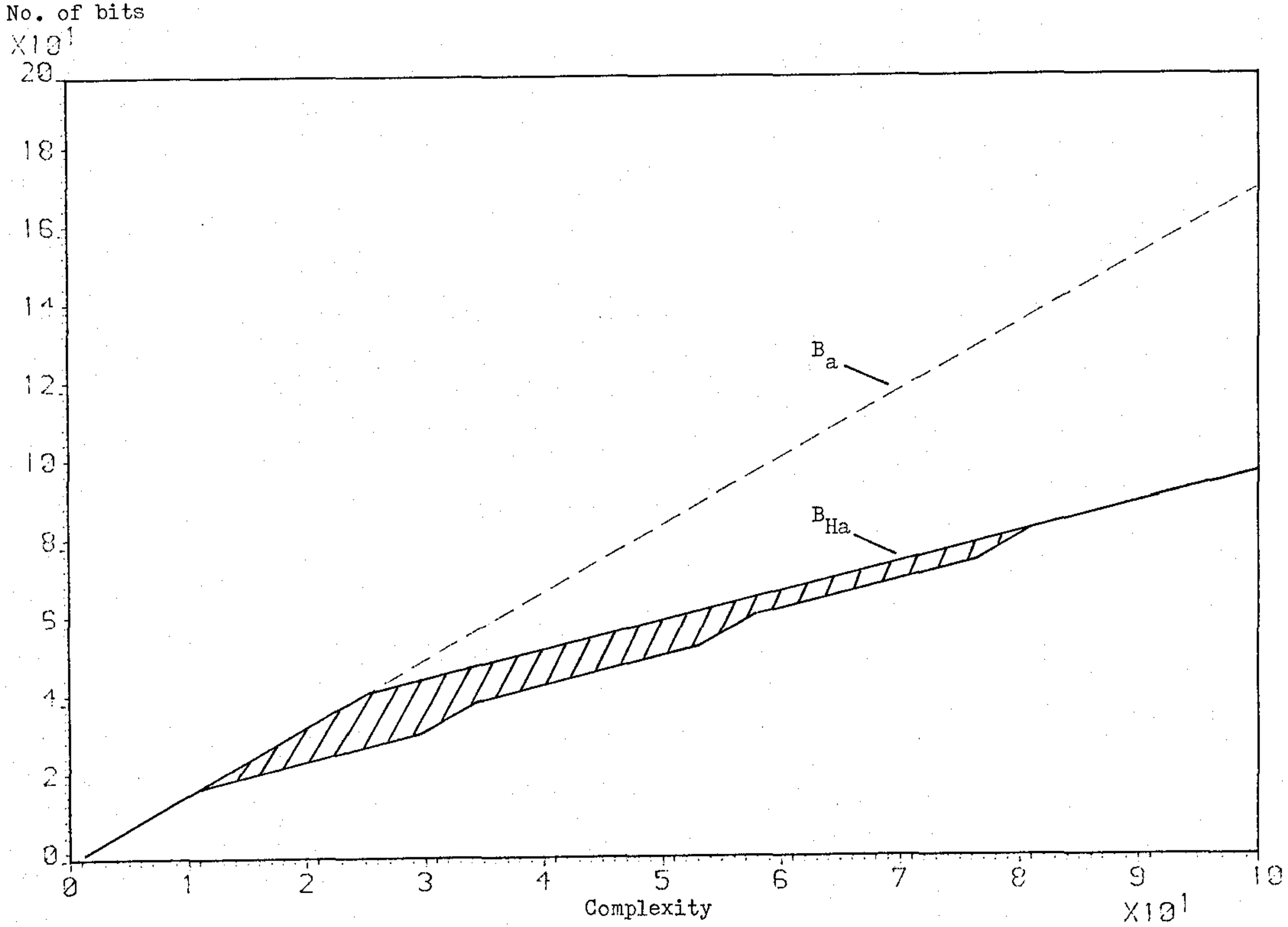


Fig. 7.5(a) - Variations of the number of bits with complexity for bit assignment (a) (See Eqn. (7.29)

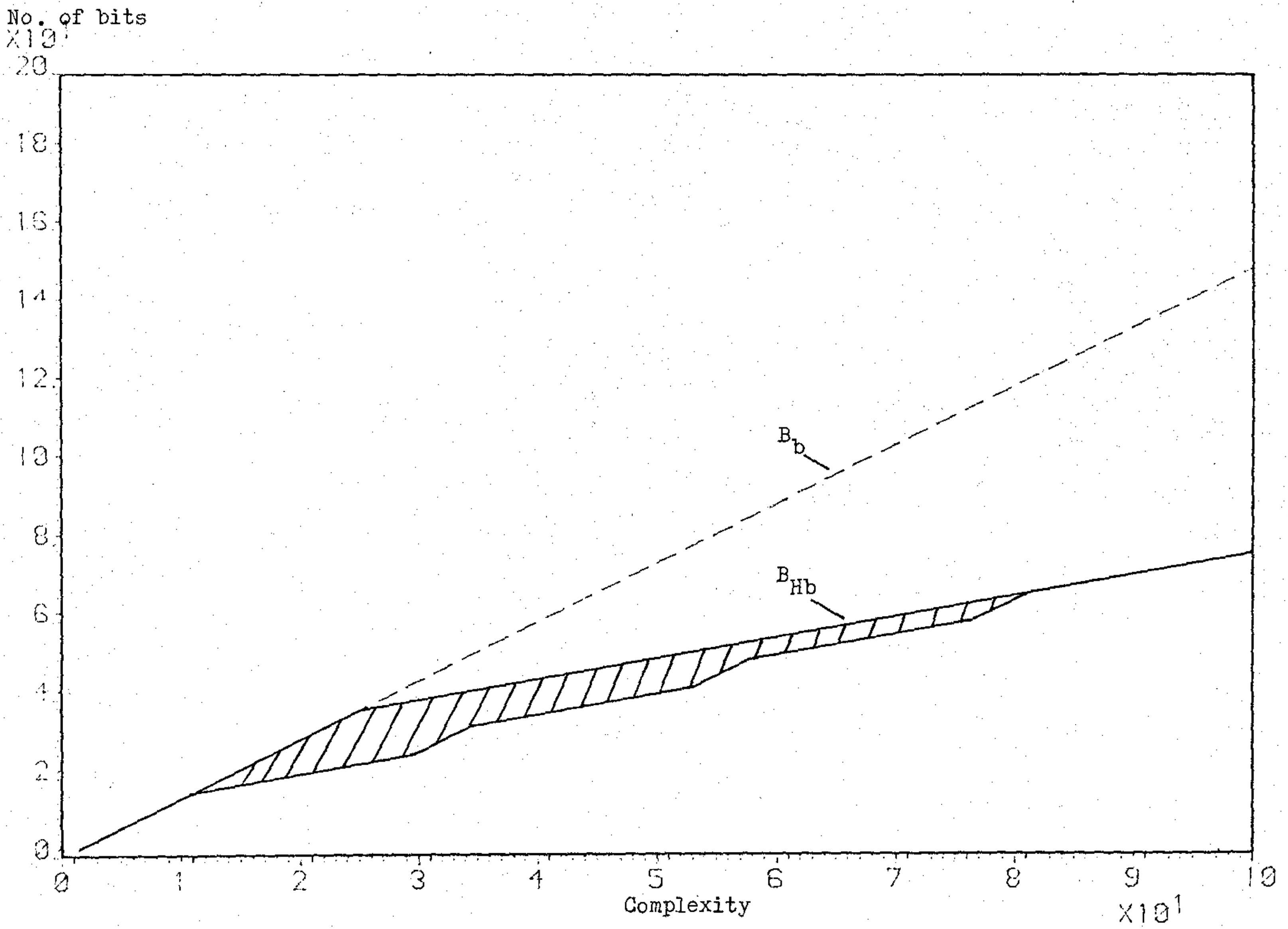


Fig. 7.5(b) - Variation of number of bits with complexity for bit assignment (b) (See Eqn. (7.31))

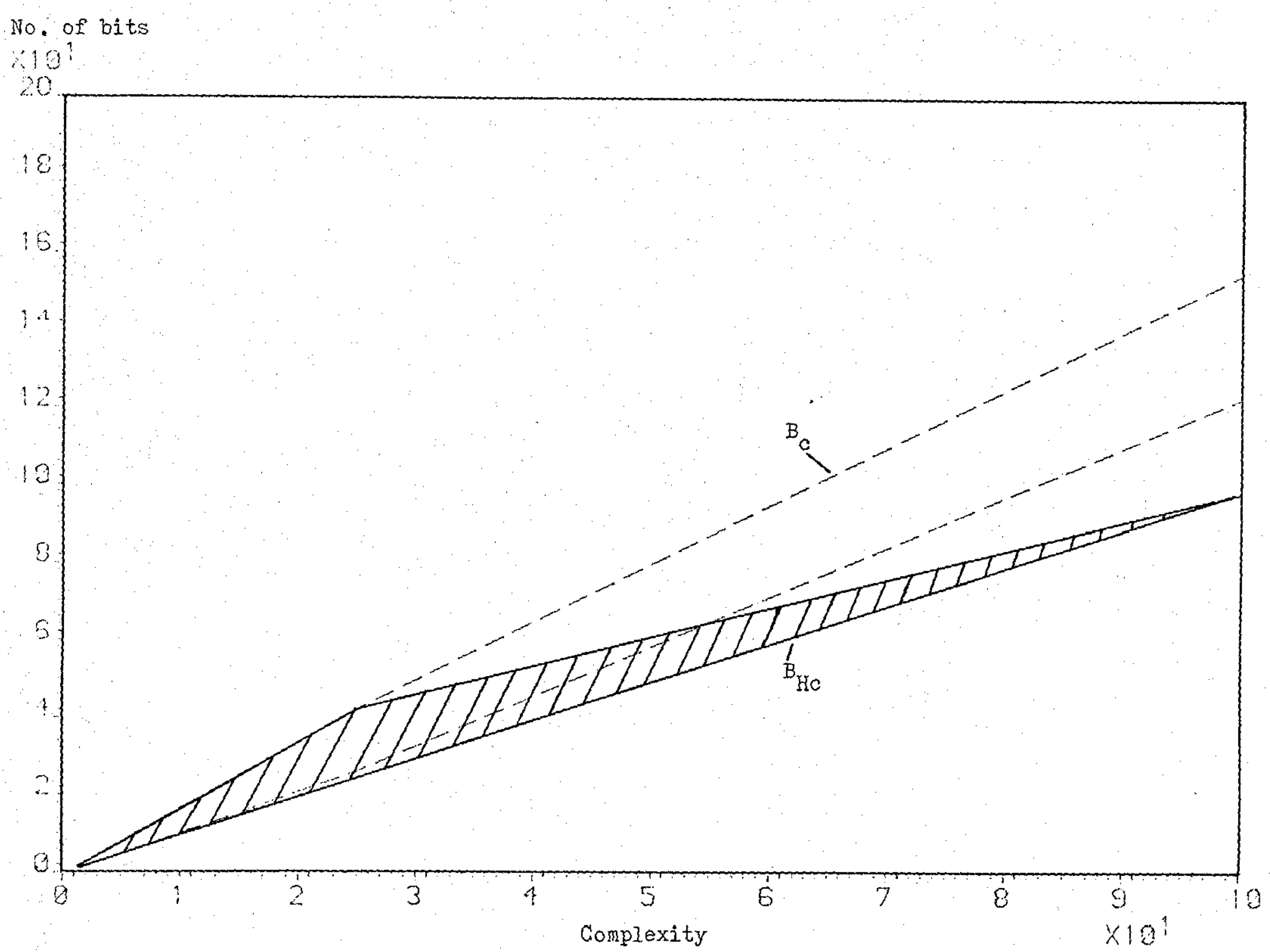


Fig. 7.5(c) - Variation of number of bits with complexity for bit assignment (c) (See Eqn. (7.36))

7.4. RESULTS OF COMPUTER SIMULATIONS

In this section, the results obtained from computer simulations are summarised. Four variations of ABLC are examined, and in each case the documents are subsampled to half their resolution in both directions prior to coding. The original CCITT documents were scanned at a resolution of 8 pels/mm both horizontally and vertically corresponding to 1728 pels/line and 2376 lines. We define:-

$$\text{Bit-rate} = \frac{1}{CR} = \frac{\text{No. of bits transmitted}}{\text{Total no. of pels in original document}} \qquad (7.37)$$

Since ABLC uses three symbols, namely white primitives greater than 2^2, black primitives greater than 2^2 and quartic divisions, two bit assignment procedures were employed. They are:-

Ⓐ $\square_{>2^2}$ = '0' $\blacksquare_{>2^2}$ = '11' + = '10' location=Huffman code

Ⓑ $\square_{>2^2}$ = '10' $\blacksquare_{>2^2}$ = '11' + = '0' location=Huffman code

The codes used for both bit assignment procedures are comma free codes. Bit assignment Ⓐ results from the fact that the frequency of white primitives greater than 2^2 is highest, whereas in bit assignment Ⓑ, the probability of quartic division is greatest. To code the location of the black pels in the BPBs, Huffman coding is used, where the probabilities of such locations are obtained from measurements on an ensemble of 7 subsampled CCITT documents. The Huffman codes constructed are shown in Table 7.1. For all the systems simulated,

BPB	Subsampled documents	Prediction error domain
■□ □□	1001	11
□■ □□	0101	010
□□ □■	1000	001
□□ ■□	0100	000
■■ □□	011	1001
□■ □■	001	1000
□□ ■■	101	1011
■□ ■□	000	1010
■□ □■	110010	01111
□■ ■□	110011	01101
□■ ■■	1101	011101
■□ ■■	1110	011000
■■ ■□	1111	011100
■■ □■	11000	011001

Table 7.1 - Huffman code for ABLC

three IPBs were used, 8 x 8, 16 x 16 and 32 x 32. Block sizes of higher degree were not used and it is envisaged that no further substantial bit-rate reduction can be achieved in this way.

7.4.1 System A

The basic configuration of System A is shown in Fig. 7.6.

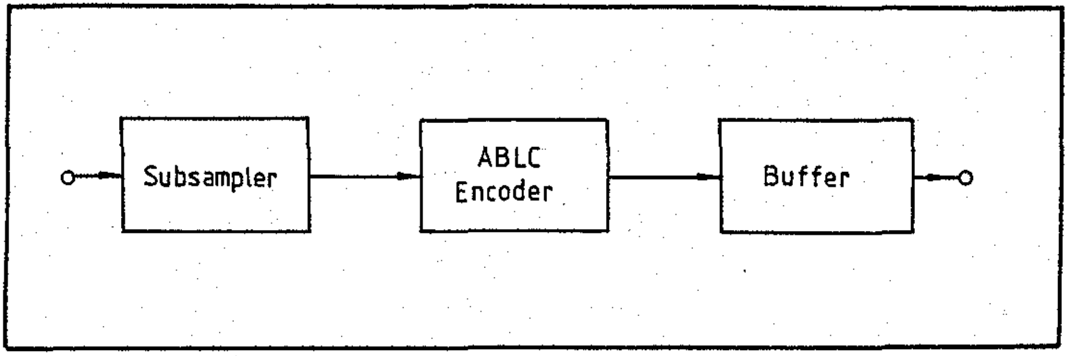


Fig. 7.6 - System A

The subsampled image is directly coded using the ABLC algorithm with the location of black pels in the BPBs coded using the Huffman code of Table 7.1. The compression ratios for bit assignment procedures Ⓐ and Ⓑ are shown in Table 7.2(a) and 7.2(b) respectively. It can be seen from the Tables that as the block size increases, the compression ratios also increase . Bit assignment Ⓐ performs better than bit assignment Ⓑ for block size 8 x 8 but the converse is true for block sizes 16 x 16 and 32 x 32. For bigger block sizes, the probability of quartic divisions is higher than that of white primitives greater than 2^2, resulting in higher CRs for block sizes 16 x 16 and 32 x 32.

Doc. No.	8 x 8	16 x 16	32 x 32
1	40.27	43.76	44.54
2	50.12	55.83	57.06
3	23.30	23.86	23.87
4	11.61	11.59	11.57
5	23.31	23.91	23.97
6	34.68	36.66	36.93
7	14.47	14.45	14.42

(a) Bit assignment Ⓐ

Doc. No.	8 x 8	16 x 16	32 x 32
1	36.97	44.38	46.03
2	44.59	56.12	59.39
3	23.09	24.90	25.20
4	12.12	12.34	12.37
5	23.07	24.95	25.31
6	32.69	37.50	38.53
7	14.70	15.05	15.11

(b) Bit assignment Ⓑ

Table 7.2 - Compression ratios for System A

7.4.2 System B

The block diagram of System B is shown in Fig. 7.7.

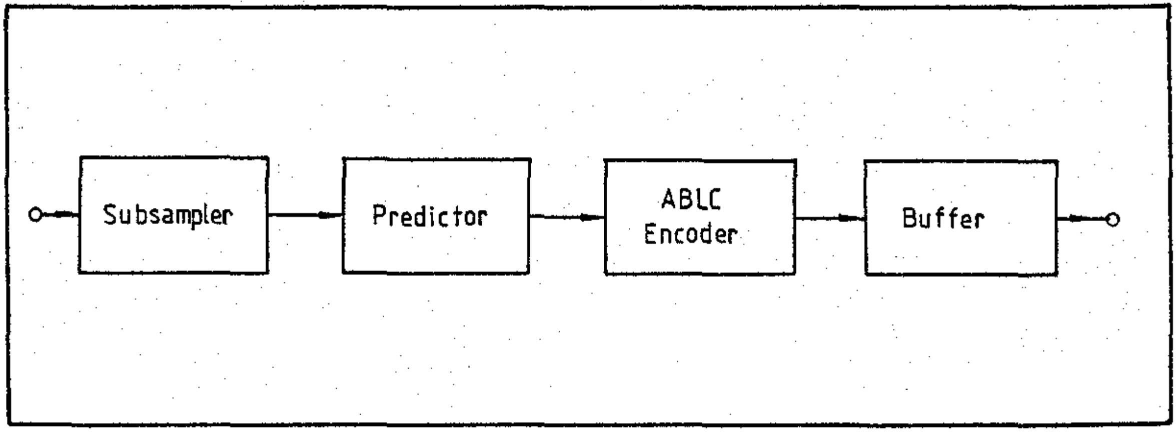


Fig. 7.7 - System B

In System B, the 7th-order Markov model predictor(60) (similar to the one used in Chapter VI) is employed on the subsampled image, hence generating a prediction error domain. The predictor is optimised over an ensemble of 7 CCITT documents and the Huffman code constructed to code the location of the black pels in the BPBs based on these documents are shown in Table 7.1. The predictor is included with the aim of reducing the complexity of the documents, thereby increasing the compression ratios (see Fig. 7.4). As can be seen in Tables 7.3(a) and 7.3(b), the performance of ABLC is improved by the inclusion of the 7th-order Markov model predictor. Again, bigger block sizes provide higher compression ratios.

7.4.3 System C

The development of System C is based on System A and System B where

Doc. No.	8 x 8	16 x 16	32 x 32
1	42.43	46.29	47.15
2	53.70	60.36	61.88
3	31.33	32.70	32.82
4	12.71	12.70	12.68
5	28.38	29.47	29.61
6	53.21	59.43	60.54
7	14.36	14.33	14.30

(a) Bit assignment Ⓐ

Doc. No.	8 x 8	16 x 16	32 x 32
1	37.91	45.70	47.87
2	45.94	58.35	61.89
3	29.37	32.86	33.52
4	12.95	13.24	13.29
5	26.88	29.76	30.35
6	45.85	57.79	60.86
7	14.52	14.86	14.91

(b) Bit assignment Ⓑ

Table 7.3 - Compression ratios for System B

it is observed that higher compression ratios can be obtained by the inclusion of the 7th-order Markov model predictor. Also, as can be seen in Fig. 7.4, the bit-rate is dependent on the complexity of the picture block. The application of the 7th-order Markov model predictor, in which the present pel is predicted based on seven previous pels, reduces the complexity of most picture blocks but does increase the complexity of certain picture blocks. Thus, the complexity of the prediction error domain is computed and then compared with the complexity of the subsampled image without prediction. The picture block with the lower complexity is coded using ABLC and then transmitted together with the necessary infomation bits. Fig. 7.8 shows the basic arrangement of System C.

Fig. 7.9(a) and Fig. 7.9(b) respectively show the complexity maps of the subsampled image and of the prediction error domain for 64 blocks of the same picture. An IPB of 8 x 8 is used. Notice that the complexity of picture blocks in the prediction error domain is generally lower, and such blocks therefore require fewer bits for coding. However, their complexity is sometimes higher than that of the subsampled image, and in such a case the latter is transmitted instead. Fig. 7.9(c) shows the resulting complexity map in which the IPB (either the subsampled image or its prediction error domain) with the lower complexity is chosen and coded using ABLC. Figs. 7.10(a) and 7.10(b), respectively, illustrate the bit allocation maps for bit assignment procedures Ⓐ and Ⓑ of the resulting complexity map of Fig. 7.9(c). Notice that the number of bits assigned to an IPB depends very much on its complexity, confirming the mathematical relationships of Eqn. (7.31) and Eqn. (7.36).

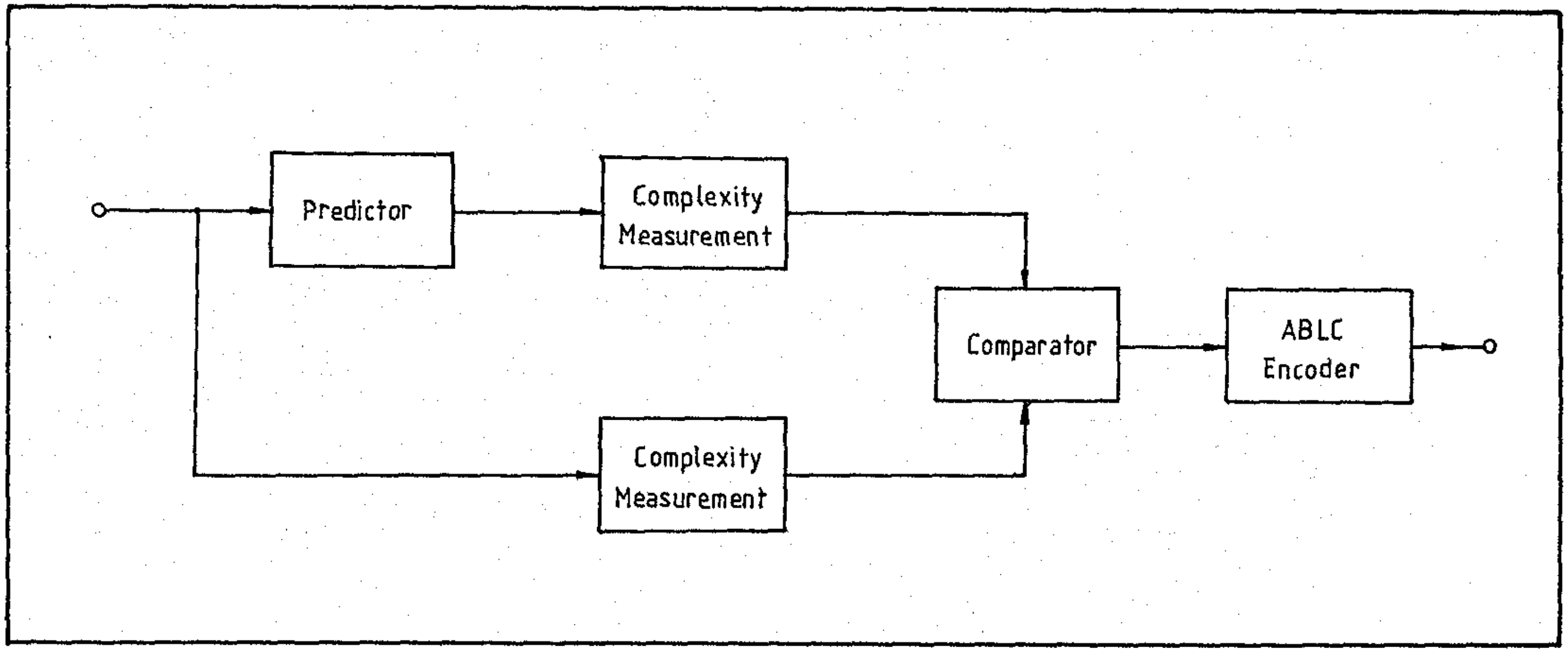
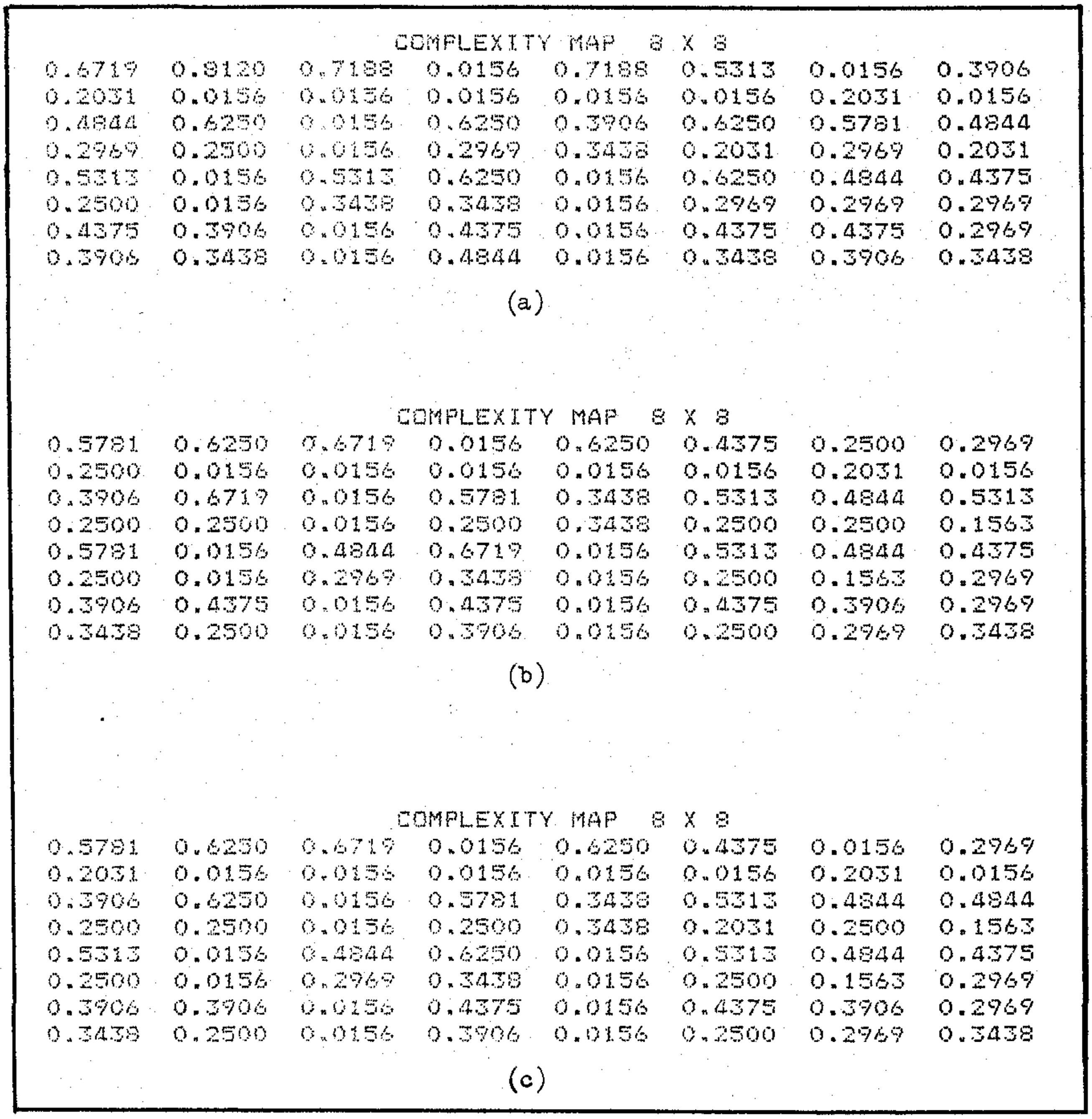


Fig. 7.8 - System C (ABLC encoder with complexity as a criterion)



```
                              COMPLEXITY MAP   8 X 8
0.6719   0.8120   0.7188   0.0156   0.7188   0.5313   0.0156   0.3906
0.2031   0.0156   0.0156   0.0156   0.0156   0.0156   0.2031   0.0156
0.4844   0.6250   0.0156   0.6250   0.3906   0.6250   0.5781   0.4844
0.2969   0.2500   0.0156   0.2969   0.3438   0.2031   0.2969   0.2031
0.5313   0.0156   0.5313   0.6250   0.0156   0.6250   0.4844   0.4375
0.2500   0.0156   0.3438   0.3438   0.0156   0.2969   0.2969   0.2969
0.4375   0.3906   0.0156   0.4375   0.0156   0.4375   0.4375   0.2969
0.3906   0.3438   0.0156   0.4844   0.0156   0.3438   0.3906   0.3438
```

(a)

```
                              COMPLEXITY MAP   8 X 8
0.5781   0.6250   0.6719   0.0156   0.6250   0.4375   0.2500   0.2969
0.2500   0.0156   0.0156   0.0156   0.0156   0.0156   0.2031   0.0156
0.3906   0.6719   0.0156   0.5781   0.3438   0.5313   0.4844   0.5313
0.2500   0.2500   0.0156   0.2500   0.3438   0.2500   0.2500   0.1563
0.5781   0.0156   0.4844   0.6719   0.0156   0.5313   0.4844   0.4375
0.2500   0.0156   0.2969   0.3438   0.0156   0.2500   0.1563   0.2969
0.3906   0.4375   0.0156   0.4375   0.0156   0.4375   0.3906   0.2969
0.3438   0.2500   0.0156   0.3906   0.0156   0.2500   0.2969   0.3438
```

(b)

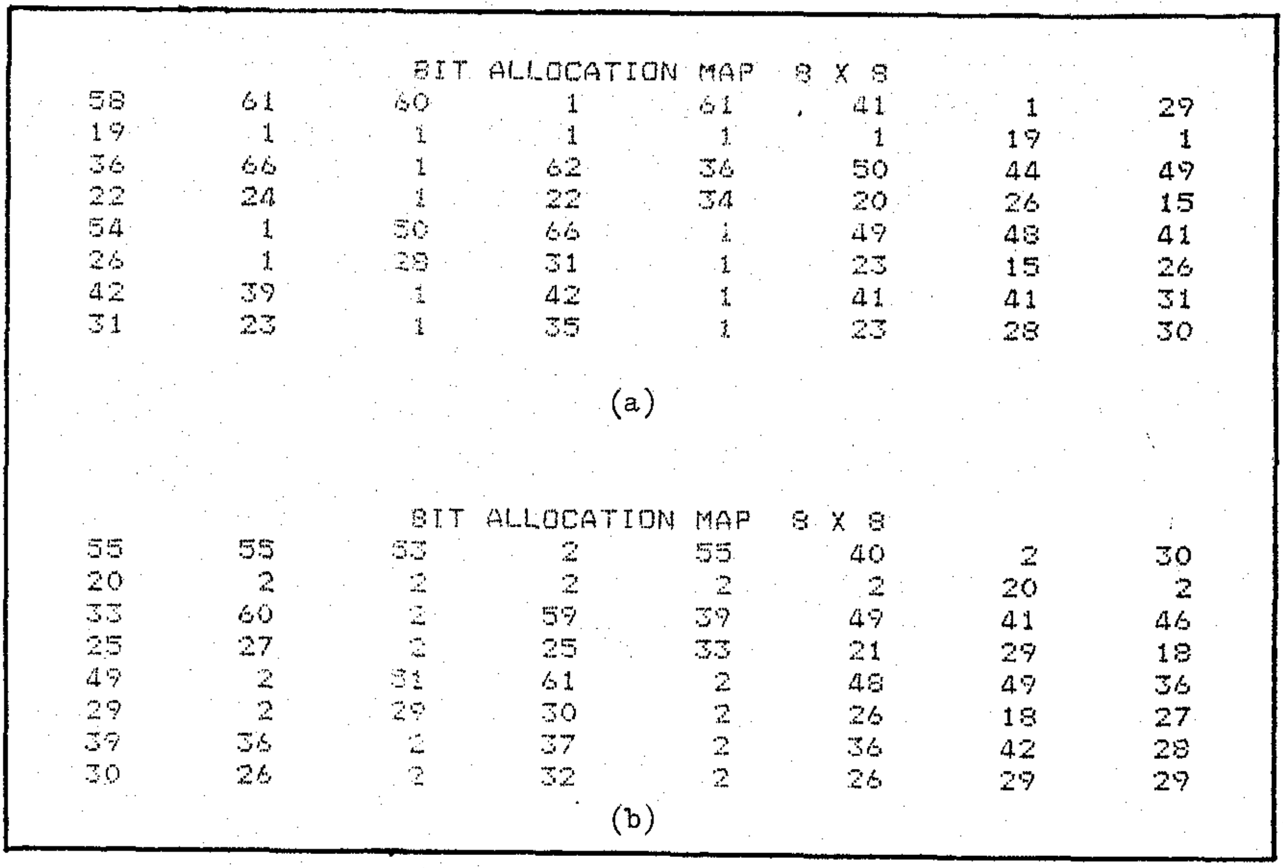
```
                              COMPLEXITY MAP   8 X 8
0.5781   0.6250   0.6719   0.0156   0.6250   0.4375   0.0156   0.2969
0.2031   0.0156   0.0156   0.0156   0.0156   0.0156   0.2031   0.0156
0.3906   0.6250   0.0156   0.5781   0.3438   0.5313   0.4844   0.4844
0.2500   0.2500   0.0156   0.2500   0.3438   0.2031   0.2500   0.1563
0.5313   0.0156   0.4844   0.6250   0.0156   0.5313   0.4844   0.4375
0.2500   0.0156   0.2969   0.3438   0.0156   0.2500   0.1563   0.2969
0.3906   0.3906   0.0156   0.4375   0.0156   0.4375   0.3906   0.2969
0.3438   0.2500   0.0156   0.3906   0.0156   0.2500   0.2969   0.3438
```

(c)

Fig. 7.9(a) - Complexity map of subsampled data

(b) - Complexity map of its prediction error domain

(c) - Resulting complexity map in which IPB of lower complexity is taken



```
                    BIT ALLOCATION MAP  8 X 8
 58        61        60         1        61        41         1        29
 19         1         1         1         1         1        19         1
 36        66         1        62        36        50        44        49
 22        24         1        22        34        20        26        15
 54         1        50        66         1        49        48        41
 26         1        28        31         1        23        15        26
 42        39         1        42         1        41        41        31
 31        23         1        35         1        23        28        30
```

(a)

```
                    BIT ALLOCATION MAP  8 X 8
 55        55        53         2        55        40         2        30
 20         2         2         2         2         2        20         2
 33        60         2        59        39        49        41        46
 25        27         2        25        33        21        29        18
 49         2        31        61         2        48        49        36
 29         2        29        30         2        26        18        27
 39        36         2        37         2        36        42        28
 30        26         2        32         2        26        29        29
```

(b)

Fig. 7.10 - Bit allocation maps for (a) bit assignment Ⓐ and (b) bit assignment Ⓑ of resulting complexity map of Fig. 7.9(c)

Tables 7.4(a) and 7.4(b) contain the compression ratios of System C for bit assignment procedures Ⓐ and Ⓑ . It is observed that for an IPB of size 8 x 8, the performance of System C is generally poorer than that of System A and of System B. This is due to the extra overhead information required to inform the receiver whether the subsampled data or its corresponding prediction error domain is being transmitted. For an IPB of size 16 x 16, compression ratios for System C are better than for System A but worse than for System B (except for document No. 7). Again, the transmission of extra side information is responsible for this slightly lower performance. For an IPB of size 32 x 32, System C outperforms both System A and System B, since the number of bits assigned to the IPBs using numerical complexity as the criterion is low enough to compensate for the extra overhead information and still allow superior performance for the same block size. As can be seen from Tables 7.4(a) and 7.4(b), bit assignment procedure Ⓑ generally performs better than bit assignment procedure Ⓐ , purely because the number of quartic divisions is highest compared with the number of white and black primitives greater than 2^2.

7.4.4 System D

For System D, shown in Fig. 7.11, ABLC is applied to an IPB of both the subsampled image and of its prediction error domain. The block which is assigned the lower number of bits is then transmitted, together with the necessary information side information.

Doc. No.	8 x 8	16 x 16	32 x 32
1	38.04	45.54	47.19
2	47.13	60.02	62.77
3	30.05	32.95	33.48
4	12.69	12.78	12.68
5	26.86	29.42	29.65
6	46.32	58.31	60.49
7	14.73	14.93	14.72

(a) Bit assignment Ⓐ

Doc. No.	8 x 8	16 x 16	32 x 32
1	34.12	44.92	48.08
2	41.03	58.48	63.77
3	28.03	33.10	34.28
4	12.77	13.85	13.31
5	25.27	29.64	30.46
6	40.36	56.59	60.75
7	14.64	15.34	15.30

(b) Bit assignment Ⓑ

Table 7.4 - Compression ratios for System C

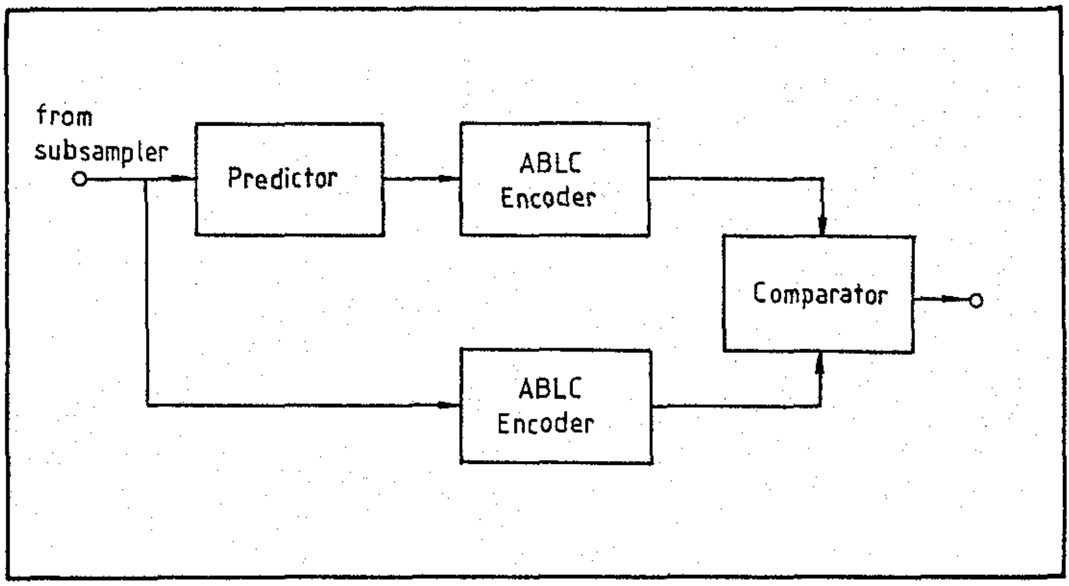


Fig. 7.11 - System D

As can be seen from Tables 7.5(a) and 7.5(b), the performance of System D is only marginally better than that of System C. This is to be expected because in System C, although the bit-rate is dependent on complexity, sometimes lower complexity does not necessarily mean that the number of bits assigned to an IPB is also lower. In some cases, fewer bits are assigned to IPBs with higher complexities, and this is due to variation in parameters such as $C_{p=2^2}$ and $C_{p\blacksquare=2^2}$ of Eqn. (7.31) and Eqn. (7.36) (i.e. complexity due to primitives in the BPBs). These are shown graphically as shaded areas in Fig. 7.5(b) and Fig. 7.5(c) respectively.

7.4.5 Discussion of Results

Comparing all the different configurations of ABLC, it is generally observed that it performs extremely well for graphics-type documents

Doc. No.	8 x 8	16 x 16	32 x 32
1	38.13	45.64	47.36
2	47.41	60.43	63.53
3	30.12	33.06	33.56
4	12.74	12.81	12.72
5	26.96	29.48	29.73
6	46.41	58.42	60.60
7	14.79	14.95	14.74

(a) Bit assignment Ⓐ

Doc. No.	8 x 8	16 x 16	32 x 32
1	34.28	45.09	48.19
2	41.25	58.73	64.06
3	28.16	33.29	34.40
4	12.86	13.35	13.35
5	25.43	29.75	30.53
6	40.49	56.73	60.86
7	14.75	15.41	15.34

(b) Bit assignment Ⓑ

Table 7.5 - Compression ratios for System D

(e.g. document Nos. 2 and 6) and still maintains significantly high CRs for alphanumeric type documents. The high CRs are to be expected, and this can be attributed to preprocessing by subsampling, where a 4:1 sample rate reduction is initially achieved. Notice that, for an IPB of size 32 x 32, System D gives the best performance, though only marginally compared with System C. System D is superior because it only transmits IPBs (either of the subsampled image or of its prediction error domain) which are assigned fewer bits. When an IPB of size 8 x 8 is used, System A and System B give higher CRs as compared with System C and System D. This is due to the one bit overhead information required in the latter case to inform the receiver whether the subsampled data or its prediction error domain is being transmitted. The introduction of the 7th-order Markov model predictor reduces the complexity of most blocks, and it is apparent from System B that its inclusion increases the CRs dramatically. System C takes complexity into account by transmitting IPBs with lower complexity, thereby improving the performance when compared with System A and System B for a block size of 32 x 32. It is expected that the use of block sizes larger than 32 x 32 will only improve the CR values marginally.

7.5. THE EFFECTS OF TRANSMISSION ERRORS ON ABLC (SYSTEM C)

In this section, efforts are concentrated on the study of the effects of single transmission errors on the ABLC system, in particular on System C, in which the encoder selects either the subsampled data or its prediction error domain dependent upon the complexity measure. System C is chosen instead of the other systems because it was developed from the combination of System A and System B in which complexity is used as a criterion, and it is therefore envisaged that the effects of transmission errors on the latter system would be the same. System D is not investigated, because, in essence, it is similar to System C and the effects of errors are expected to be identical to those occurring with the latter system.

It has been mentioned in Section 6.6 that the penalty for high compression is greater susceptibility of the system to transmission errors. In ABLC, since IPBs of size m x m are used, the errors tend to cause propagation in the vertical and horizontal directions within the block. The severity of the effect is dependent on where the error actually occurs. If an error happens to occur in the bits that are assigned to quartic divisions and primitives greater than 2^2, then it can be assumed that the effect will be catastrophic but, on the other hand, if errors are confined within the BPBs, then only a small part of the block will be affected. Furthermore, being a variable rate system, loss of synchronisation is inevitable, and in the event of an error occurring, if the system has not achieved resynchronisation, the effect will propagate from one IPB to another. This can be prevented by sending a unique 'end of IPB' codeword, after each coded IPB, thus limiting the effects to a single IPB only.

Unlike the situation for the CABC system described in Chapter VI, the Error Sensitivity Factor (ESF) is not measured for ABLC. It is only intended to observe the effects of single transmission errors occurring at random on the decoded image, and to see the extent of the damage caused due to loss of synchronisation. It is also intended to prevent further obliteration and to observe the resulting effect on the decoded image by the inclusion of a unique 'end of IPB' codeword after every coded IPB. No attempt will be made to apply any error correction or error concealment techniques to the received image.

7.5.1 Experimental Procedures

The basic configuration illustrating the experimental procedures for the observation of the effects of channel errors on ABLC System C is depicted in Fig. 7.12.

The input test images, stored on magnetic tapes, are parts of CCITT document Nos. 1 and 2, as shown in Figs. 7.13(a) and 7.13(b) respectively. The images used were not subsampled, (unlike the case with System C previously described), to give a clearer view of the effects of transmission errors on the decoded picture. Throughout the simulation, an IPB of 16 x 16 is used. IPBs containing the input test data generate their corresponding prediction error domain using the 7th-order Markov model predictor, after which their complexities are computed. Whichever is the lower is then coded using ABLC. Figs. 7.13(c) and 7.13(d) illustrate which section of the image corresponds to the lower value of complexity for parts of document No. 1 and No. 2 respectively.

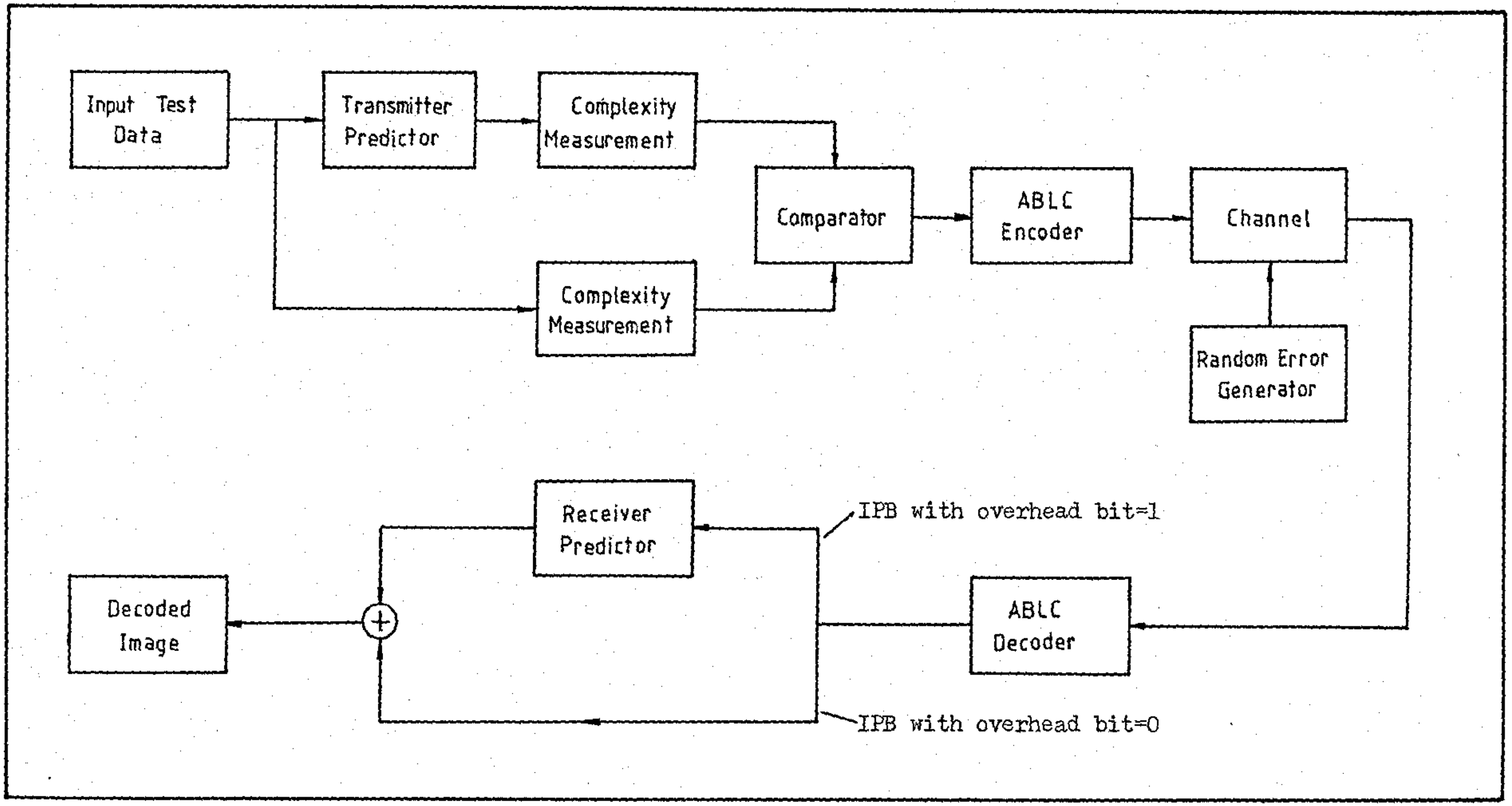
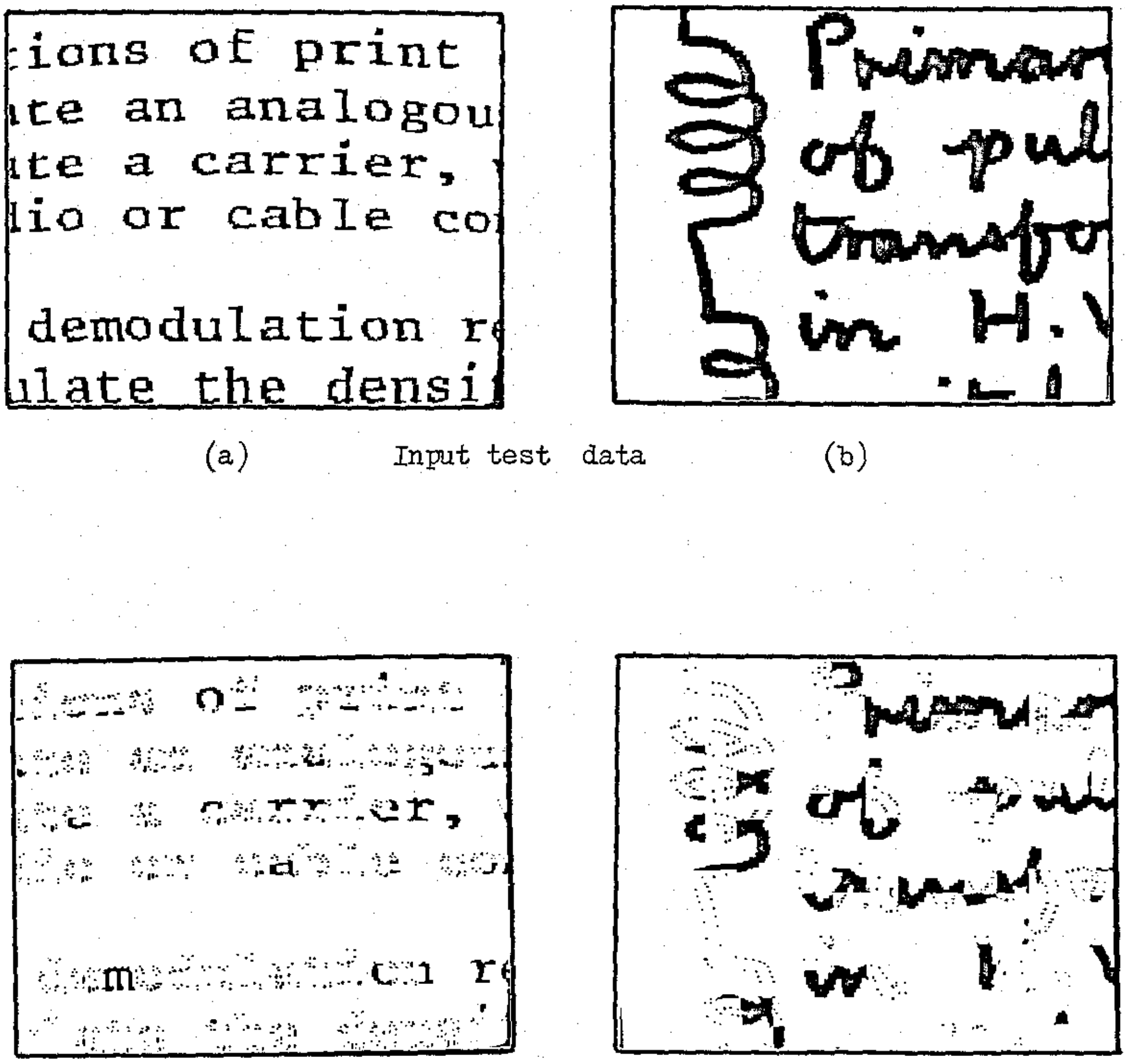


Fig. 7.12 - Block diagram of experimental procedure



(a) Input test data (b)

(c) Section of image having the lower complexity (d)

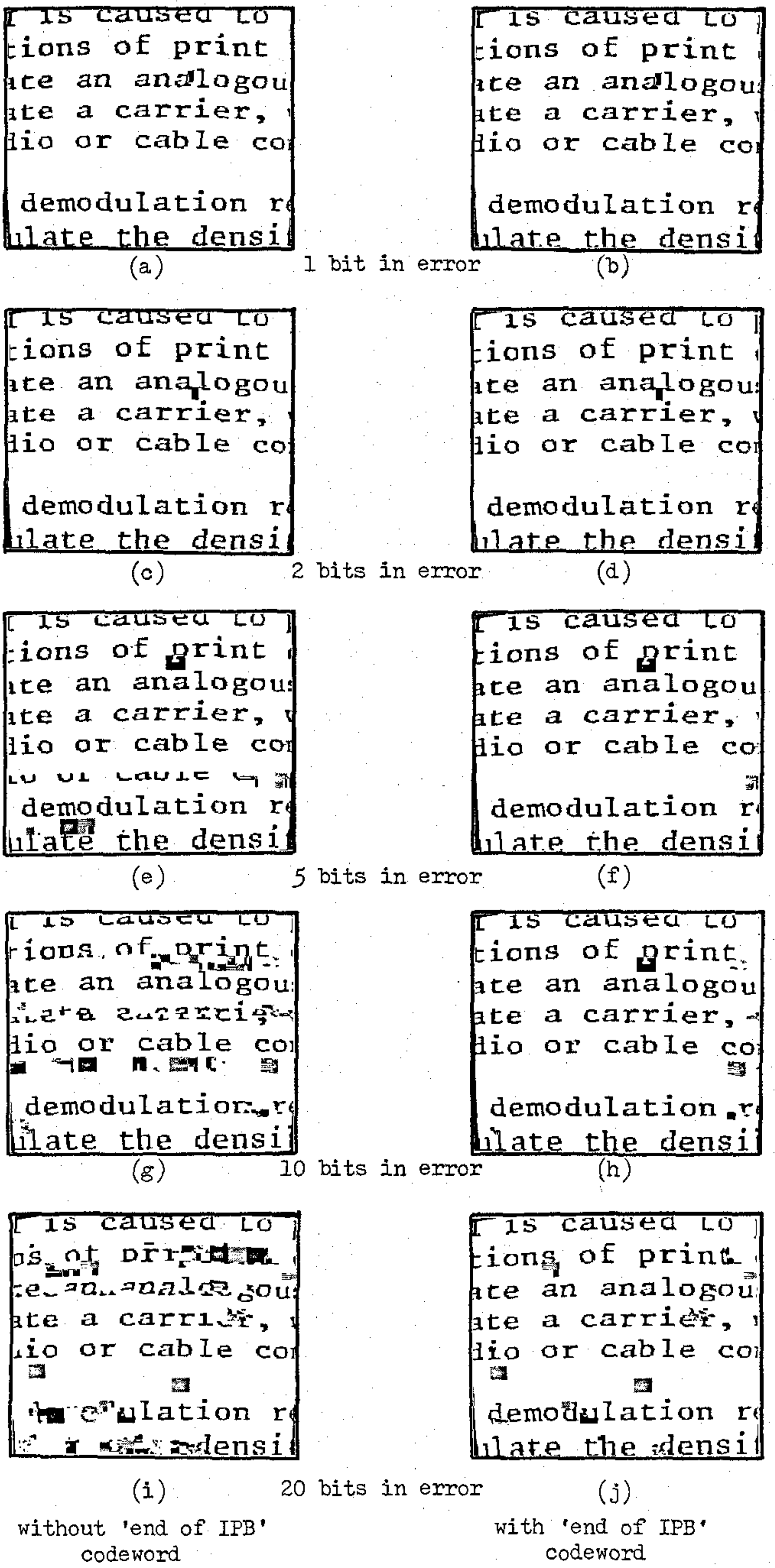
Fig. 7.13

Notice that, in the first case, the prediction error domain contains most of IPBs of lower complexity. The output bitstream of the ABLC encoder is then subjected to single transmission errors simulated using a Random Error Generator. The bit in question is inverted, after which the corrupted bitstream is decoded. The application of the receiver predictor restores the decoded image fully.

7.5.2 Test Results

The result of the simulation are shown in Fig. 7.14, for document No. 1 and Fig. 7.15, for document No. 2. The number of single transmission errors introduced at random into the channel are, successively, 1, 2, 5, 10 and 20. Thus in Figs. 7.14 and 7.15, pictures (a) and (b) contain 1 transmission error, picture (c) and (d) contains 2 errors, pictures (e) and (f), 5 errors, pictures (g) and (h), 10 errors and pictures (i) and (j) contain 20 single transmission errors. Images in Figs. 7.14 and 7.15 which are on the left hand side of the page are those without a unique 'end of IPB' codeword, while those on the right hand side have such a codeword.

From these test results, it is observed that even a few single transmission errors have a devastating effect on the resulting decoded image. As the error rate increases, the effect becomes more catastrophic. Being an area coding algorithm, the catastrophic error effect on the ABLC scheme is as expected, in which errors tend to propagate from one IPB to another. One way of limiting the error propagation is to transmit an 'end of IPB'



(a) 1 bit in error (b)

(c) 2 bits in error (d)

(e) 5 bits in error (f)

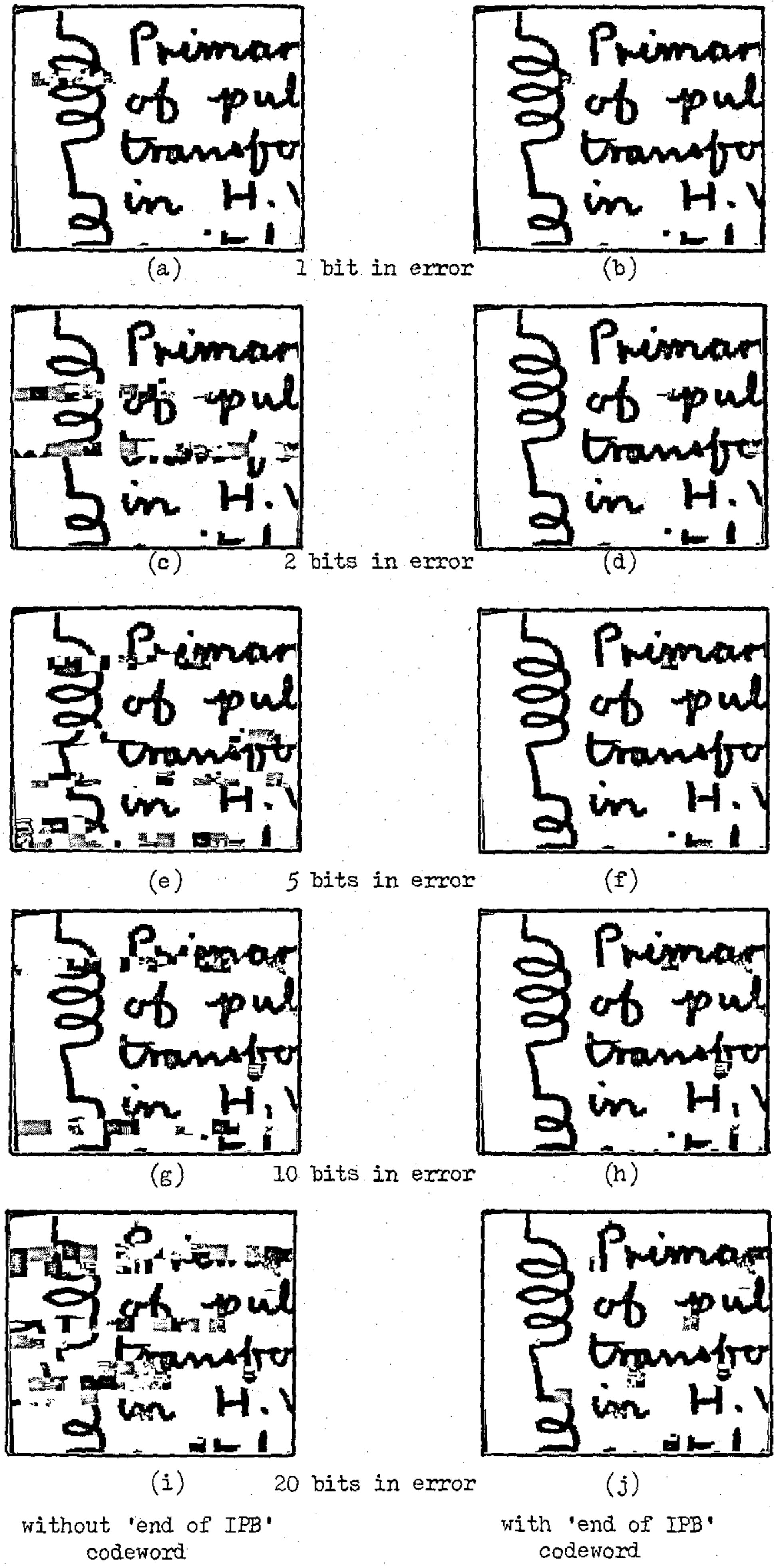
(g) 10 bits in error (h)

(i) 20 bits in error (j)

without 'end of IPB' codeword

with 'end of IPB' codeword

Fig. 7.14



(a) 1 bit in error (b)

(c) 2 bits in error (d)

(e) 5 bits in error (f)

(g) 10 bits in error (h)

(i) 20 bits in error (j)

without 'end of IPB' codeword

with 'end of IPB' codeword

Fig. 7.15

codeword after each coded IPB, and as can be seen from Figs. 7.14 and 7.15, the resulting image is more acceptable. The inclusion of this unique codeword, however, will increase the transmission time slightly. With data networks becoming more common place however, the problem of transmission errors will (hopefully) be reduced since special error control mesagges will be an integral part of the network, and so an 'end of IPB' codeword will not be needed.

7.6. SUBSAMPLING AND THE CCITT CODING STANDARD

This section discusses the use of the CCITT one- and two-dimensional coding standards for digital facsimile transmission of the subsampled documents. The Modified Huffman code which is used to represent black and white run-lengths is the one-dimensional standard whilst the Modified Relative Element Address Designate (READ) code (a line-by-line scheme in which the position of each changing element on the coding line is coded with respect to either the position of the corresponding changing element on the reference line, or with respect to the preceding changing element on the coding line), is the two-dimensional standard. Details of the above schemes are given in Chapter III and in (43).

It has already been shown in Chapter V that preprocessing by subsampling and post-processing by interpolation with proper restoration, is feasible for binary images and that the loss in quality is compensated by the high gains in compression ratios obtained. Since the Modified Huffman code and the Modified READ code have been widely accepted as the international standards, the inclusion of a preprocessor and a post-processor using subsampling and interpolation, by utilising the yet unspecified one- and two-dimensional extension codewords (see Table 3.3), would still allow intermachine compatibility and would therefore pose little difficulty. It is therefore important to note what gain in compression ratios can be achieved if subsampling and interpolation are incorporated into the CCITT one- and two-dimensional coding standards.

7.6.1 Modified Huffman Code and the Subsampled Documents

The performance of the Modified Huffman code on the subsampled CCITT documents was assessed by computer simulation. Each document, after subsampling, is represented by a total of 864 pels per line and 1187 lines. The average run-lengths, entropies and the maximum theoretical compression factor (Q_{max}) for the subsampled documents, as determined by Eqns. (2.4), (2.5) and (2.6) respectively, are listed in Table 7.6. Q_{max} in Table 7.6 is calculated based on the statistics of the subsampled documents, but since a 4:1 sample rate reduction is initially achieved, the value of Q_{max} must be multiplied by 4 in order to give a true indication as to how much compression can be theoretically obtained. Included in Table 7.6 are the actual compression ratios obtained (calculated from Eqn. (7.37)) when Modified Huffman coding, including an EOL codeword, is applied to the subsampled documents. For the sake of comparison, the compression ratios of the original, unprocessed, documents, using the Modified Huffman code, are also shown in Table 7.6. From Table 7.6, it is observed

Document Number	SUBSAMPLED CCITT DOCUMENTS						CR of Original Documents
	$\bar{r}_w$	$\bar{r}_b$	H_w	H_b	Q_{max}	CR	
1	82.39	2.25	4.61	2.18	12.46	34.52	15.59
2	131.61	6.08	7.29	3.28	13.02	39.13	17.08
3	45.33	3.03	4.79	2.33	6.79	19.54	9.02
4	21.17	2.09	3.82	2.00	3.99	11.75	5.35
5	42.67	2.12	4.90	1.83	6.65	19.37	8.89
6	70.17	2.56	5.43	2.11	9.65	24.69	11.15
7	24.82	1.59	4.99	1.46	4.09	11.49	5.29
					Average	22.93	10.34

Table 7.6 - Entropy, Q_{max} values and compression ratios for the Modified Huffman code

that the average compression ratio of the seven subsampled CCITT documents is 22.93, corresponding to an average transmission time, at a data rate of 4800 bits per second, of 45.2 seconds, for a minimum scan-line time (MSLT) of 0 ms. This represents an average improvement of about a factor of two when the Modified Huffman code is applied to the unprocessed documents.(121)

7.6.2 Modified READ Code and the Subsampled Documents

The READ code, a line-by-line sequential coding scheme, was proposed to the CCITT SG. XIV Study Group by Japan, as an optional extension to the one-dimensional coding scheme for Group 3 apparatus. The two-dimensional scheme provides greater compression efficiency when compared with its one-dimensional counterpart, particularly when documents are scanned at high resolution, by exploiting the high line-to-line correlation that exists in most facsimile documents.

Again, the performance of the Modified READ code using the subsampled CCITT documents was evaluated by means of computer simulation. Table 7.7 lists the frequency of occurrence of the various coding modes defined by the Modified READ code where P represents the Pass Mode, H, the Horizontal Mode and V, the Vertical Mode. The subscripts L and R indicate whether the position of element a_1 is to the left or the right of element b_1 respectively, and the number in brackets represents the value of the distance a_1b_1. (see Chapter III). It is observed that the probability of V(0) is greatest and this has therefore been assigned the shortest codeword.

Doc. No.	SUBSAMPIED CCITT DOCUMENTS										CR of Original Documents
	P	H	V(0)	$V_L(1)$	$V_L(2)$	$V_L(3)$	$V_R(1)$	$V_R(2)$	$V_R(3)$	CR	
1	1797	2468	12316	2148	724	555	1735	389	242	48.34	24.32
2	578	911	8026	1254	244	123	1582	563	156	76.93	35.39
3	2821	3394	29293	2490	739	1277	2146	555	256	32.66	15.92
4	8190	10355	45281	9829	2648	1607	6506	1346	492	14.14	7.10
5	3521	4197	27048	3520	1066	903	2927	526	283	29.42	14.84
6	1207	1508	19190	987	336	374	2237	726	182	53.75	24.58
7	7810	9520	33669	5723	3792	2799	8883	1919	582	14.04	7.58
								Average		38.47	18.53

Table 7.7 - Compression ratios and frequency of occurrence of the Pass, Horizontal and Vertical Modes for the Modified READ code

The compression ratios of the Modified READ code for the subsampled and the original CCITT documents are also shown in Table 7.7. The values obtained are with EOL included, the K factor taken to be infinity and MSLT set to 0 ms. Notice the very high compression ratios obtained for the subsampled documents, especially for document No. 2 and No. 6, (subsampling plays an important role in achieving these high values). On average, the compression ratio of the subsampled documents is more than twice that of the original documents. The average transmission time for the seven subsampled CCITT documents used, at a data rate of 4800 bits per second, is 23.8 seconds, which is 59 percent below the transmission time for the application of the one-dimensional Modified Huffman code to the subsampled documents.

7.7. CONCLUSION

In this chapter, the Adaptive Block/Location Coding (ABLC) scheme has been described, and mathematical analysis included, which demonstrates that ABLC is more efficient than D.F. coding by a substantial margin. Four different configurations of ABLC have been employed, in conjunction with subsampling as a means of preprocessing. The most efficient scheme is System D (see Section 7.4.4), in which an average compression ratio of 38.10 is achieved for the 7 subsampled CCITT documents. The high compression ratios obtained are attributed in part to subsampling prior to coding, where a 4:1 sample rate reduction is initially achieved. It is observed that ABLC performs extremely well for graphics-type documents whilst still maintaining high compression for alphanumeric documents. The effects of a single transmission error are also studied and pictorial illustrations indicate, as with variable-rate systems, in general, that errors tend to propagate, and the inclusion of a unique 'end of IPB' codeword is successful in limiting error extension effects to within one IPB only. The international CCITT one- and two-dimensional coding standards have also been applied to the subsampled documents, to indicate how efficient these schemes are, should preprocessing by subsampling be incorporated. Coding the subsampled documents using the Modified Huffman code and the Modified READ code, gives average compression ratios of 22.93 and 38.47 respectively for the 7 CCITT documents. Comparing these values with those of ABLC. it is clear that ABLC is about 40 percent more efficient than the Modified Huffman code and is comparable in performance to the Modified READ code.

7.8. NOTE ON PUBLICATIONS

A preliminary paper entitled 'Adaptive Block/Location Coding of Facsimile Signals Using Subsampling and Interpolation for Pre- and Post-processing' has been presented at the International Conference of Digital Processing of Signals in Communications held at Loughborough University on 7-10 April 1981.

A full paper of the same title has appeared on the Special Issue of the IEEE Transactions on Communications, Vol. COM-29, No. 12, December 1981, pp. 1925-34, devoted to Picture Communication Systems. Both papers described System C (see Section 7.4.3) and were jointly authored with R.J. Clarke.

Another paper entitled 'Subsampling and the CCITT Coding Standards' is due for presentation at the International Conference on Electronic Image Processing, University of York, England, on 26-28 July 1982. This paper, authored by M.G.B. Ismail and R.J. Clarke, describes Section 7.6 of this chapter.

CHAPTER VIII

SPEECH CODING USING A FACSIMILE TECHNIQUE

8.1. INTRODUCTION

In this chapter, the possibility of transmission rate reduction of speech signals through the use of entropy or noiseless coding is considered. The noiseless coding strategy of Chapter VI is employed, with the aim of transmitting both speech and text using the same encoder. As a preliminary preprocessing step, delta modulation (DM) systems (very cheap to produce and now available in integrated circuit form) are used since their binary output is very similar to a facsimile signal and is therefore amenable to coding in a similar way.

Delta modulation (DM) systems offer an attractive method of digitizing speech and video signals. Their use as source encoders has been investigated in recent times and the search for techniques which will lower the bit rate and hence increase efficiency without significant loss of quality has yielded several adaptive delta modulators (ADM)(122). ADM can produce, at a transmission rate of 32 kbits/s, excellent quality speech that is nearly as good as pulse code modulation (PCM) coded speech at a rate of 56 kbits/s. The transmission rate of ADM can be as low as 8 kbits/s, whilst still producing reasonably good quality speech. ADM is very simple, in terms of hardware realisation, and has many advantages (for example, less stringent filtering requirements and slow quality degradation with a noisy channel)$^{(123)}$ when compared with PCM or adaptive differential PCM (ADPCM). For this reason, ADM is becoming popular, and is replacing PCM in some parts of telephone communications where PCM has been exclusively used.

The output of a delta modulator, which is essentially a digital bitstream of 1's and 0's (similar to facsimile signals), presents the possibility of transmission rate reduction through entropy coding. If all the DM bits are totally independent and equally likely, then each DM bit would contain one bit of information. Generally, there is some dependence between the DM bits and the bitstream therefore contains, on average, less than one bit of information. If the average entropy is less than unity, then the information contained within the DM bitstream can be transmitted at a lower rate using suitable noiseless coding schemes (such as the coding algorithm of Chapter VI) whilst still maintaining the original DM quality of speech.

The work described in this chapter is exploratory and investigative in nature. It is carried out with the intention of marrying the transmission of speech and documents using the same encoder, thus saving system costs. The principal aim of this chapter is to investigate the maximum possible transmission rate reduction by measuring the average entropy of the bitstream generated when speech signal is applied to three DM systems, namely Linear Delta Modulation (LDM), Constant Factor Delta Modulation (CFDM) and Continuous Variable Slope Delta Modulation (CVSD). The detailed description of these three DM coders is contained in Section 8.2. Further search for a possible reduction in average entropy of the DM bitstream is carried out by the application of one bit and two bit prediction (similar to the Markov model predictor of Chapter VI). After the maximum possible reduction in average entropy is achieved using these predictors, the effect of applying a practical noiseless coding scheme (i.e. the coding strategy of Chapter VI) is studied, in order to reduce the transmission rate, and comparison is made with the theoretical entropy lower bound. The noiseless coding algorithm of Chapter VI, with some modifications, is explained in Section 8.3. Since speech is, by its very nature, quasiperiodic, the effects of exploiting this property by forming a two-dimensional image of the DM bitstream (like an ordinary facsimile image) over the voiced region only is studied. By the application of appropriate two-dimensional prediction and the practical noiseless coding scheme, it is hoped that further reduction in average transmission rate may be achieved. The one and two-dimensional cases are explained in detail in Section 8.4 and 8.5 respectively. Results of computer simulations are also given in these sections.

Since the study is preliminary in nature, various aspects of the system necessary for its successful implementation have not been carefully considered. One such aspect is buffer control and management, which is essential for the variable rate noiseless coding scheme used. The other aspect which has not been investigated is the effect of channel errors on the quality of speech at the receiver. Since the coding scheme is variable rate, error propagation due to loss of codeword synchronisation caused by channel errors is inevitable and this should be appreciated. The transmission of one PCM coded sample every few speech samples may be a solution to error propagation. These problems are, however, beyond the scope of this chapter.

8.2. DESCRIPTION OF CODERS USED

The delta modulator has been chosen as a preprocessing stage, transforming the analogue input speech signal into a stream of binary pulses, prior to actual noiseless coding, because of its relative simplicity in basic operation and its extremely inexpensive implementation. Furthermore, the resulting bitstream generated by the delta modulator (either '1' or '0'), being very similar to a facsimile signal, makes it suitable for further coding. Three delta modulators have been employed, namely the non-adaptive Linear Delta Modulator (LDM), and two adaptive delta modulators, i.e., Constant Factor Delta Modulator (CFDM) and Continuous Variable Slope Delta Modulator (CVSD). LDM, which works with a fixed step size for the 'staircase' approximation to an input signal, has various basic limitations. Small values of the step size introduce slope overload distortion during bursts of large signal slope; large values of step size accentuate the granular noise during periods of small signal slope; and, even when the step size is optimised, the performance of these modulators will be satisfactory only at sampling frequencies that may be undesirably high. For these reasons, the sampling rate for LDM in this chapter is chosen to be 64 kbits/s.

Lower rates for LDM are expected to give poorer quality output speech. To overcome the major limitations of LDM, and to employ DMs at relatively low operating sampling frequencies, several types of ADM have been proposed(122,124-126). In these schemes, the step size is changed in accordance with the time varying slope characteristics of the input signal, following a predetermined adaptation strategy. ADM systems can generally be classified as instantaneously (short-term) or syllabically (long-term) adaptive or 'companded' systems. In instantaneously companded ADM, the step size changes discretely at every sampling instant, based on the present and previous sign bits, and normally varies considerably from one sampling instant to the next. Systems described in refs. (124-126) belong to this scheme. In syllabically companded ADM, on the other hand, the step size varies in accordance with the input signal at a syllabic rate, i.e. the variation of step size is relatively slow compared with the sampling frequency. Such systems, like CVSD, are mainly used with speech or speech-like signals. Because of their ability to adapt more successfully to the input speech signal, the sampling frequencies for these coders, can be much lower than for LDM without causing severe degradation to the input signal and in the present simulations, sampling frequencies of 16 and 32 kbits/s are chosen.

A comprehensive survey of instantaneously and syllabically companded ADMs has been presented by Steele(122) and the following subsections only attempt to give a brief summary of the three DM systems used.

8.2.1 Linear Delta Modulator

The schematic diagram of a basic LDM is shown in Fig. 8.1. In such an encoder, the analogue input signal x(t), bandlimited to 3.4 kHz, is approximated by the tracking feedback signal y(t). This signal is constructed by locally decoding the sampled output of the quantizer using an integrator in the feedback loop and then subtracting the decoded signal from the input signal to form an error signal which is quantised to one of two possible levels depending on its polarity. The output of the quantizer is sampled periodically to produce the binary output pulses. To simplify the description, the sampled output is assumed to be constituted of impulses of unit amplitude and sign given by:-

$$L(nT_s) = \text{sgn}\left[e(nT_s)\right]$$

$$\text{and} \qquad e(nT_s) = x(nT_s) - y(nT_s) \qquad (8.1)$$

where n is an integer and $T_s = {}^1/f_s$ is the sampling period.

If the integrator is ideal, with gain γ , the feedback signal y(t) will be constructed by adding or subtracting a constant step size of amplitude γ at each consecutive sampling instant. Fig. 8.1(b) illustrates the waveforms that occur in an LDM system with ideal integration.

Decoding the binary pulses is carried out at the receiver through the local decoder, an integrator, to produce a waveform which differs from the original signal by the error signal in the encoder. The final decoded signal is obtained by low-pass filtering the waveform

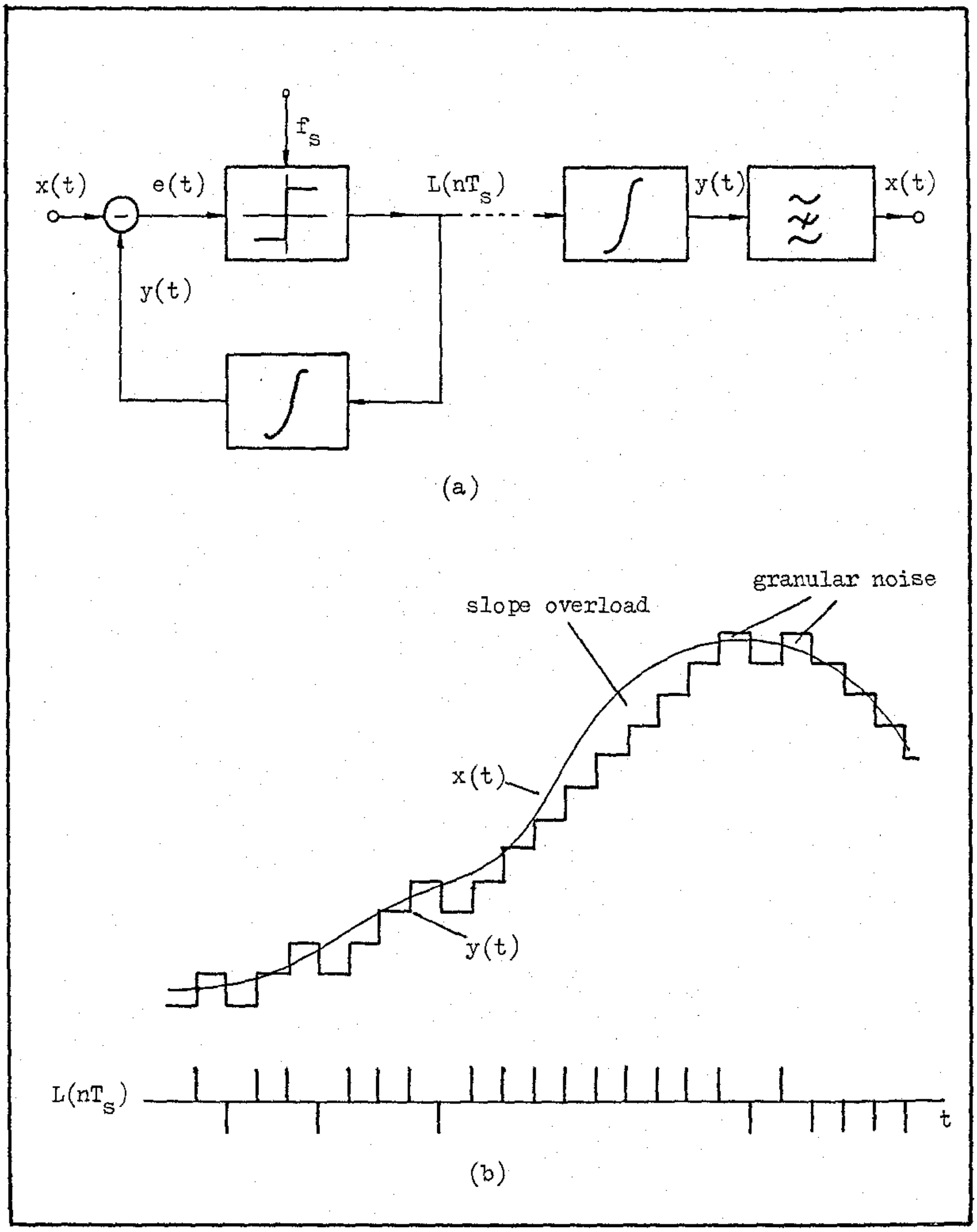


Fig 8.1 (a) - Linear delta modulation principle
(b) - Arbitrary waveform

at the output of the local decoder to remove any noise due to quantization effects.

8.2.2 Constant Factor Delta Modulator

The CFDM used in this chapter is similar to Jayant's Adaptive Delta Modulator with one bit memory(125), which uses instantaneous, exponential adaptation in the sense that the step size is changed at every sampling instant by one of two specific factors. Furthermore, the adaptation logic incorporates a one bit memory in which the immediately preceding CFDM bit L_{r-1} is stored. It is then compared with the incoming bit, L_r, for a decision on the new step size γ_r, where the subscript r denotes the rth sampling instant. Specifically, if the previous step size is denoted by γ_{r-1}, then the new step size is:-

$$\begin{aligned} \gamma_r &= P.\gamma_{r-1} \qquad \text{if} \quad L_r = L_{r-1} \\ \gamma_r &= -Q.\gamma_{r-1} \qquad \text{if} \quad L_r \neq L_{r-1} \end{aligned} \tag{8.2}$$

where P and Q are time invariant adaptation constants. It can be seen that the step size γ_r is dependent on the values of the previous and present bits and on P and Q which in turn results in the waveform $y(t)$ containing steps of varying heights. In particular, if y_r is the value of $y(t)$ at the rth sampling instant, then the output of the integrator is:-

$$y_r = y_{r-1} + \gamma_r$$

and $$L_r = \text{sgn}\ (x_r - y_r) \tag{8.3}$$

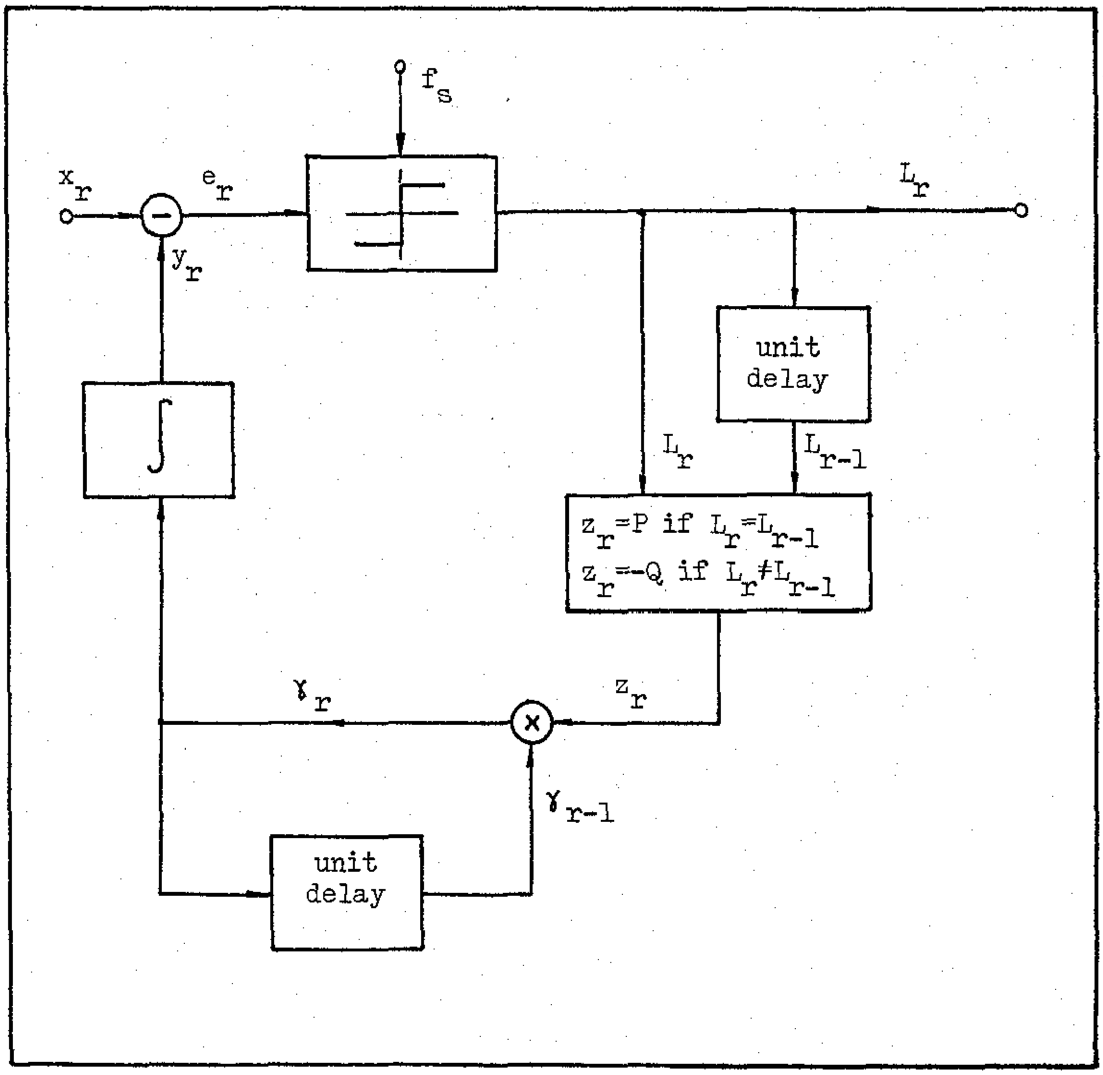


Fig. 8.2 - Block diagram of Jayant's CFDM

The decoder has the usual arrangement of a local decoder in the feedback loop followed by a low-pass filter to remove unwanted noise. The block diagram of Jayant's CFDM is shown in Fig. 8.2.

The adaptation constants P and Q are important parameters in this system. In order to adapt to the signal during slope overload, it is necessary to have $P > 1$. Also, in order to converge to a constant input signal, it is required to have $Q < 1$. In addition, to ensure stability of the decoded signal, it is necessary that $P.Q \leqslant 1$. Notice that when $P = Q = 1$, linear delta modulation is obtained. In the simulation carried out, ideal integration was used and the values of P and Q were optimised (in the sense of maximising signal to noise ratio) to 1.1 and 0.9 respectively.

8.2.3 Continuous Variable Slope Delta Modulator

In this scheme, the coder step size γ varies at a much slower rate than the instantaneous variations in the speech signal. The typical adaptation constant is of the order of 10 ms, and consequently γ approximately follows the variation of the speech envelope. The main advantage of such long term average adaptation technique is its robustness to channel errors. The coder maintains good performance at a channel error rate of 10^{-3} and the performance is fair even at 10^{-2}, whereas CFDM produces noticeably degraded speech quality around 10^{-4}, which become unacceptable at values of 10^{-3} and above.

Fig. 8.3 shows the block diagram of the CVSD. The CVSD algorithm used is as follows:-

$$L_r = \mathrm{sgn}\ (x_r - y_r)$$

$$y_r = \alpha y_{r-1} + \gamma_r . L_r \qquad (8.4)$$

and $$\gamma_r = \beta . \gamma_{r-1} + \mathrm{DLTA.ID}_r + \mathrm{DNULL}$$

where x_r = the input sample at the rth sampling instant.

y_r = the estimate of the incoming input sample (i.e. the integrator output).

y_{r-1} = the previous estimate of the input sample.

γ_r = the new step size.

γ_{r-1} = the previous step size.

γ_r, the new step size, is controlled by the step size generator which adds a constant DLTA to the previous step size when four consecutive CVSD outputs are identical, viz:-

$$\begin{aligned} \mathrm{ID}_r &= 1 \quad \text{if} \quad L_r = L_{r-1} = L_{r-2} = L_{r-3} \\ &= 0 \quad \text{otherwise.} \end{aligned} \qquad (8.5)$$

DNULL, usually small compared with DLTA, is a constant added to γ_{r-1}, when $\mathrm{ID}_r = 0$, to ensure that the minimum step size is non-zero. DLTA and DNULL are given by:-

$$\begin{aligned} \mathrm{DLTA} &= (1-\beta)(1-\alpha)V \\ \mathrm{DNULL} &= (1-\beta)(1-\alpha)V_1 \end{aligned} \qquad (8.6)$$

where V_1 and V are constants taken to be 5 and 1000 respectively.

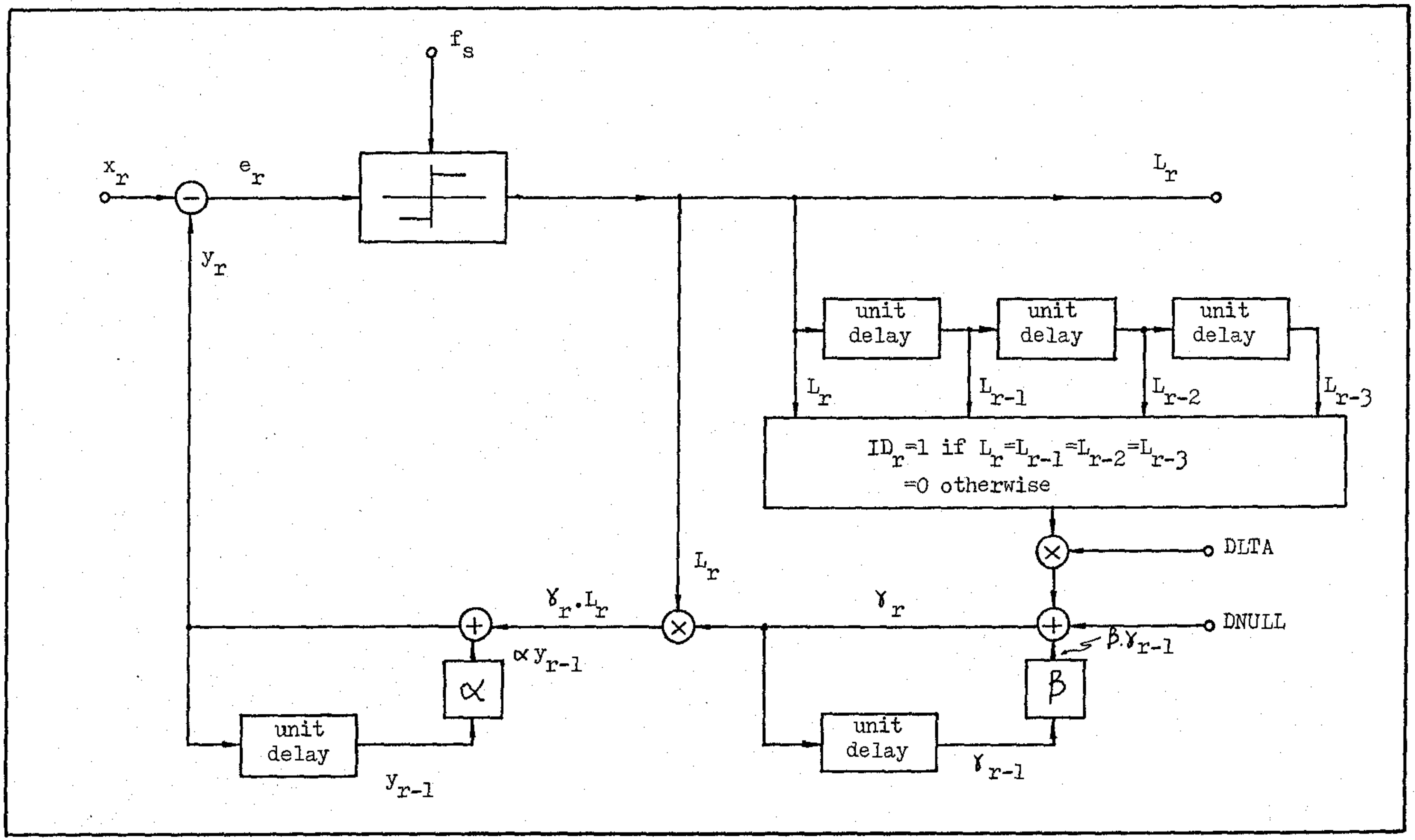


Fig. 8.3 - Block diagram of CVSD

α and β are leakage factors associated with the estimating integrator and the step size integrator given by:-

$$\alpha = \exp\left[(-{}^{1}/f_s)/\tau_2\right]$$
$$\beta = \exp\left[(-{}^{1}/f_s)/\tau_1\right] \qquad (8.7)$$

where f_s is the sampling frequency. The time constant for the step size generator τ_1 and the time constant for the estimating integrator τ_2 are chosen with reference to the actual waveform of the voice signal. A typical voice signal has most energy in the frequency range between 700 Hz and 1000 Hz and the envelope varies at a rate between 60 and 100 Hz. Therefore, τ_1 is adjusted to 5.69 ms, corresponding to approximately 100 Hz, and τ_2 is adjusted to 1 ms corresponding to 1000 Hz. It is of interest to note that the coefficients α and β can be adjusted differently for different CVSD processor.

Table 8.1 gives the values of α and β at different f_s, in particular, at 16 kHz and 32 kHz.

f_s kHz	α	β	DLTA
16	0.94	0.99	0.6
32	0.96	0.99	0.4

Table 8.1 - Values of α and β at different f_s.

8.2.4 Optimisation of Coders

Before any possible bit rate reduction is examined, the three coders need to be optimised to achieve an operating region of maximum signal-to-quantization noise ratio (SQNR). This is carried out by using an input speech signal bandlimited to 3.4 kHz with a sampling rate of 8 kHz. The input speech signal, spoken by a male, is " Sister, Father, S.K. Harvey,....... ", corresponding to a duration of about 4 seconds.

Since the delta modulation systems used operate at higher frequencies than the sampling rate of speech (i.e. LDM at 64 kHz, CFDM and CVSD at 16 and 32 kHz), the input speech signal (at 8 kHz) is interpolated to obtain the sampling frequencies required by the respective coders. Interpolation is carried out using the Cubic B-spline algorithm described in Chapter V. For instance, if the DM system is required to operate at a sampling frequency of 32 kHz, the input speech signal is first interpolated to 16 kHz and then 32 kHz. This is achieved by constructing the sample between two consecutive samples at 8 kHz, and then constructing the sample between two consecutive samples at 16 kHz. Similar operations apply for the DM system running at 64 kHz.

Once the coders are set to the desired operating sampling frequencies, they are optimised by observing the peak signal-to-quantization noise ratio (SQNR) which is determined by:-

$$\text{SQNR} = \frac{\sum_{r=1}^{NS} x_r^2}{\sum_{r=1}^{NS} (x_r - y_r)^2} \qquad (8.8)$$

where NS is the total number of input samples.

The SQNR performance curves as a function of input signal power for LDM, CFDM, and CVSD whose sampling frequencies are 64 kHz, 32 kHz and 16 kHz are depicted in Fig. 8.4.

As can be seen from Fig. 8.4, LDM, with a sampling frequency of 64 kHz, operates in an optimum condition (i.e. maximum SQNR), when the input signal power is increased by a factor of 7.94 relative to the input signal power of the original speech sequence. In other words, the input speech signal is amplified by 9 dB to achieve optimum condition. It is also observed that although LDM has a sampling frequency of 64 kHz, its performance is only as good, if not worse, than that of CFDM and CVSD operating at 32 kHz. It is therefore obvious that LDM has various limitations, such as small dynamic range, inability to 'track' the input signal well (slope overload) and problems of granular noise for signals with low input power. In order to obtain even moderately satisfactory performance, the sampling rate for LDM must be undesirably high.

On the other hand, CFDM (operating at 16 and 32 kHz) has a large dynamic range and operates in an optimum condition when the input signal power is a factor of 19.95 lower than the original input.

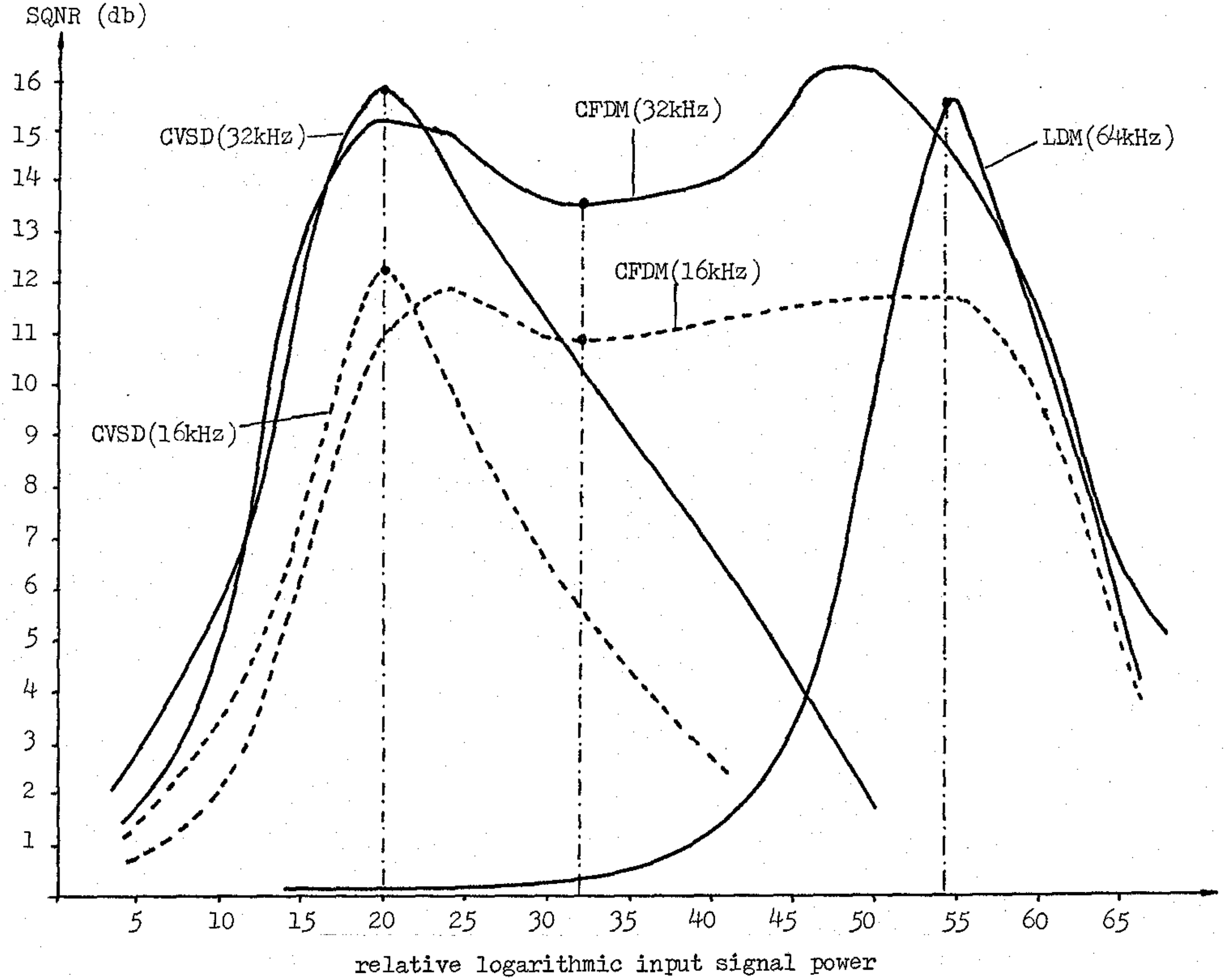


Fig. 8.4 - SQNR curves as a function of input signal power

Optimisation is carried under the condition that P.Q is always equal to 0.99, and the value of P (the step size multiplier) obtained is 1.1. CVSD (also operating at 16 and 32 kHz), with a smaller dynamic range, achieves its maximum SQNR when the input signal power is reduced by a factor of 316.2 (25dB attenuation). Parameters α, β, V and V_1 have been specified earlier in Section 8.2.3.

Having optimised the coders, entropy measurements can be made and the application of a practical noiseless coder, in order to reduce the number of bits required for transmission, can be examined.

8.3. NOISELESS CODING ALGORITHM

Delta Modulators are basically waveform coders with a two-level quantizer and as such they only produce a binary sequence of 1's and 0's. Generally, some form of entropy coding can be applied to this sequence of 1's and 0's further to reduce the transmission rate. Entropy coding is a technique whereby a string of digits is rearranged into a set of, on average, shorter sequences which carries the exact information of the original sequence. Specifically, the average entropy of the DM binary sequence is given by:-

$$H = -\frac{1}{B} \sum_{k=1}^{2^B} p_k \log_2 p_k \qquad (8.9)$$

where B is the block size taken and p_k is the probability of occurrence of a particular state (2^B possible states). To encode optimally the DM binary sequence, the Huffman code(36) can be used, where the most probable B bit sequence is assigned the shortest codeword, whilst larger codewords are used for the less probable sequences. The problem with the Huffman code is that it requires a large codebook and is difficult to realise practically. Furthermore, such a code is only optimum for one particular probability distribution and a mismatch in the distribution would render the code inefficient. However, to consider a set of Huffman code for a set of different probability distributions would be rather impractical.

Coding schemes such as Huffman codes or run-length codes depend parametrically on the statistics of the source. It is well-known that discrete data sources arising in practical systems are generally characterised by only partially known and varying statistics(127). Ideally, a class of universal codes(128,129) which are totally independent of the source statistics is required to code the source efficiently. In this section, the use of the coding strategy of Section 6.4 (Adaptive Block/Run-Length coding) for efficient coding of discrete data sources with partially known probability ordering but unknown values is described. The application of this coding strategy can be generalised to other data sources by transforming them using suitable preprocessing (as in the case of facsimile images employing the 7th order Markov model predictor to generate a prediction error domain).

We begin by asserting that the binary output of the DM coders is made up of a series of concatenated fundamental sequences+. An

+The definition is similar to that used by R.F. Rice(127,130) example of a fundamental sequence is given below:-

0000010000000010010000010000000001010010010001

where w is the number of 1's in the fundamental sequence (in this case $w = 9$). Note that a fundamental sequence always has exactly w 1's and will always end with a 1. For a fixed w, the length N of a fundamental sequence is a random variable. As in the case of facsimile using Adaptive Block/Run-Length Coding, in order to code the fundamental sequence, it is divided adaptively into smaller blocks where the maximum block size is M and M is given by Eqn. (6.34) as $M = \frac{N \ln 2}{w}$. The determination of an optimum M has been described in Section 6.4.1. Coding of the fundamental sequence is carried out in exactly the same manner as was done for facsimile images explained in Chapter VI and as illustrated in Fig. 6.8.

However, as a further development, in order to ensure that no data expansion occurs, another relationship is introduced:-

(a) if $w \leqslant 0.34N$; then the data is coded using the coding procedure explained earlier (i.e. Adaptive Block/Run-Length Coding) together with an identification code '0'.

(b) if $0.34N < w \leqslant 0.70N$; then the data is transmitted uncoded with an identification code '10'.

(c) if $0.70N < w \leqslant 1.0N$; then the bit sequence N is complemented (bit by bit complement) and an imaginary one is inserted at the end of the sequence. The complemented data is coded using Adaptive Block/Run-Length Coding and transmitted together with an identification code '11'.

N is defined as the length of the fundamental sequence and w is the number of 1's present. These relationships are such that, when tested using real data, good results are obtained.

An example of condition (c) in which $0.70N < w \leqslant 1.0N$ is illustrated in Fig. 8.5.

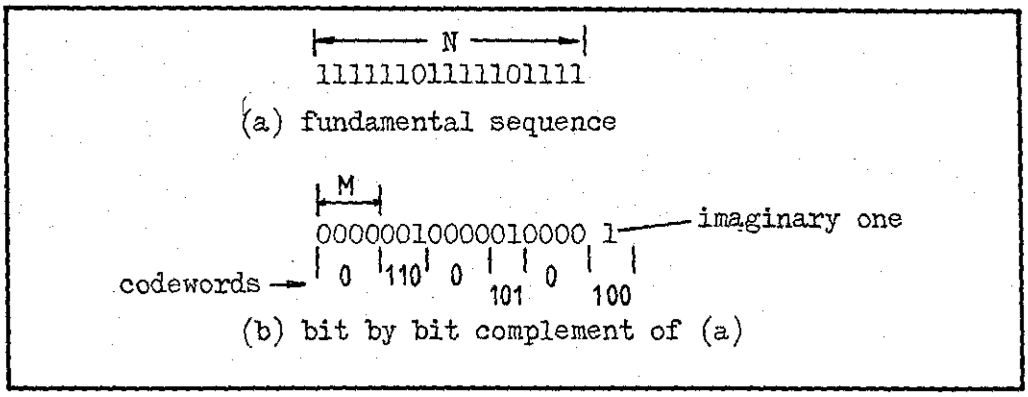


Fig. 8.5 - Example of condition (c) $(0.70N < w \leqslant 1.0N)$

If the fundamental sequence of Fig. 8.5(a) is considered, it would be uneconomical to code it directly using Adaptive Block/Run-Length Coding. Instead, the bit sequence is complemented bit by bit and an imaginary one inserted at the end of the sequence as shown in Fig. 8.5(b). Coding is then carried out, and by so doing, fewer bits are required to represent the fundamental sequence.

To verify that the coding strategy requires only partial knowledge of the probability ordering and not the unknown values, let us define $P = \{p_i\}$ as the probability distribution of the locations of 1's in the block. For example, p_0 is the probability of a block of size M containing all zeros, p_1 is the probability of a one located at position 1, p_2 is the probability of a one located at position 2 and so on, viz:-

p_0	representing	0000.........00
p_1	representing	1
p_2	representing	01
.	.	.
.	.	.
.	.	.
.	.	.
.	.	.
.	.	.
p_M	representing	0000.........01
		$\longleftarrow$ M $\longrightarrow$

Adaptive Block/Run-Length Coding assumes that $p_0 \geqslant p_1$, $p_0 \geqslant p_2$, ... $p_0 \geqslant p_M$. In other words, p_0 is always greater than any of the other probabilities. The probability ordering of p_1, p_2, ... p_M is immaterial. The coding algorithm, therefore, requires a partial knowledge of the probability ordering but not the actual values. Note that because of the assumed probability ordering, the shortest codeword is assigned to the most probable event while equal length codewords are assigned to other events (See Fig. 6.8).

8.4. ONE DIMENSIONAL CASE

The one-dimensional DM system using entropy coding is shown in Fig. 8.6. The input speech signal enters the delta modulation system whose sampling frequency is f_s kHz. The output DM bitstream is then stored for further noiseless coding to reduce the transmission rate. Since the codes generated by the noiseless coder (Adaptive Block/Run-Length Coding) are variable in length, a buffer is required for a constant rate transmission.

The input speech signal spoken by a male (" Sister, Father, S.K. Harvey, ... ") is bandlimited to 3.4 kHz and for the simulation, four seconds of it is used. The entropies of the stored DM bitstream for LDM, CFDM and CVSD are measured using Eqn. (8.9) for values of B = 1,2,4,5,8,10. The application of both a one bit and a two bit predictor, in which the present DM bit is predicted based on the previous one and two bits, respectively, generates a prediction error domain for the DM bitstream and reduces the entropy values still further. The further application of the one bit and two bit predictors to the prediction error domain of the DM bitstream already generated, produces another prediction error domain which in turn allows further reduction in entropy.

The entropy measurements for the three DM systems used are shown in Fig. 8.7 for LDM operating at a sampling frequency of 64 kHz, Figs. 8.8(a) and 8.8(b) for CFDM operating at 16 kHz and 32 kHz, respectively, and Figs. 8.9(a) and 8.9(b), respectively, for CVSD at 16 kHz and 32 kHz. The entropy measurements, shown as continuous

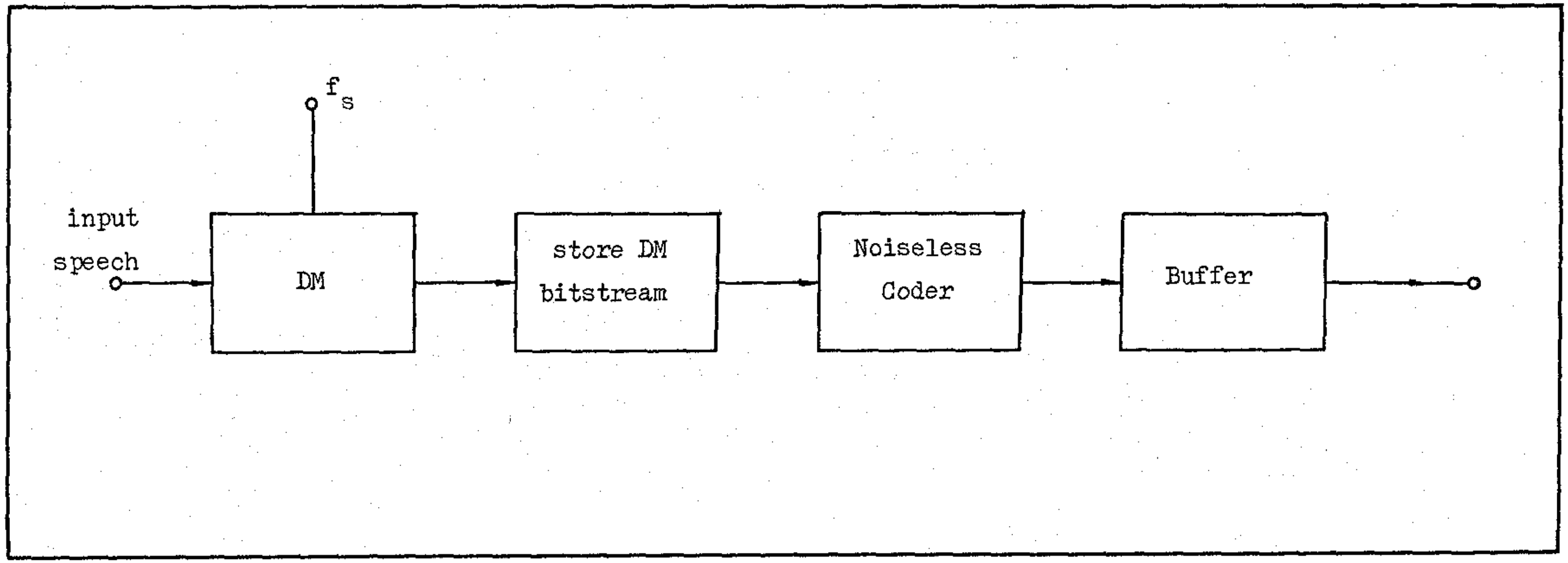


Fig. 8.6 - Block Diagram of One-Dimensional DM System

curves on the graphs, are for the cases where no prediction is applied and also where the one bit and two bit predictors are applied twice. It can be seen from all the graphs that the application of a two bit predictor twice yields the lowest entropy for various values of block size (B) used. There is a reduction from 1 bit per DM sample to 0.277 bits per DM sample for LDM (at 64 kHz), to 0.825 and 0.720 bits per DM sample for CFDM (at 16 and 32 kHz respectively), and to 0.414 and 0.399 bits per DM sample for CVSD (at 16 and 32 kHz respectively). In other words, there is a maximum possible reduction in transmission bit rate from 8 bits per Nyquist sample to 2.22 bits per Nyquist sample for LDM, from 4 bits per Nyquist sample for CFDM and CVSD (at 32 kHz) to 2.88 and 1.59 bits per Nyquist sample respectively, and from 2 bits per Nyquist sample to 1.65 and 0.828 bits per Nyquist sample for CFDM and CVSD (at 16 kHz) respectively. (All figures quoted above are taken at value of $B = 10$). This represents a substantial possible transmission rate reduction for all cases.

Using Adaptive Block/Run-Length Coding to code the prediction error domains of the DM bitstream, the average number of bits per DM sample for each case were computed and are shown as broken lines on the graphs of Figs. 8.7, 8.8, and 8.9. The points are plotted as functions of w (the number of 1's in a fundamental sequence) where the length N is a random variable.

It is observed that by suitable reversible preprocessing (prediction in this case), the average bit rate obtained is close to the theoretical entropy bound, particularly for the case where

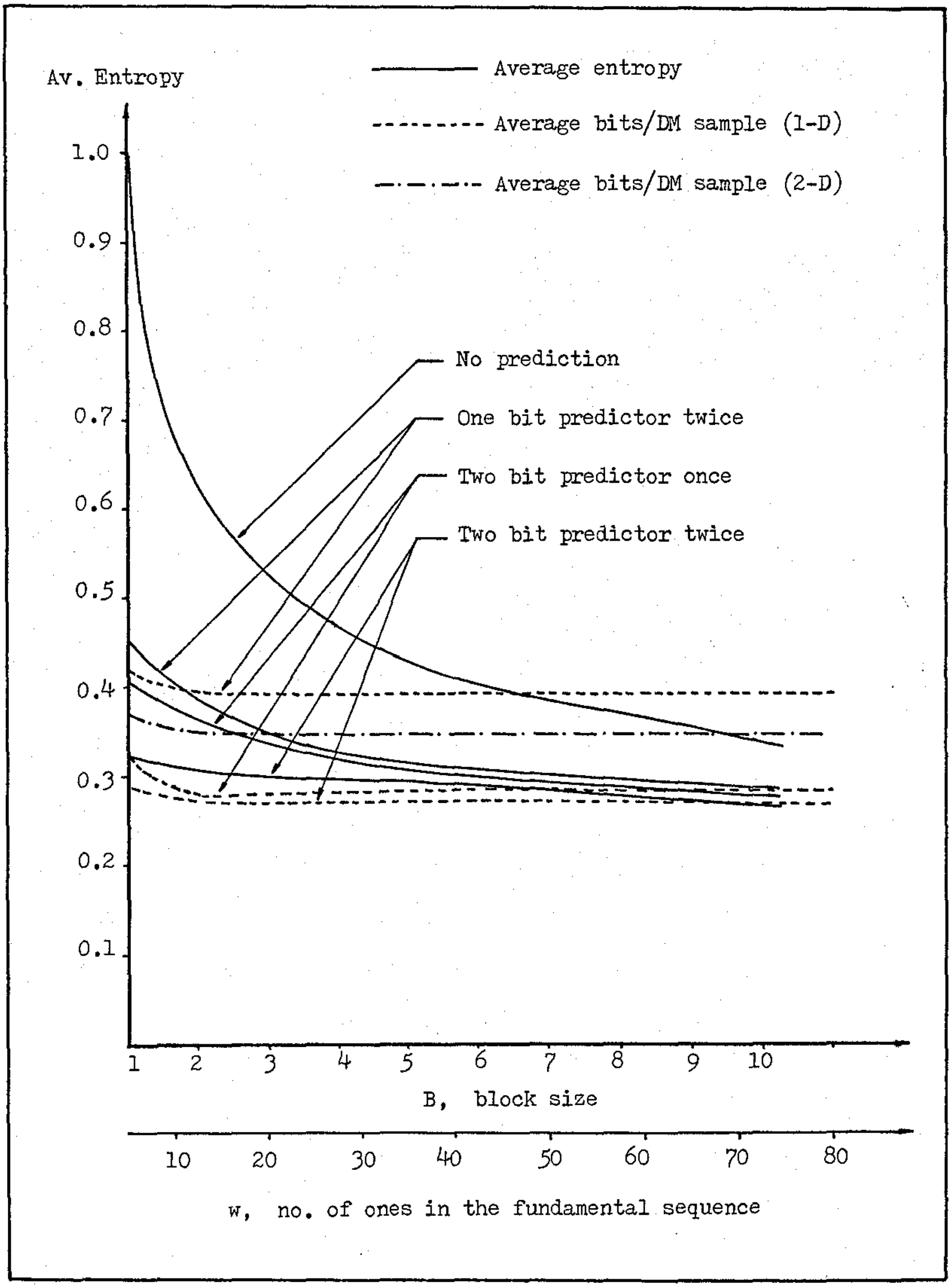


Fig. 8.7 - Performance of LDM (64kHz)

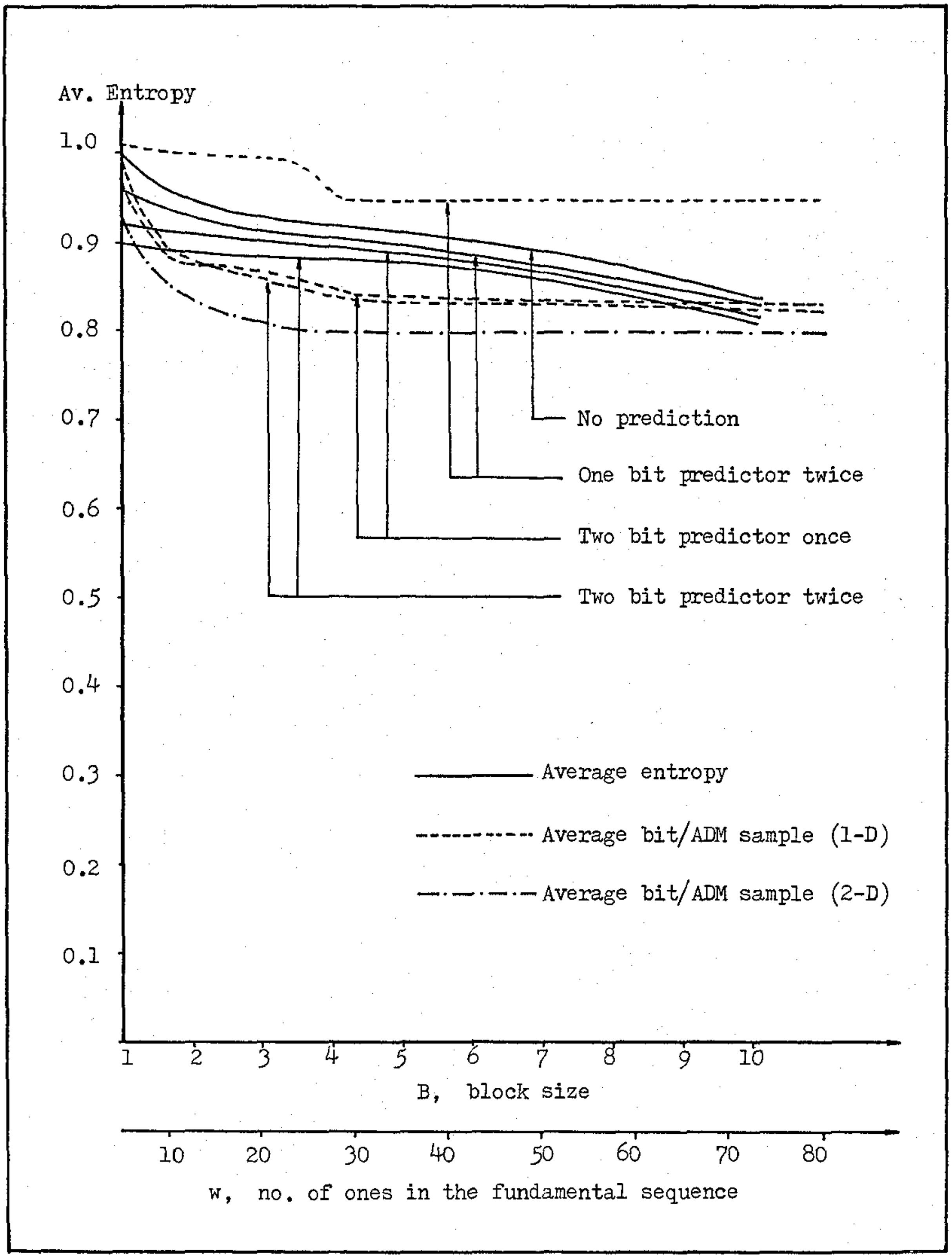


Fig. 8.8(a) - Performance of CFDM (16kHz)

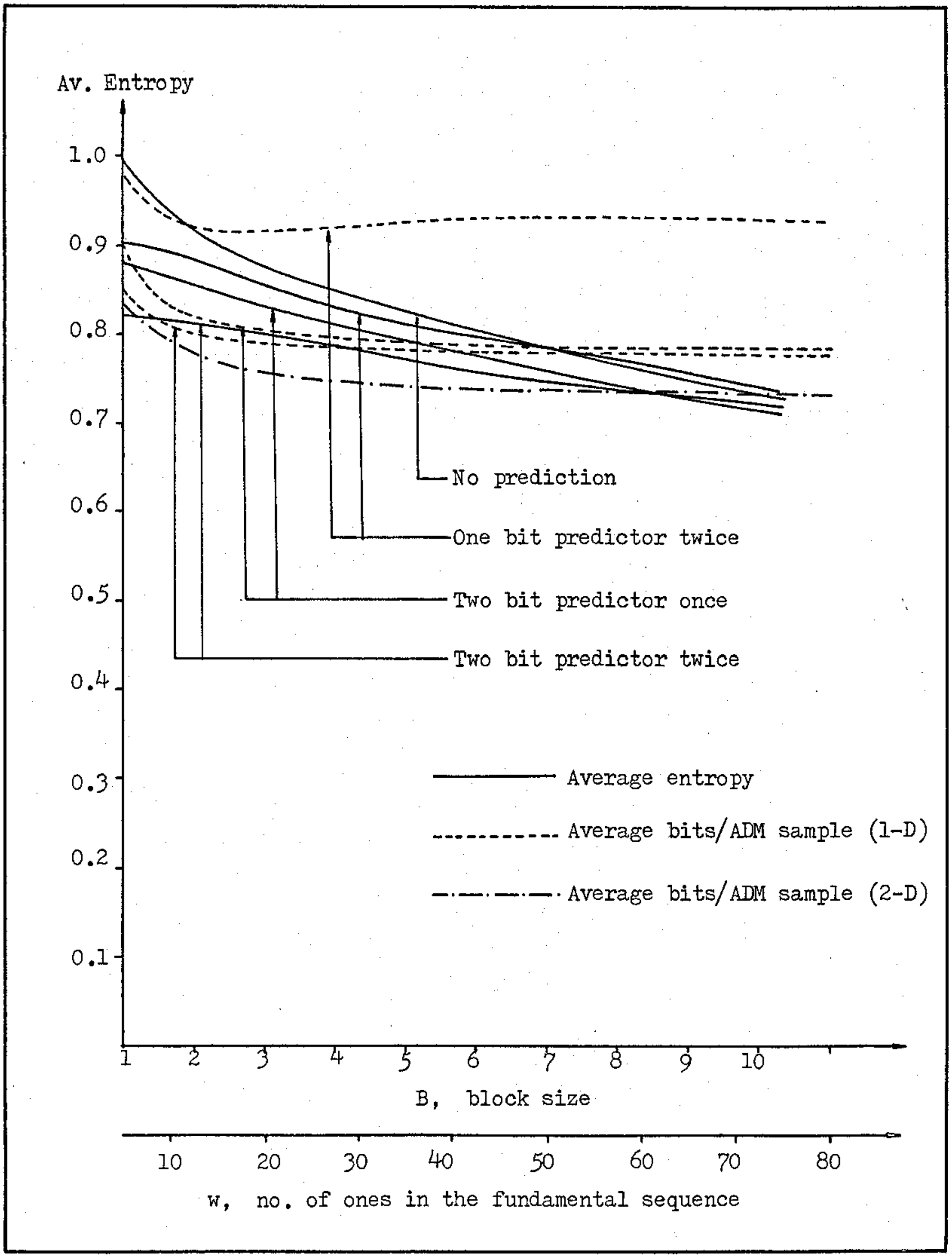


Fig. 8.8(b) - Performance of CFDM (32kHz)

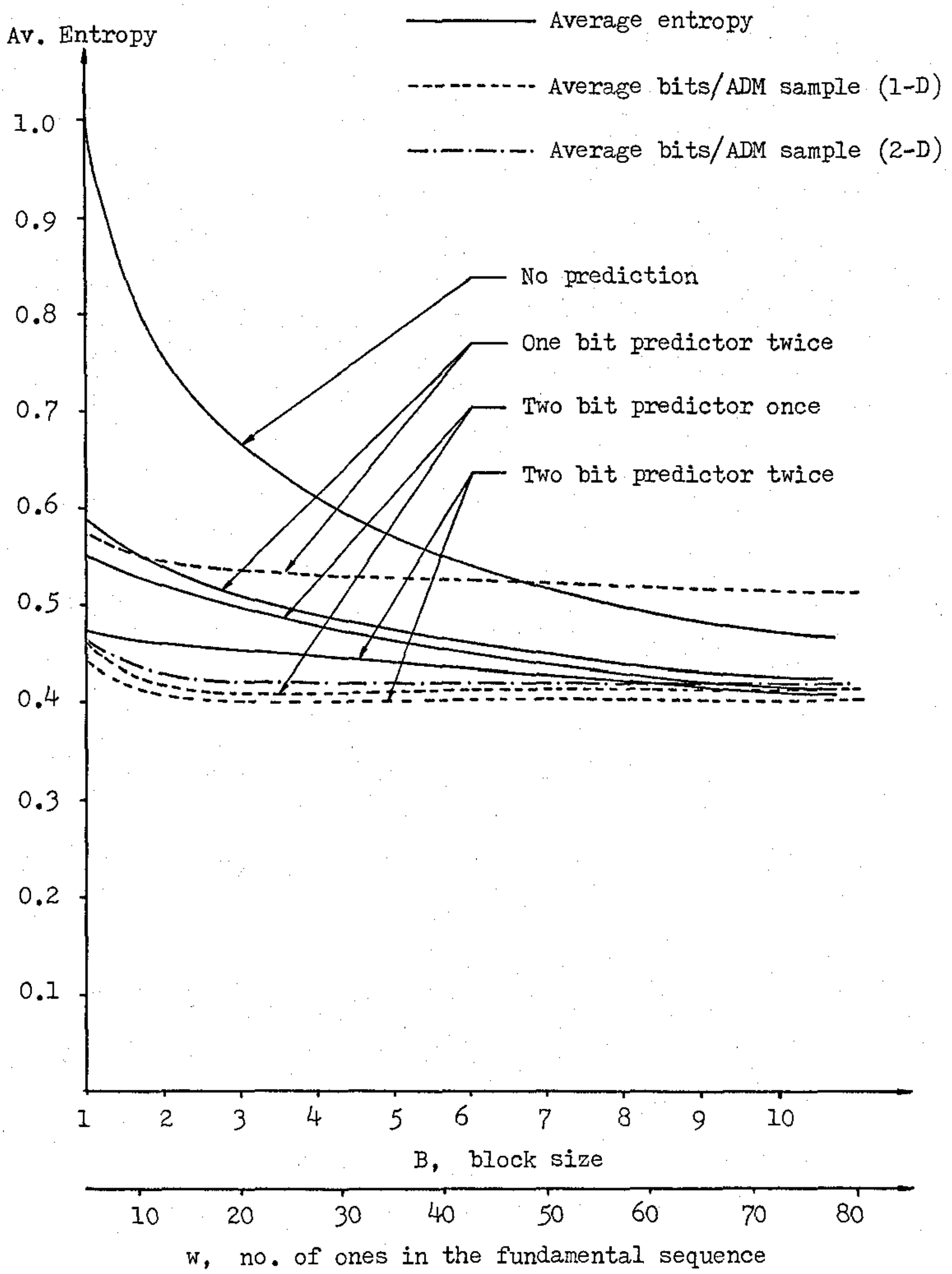


Fig. 8.9(a) - Performance of CVSD (16kHz)

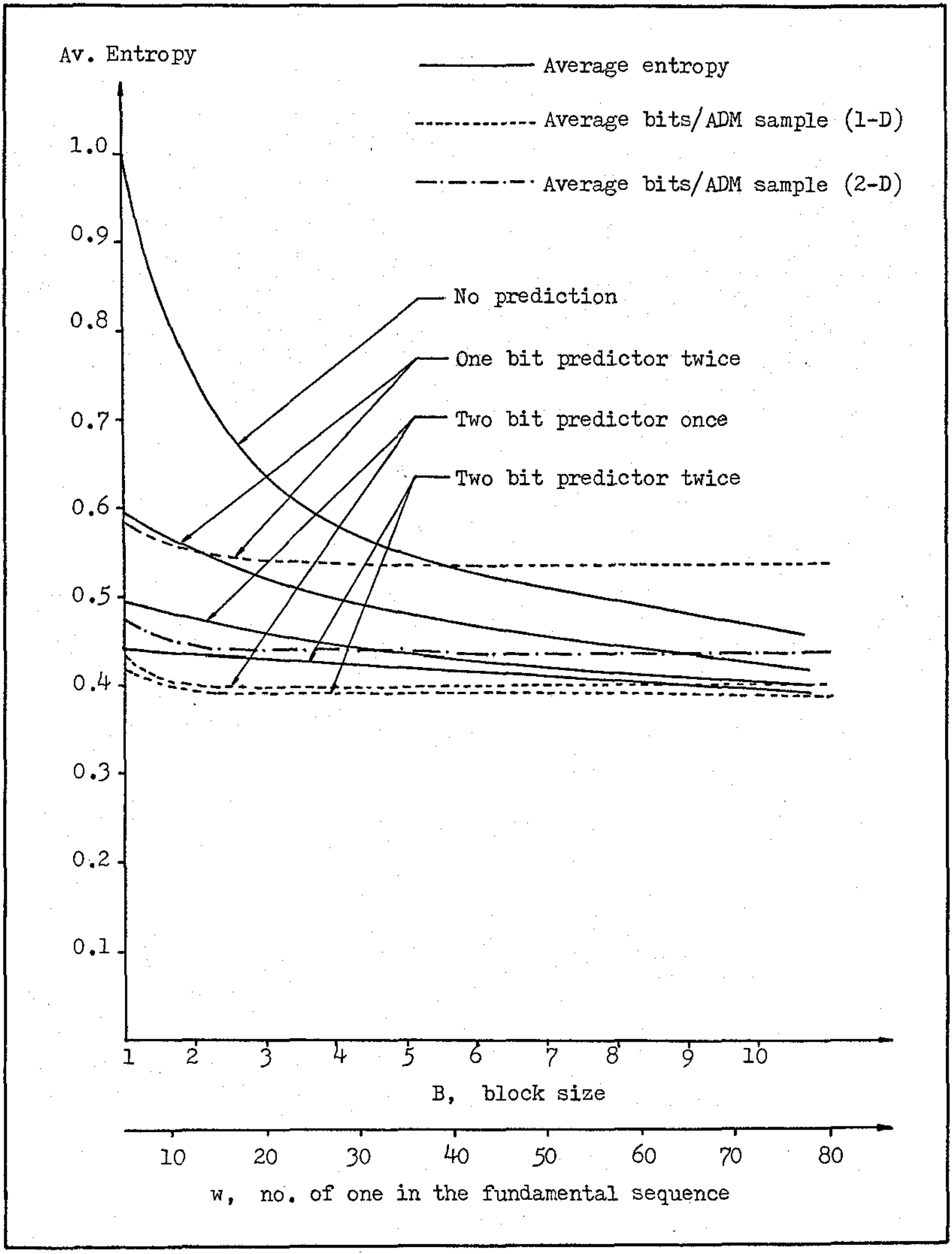


Fig. 8.9(b) - Performance of CVSD (32kHz)

two bit prediction is used. It can be seen therefore that the code is highly efficient if proper preprocessing is applied, i.e. the code will perform well if the probability ordering relationship described earlier is satisfied. The measured average bit rate when two bit prediction is applied twice falls to as low a value as 0.275 bits per DM sample for LDM (64 kHz), 0.827 and 0.782 bits per DM sample for CFDM (at 16 and 32 kHz respectively), and 0.405 and 0.385 bits per DM sample for CVSD (at 16 and 32 kHz respectively). (Notice that some of these figures are slightly lower than the entropy measurements when B = 10. This is due to the adaptivity of block size M of the coding strategy, where in certain cases M exceeds 10). In other words, the transmission rate for LDM can be reduced from 64 kbits/s to approximately 17.6 kbits/s, from 32 kbits/s to 25.03 kbits/s and 12.42 kbits/s for CFDM and CVSD, respectively, and from 16 kbits/s to 13.23 kbits/s and 6.48 kbits/s, respectively, for CFDM and CVSD. These represent substantial transmission rate reductions, and the advantage of employing such a scheme (entropy coding) is that the quality of the received speech signal (provided the channel is ideal) will be the same as if the DM systems were operating at their original sampling frequencies.

8.5. TWO-DIMENSIONAL CASE

The notion of a two-dimensional DM system stems from the fact that speech by its very nature is quasiperiodic, and possesses the property of having good correlation between samples of one pitch period and corresponding samples in the following pitch period. Fig. 8.10 shows a segment of voiced speech demonstrating its quasiperiodicity. Exploitation of this property provides an opportunity for further transmission rate reduction. Since the output of a DM system consists of binary bitstream of 1's and 0's, the generated bit sequence over each pitch period can be formed into a two-dimensional binary array (similar to a facsimile image), where there is a tendency for these bit sequences to have good correlation as well between pitch periods. It is hoped therefore, that, after the formation of the 'bit sequence image', the application of a two-dimensional predictor will further reduce the average number of bits required per DM sample.

The block diagram of the two-dimensional DM system using entropy coding is shown in Fig. 8.11. The system is more complicated than its one-dimensional counterpart. The input speech signal, bandlimited to 3.4 kHz, has a duration of 4 seconds. It is segmented into 250 ms sections and then fed to a pitch detector(131) where the number of pitch periods is computed. If the number of pitch periods is less than two (i.e. a 'bit sequence image' cannot be formed), the whole speech segment is delta modulated (switch SW position A). As is the case of one-dimensional DM, the application of a two bit predictor twice to the stored DM bitstream generates a prediction error domain where Adaptive Block/Run-Length Coding is subsequently employed. On the other hand, if the number of pitch periods is greater than or equal to two, a 'bit sequence image' can be formed (switch SW position B). At the same time, the location of the first

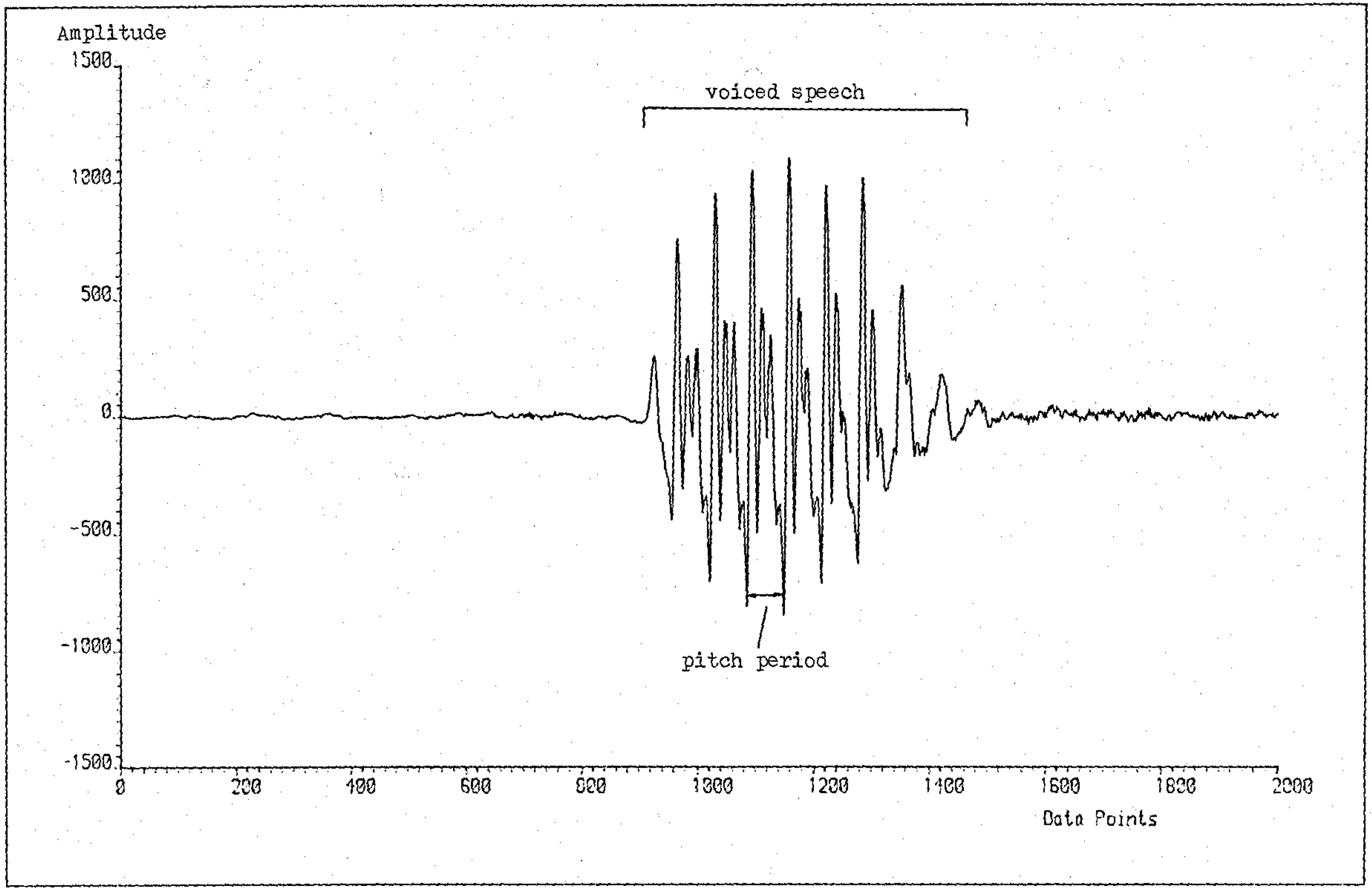


Fig. 8.10 - Segment of voiced speech demonstrating its quasiperiodicity

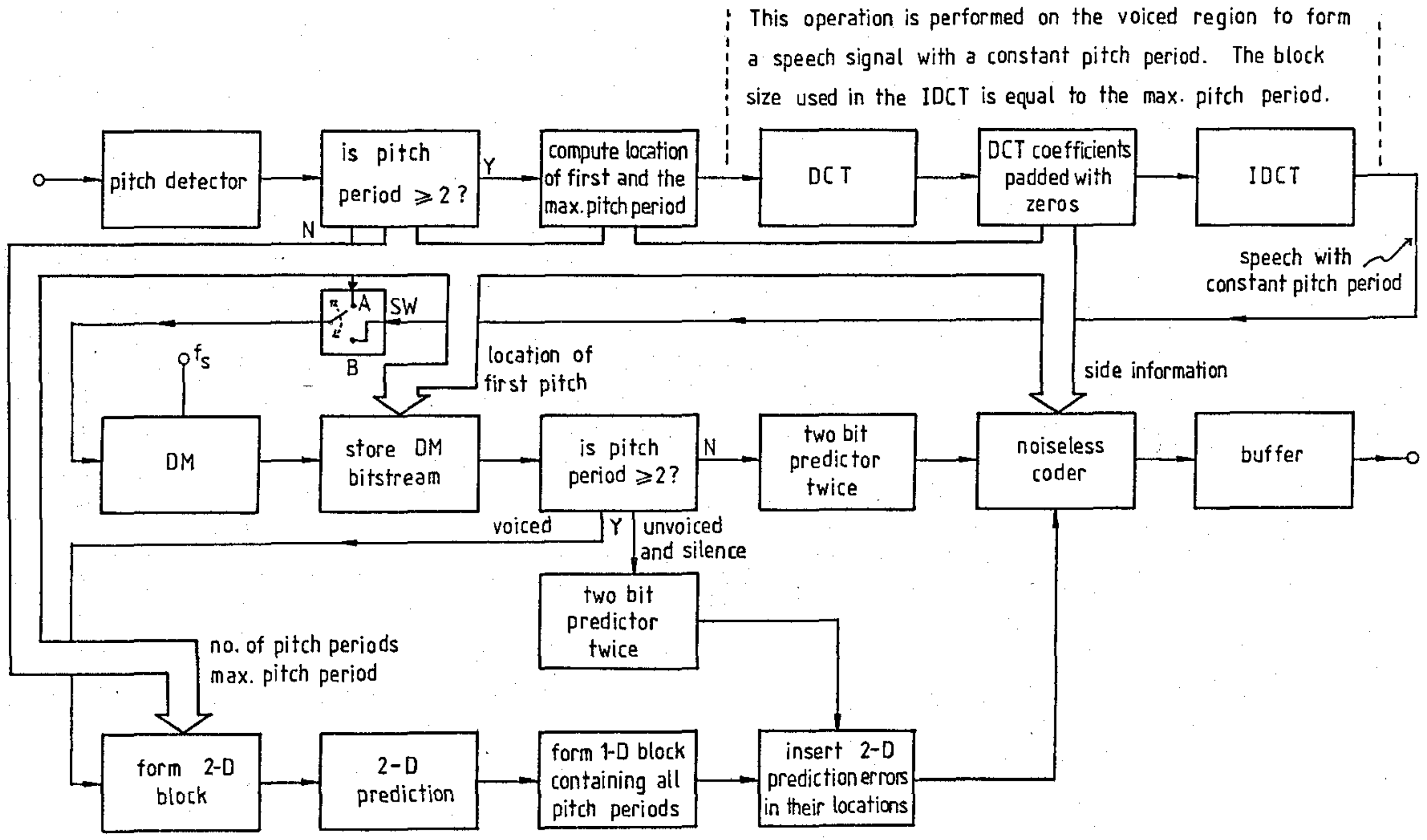
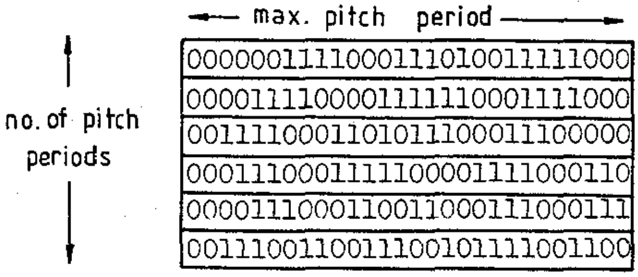

This operation is performed on the voiced region to form a speech signal with a constant pitch period. The block size used in the IDCT is equal to the max. pitch period.
pitch detector
is pitch period ⩾ 2 ?
Y
N
compute location of first and the max. pitch period
DCT
DCT coefficients padded with zeros
IDCT
speech with constant pitch period
A
SW
B
f_s
location of first pitch
side information
DM
store DM bitstream
is pitch period ⩾2?
N
two bit predictor twice
noiseless coder
buffer
voiced
Y
unvoiced and silence
two bit predictor twice
no. of pitch periods
max. pitch period
form 2-D block
2-D prediction
form 1-D block containing all pitch periods
insert 2-D prediction errors in their locations

Fig. 8.11 - Block Diagram of Two-Dimensional DM System

and the maximum pitch period are computed. A check is also made to ensure that the pitch period has duration between 4 ms and 16 ms. Four milliseconds is used to limit the detection of pitch frequencies to those below 250 Hz(131) and a pitch period within a continuously voiced interval cannot be separated by more than 16 ms. Pitch periods having a duration of greater than 16 ms indicate a voicing discontinuity.

In order to form the two-dimensional 'bit sequence image', each pitch period has to be of equal size and this is carried out with the aid of the Discrete Cosine Transform (DCT)(91-93). Having obtained the maximum pitch period for a particular speech segment, the DCT is applied to the speech samples contained within each pitch period. The block size used for the forward transformation is equal to the individual pitch period itself. As a result, a set of DCT coefficients is generated which is then padded with zeros to form a block size equal to that of the maximum pitch period. (The number of zeros added is transmitted as side information). The inverse DCT operation is then performed, in which the block size is equal to the maximum pitch period, resulting in speech segments of equal pitch period. These are then fed to the DM systems, whose sampling frequency is f_s kHz, with the output DM bitstream stored for further reversible processing.

Knowing the location of the first pitch period, the number of pitch periods, the maximum pitch period and having made all pitch periods of equal size in a particular speech segment, the two-dimensional 'bit sequence image' (similar to a facsimile image) can now be formed as shown in Fig. 8.12.



00000011110001110100111111000
00001111000011111110001111000
00111110001101011100011100000
00011100011111100001111000110
0000111000110011000111000111
00111001100111001011111001100

Fig. 8.12 - 'Bit sequence image'

With the formation of the 'bit sequence image', a two-dimensional predictor can be applied where the reference elements are as shown in Fig. 8.13.

X_4	X_3	X_2
X_1	X_0	

Fig. 8.13 - Reference elements for 2-D prediction

The present bit X_0 is predicted based on the previous four bits of the same pitch period (X_1) and the previous pitch period (X_2, X_3, X_4). The prediction function for each DM coder used is optimised and the application of the 2-D predictor generates a 'prediction error image' of the DM bitstream after which a 1-D block, containing the 'predicted bits' from all the pitch periods, is formed. The remaining DM bitstream is one-dimensionally processed using a two bit predictor twice. The 'predicted bits' for the one and two-dimensional situations are then located in their apppopriate places.

Conceptually, 2-D processing is applied to the voiced region of the speech segment whilst 1-D processing is applied to silence and unvoiced regions as shown in Fig 8.14.

The location of the first pitch period, the duration of the maximum pitch period, the number of pitch periods and the number of zeros padded during the DCT operation for each pitch period are transmitted as side information in order for the receiver faithfully to reconstruct the speech signal. All this side information, together with the prediction errors generated during the one and two-dimensional processing, are coded using the noiseless coding algorithm described earlier. A buffer is required to ensure that constant rate transmission is maintained.

The results of applying Adaptive Block/Run-Length Coding to the two-dimensional DM system are shown in Fig. 8.7 for LDM (64 kHz), Figs. 8.8(a) and 8.8(b) for CFDM (16 and 32 kHz), and Figs. 8.9(a) and 8.9(b) for CVSD (16 and 32 kHz). From the graphs, it is observed that only the two-dimensional DM system employing CFDM performs better than its one-dimensional counterparts. There is an improvement of about 4 and 5 percent respectively, for CFDM operating at 16 and 32 kHz compared with the one-dimensional case. When LDM and CVSD were used in the two-dimensional DM system, there is a decrease in performance compared with the one-dimensional case. The decrease is about 21 percent for LDM and for CVSD, about 0.7 and 12 percent at sampling frequencies of 16 and 32 kHz respectively. This may be due to the fact that both these coders were optimised in the voiced regions and as such slope overload in these regions

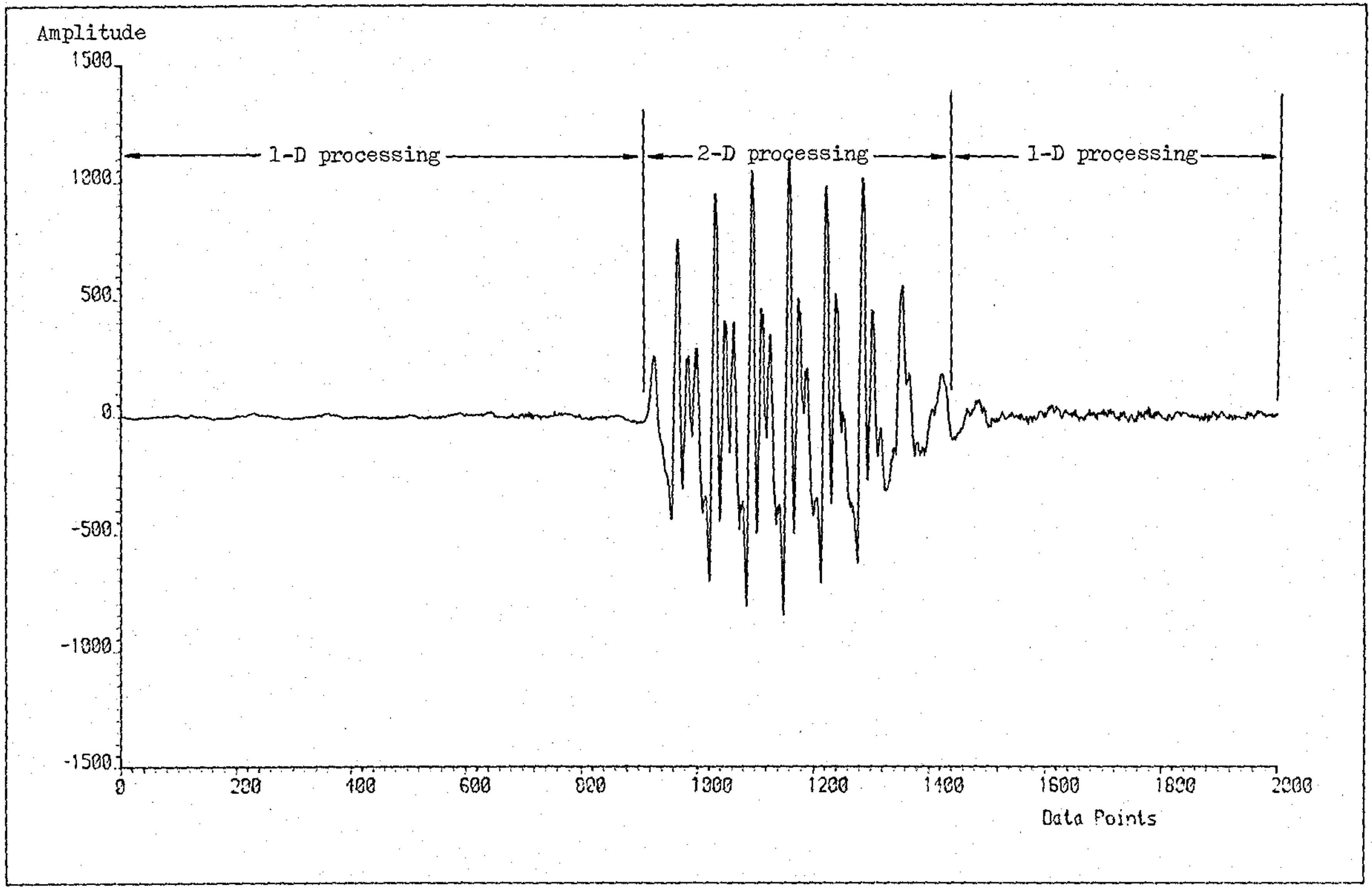


Fig. 8.14 - One and two-dimensional processing situations

is not pronounced. The occurrence of slope overload means that the DM bitstream generated consists primarily of long sequences of 1's and 0's which are correlated between one pitch period and another, and this condition is especially advantageous when 2-D prediction is applied. Such is not the case for LDM and CVSD. As for CFDM, optimisation of the coder might have occurred in the unvoiced region and this will create slope overload in the voiced region, producing long sequence of 1's and 0's. The high amount of side information transmitted may be another contributory factor to the poor all round performance of the two-dimensional DM system, particularly in the case of LDM and CVSD

8.6. CONCLUSION

The possibility of reducing the average transmission bit rate for Delta Modulators through the use of entropy coding has been investigated in this chapter. Theoretical entropy measurements obtained reveal that substantial transmission rate reductions can be achieved. Adaptive Block/Run-Length Coding, a noiseless coding algorithm, for efficiently coding discrete data sources with partially known probability ordering, but unknown values, has also been described. Using this algorithm and applying a two bit predictor twice for the one-dimensional DM system results in a dramatic reduction in average transmission bit rate from 64 kbits/s to 17.6 kbits/s for LDM, from 32 kbits/s to 25.03 kbits/s and 12.42 kbits/s for CFDM and CVSD and from 16 kbits/s to 13.23 kbits/s and 6.48 kbits/s for CFDM and CVSD respectively, whilst still maintaining the quality of speech of the DMs operating at the original sampling frequencies. The more complicated two-dimensional DM system using this technique is also described, and gives further transmission rate reduction in the case of CFDM but not with LDM and CVSD.

CHAPTER IX

RECAPITULATION AND CLOSING COMMENTS

9.1. INTRODUCTION

The objective of the research reported here is to digitally encode black and white facsimile signals with the aim of achieving as low a transmission time as possible for an A4 size document conveyed over the public switched telephone network. In pursuing the desired aim, in order to reduce the high volume of data required to represent an A4 size document, two possible avenues, which can be broadly categorised as preprocessing and data compression techniques, have been considered. On the one hand, preprocessing is basically a non-information preserving approach to facsimile coding in which a proper post-processing technique is subsequently required to restore the received document as closely to the original as possible, whilst on the other hand, the remaining techniques described in this thesis are in essence information preserving, in which documents are transformed into a coded form suitable for transmission.

Initially, a simple strategy was employed to reduce the number of bits necessary for the transmission of black and white documents. An area coding algorithm (non-information preserving), it incorporates both preprocessing and data compression within one algorithm. Its success was possible due to the inherent redundancy and high two-dimensional correlation that exist in high resolution documents. This simple work was principally undertaken in order to become familiar with the statistics and properties of black and white images, and to understand the principles of some of the coding algorithms already available. Nevertheless, the performance of the scheme can be considered as highly satisfactory.

With this knowledge gained, greater emphasis was then placed on developing schemes which are highly efficient in reducing redundant information present in a document. Two pre- and post-processing techniques were investigated; the first scheme using a set of masks and the second employing, respectively, subsampling and interpolation. The latter scheme is more complicated, and introduces greater degradation but results in higher compression ratios being obtained. Formal visual assessment of the restored documents confirmed that the schemes are suitable for use prior to the actual coding process. The potential of preprocessing was soon realised and pre- and post-processing with masks was used in conjunction with Classified Adaptive Block/Run-Length Coding, whilst subsampling and interpolation, respectively, were coupled with Adaptive Block/Location Coding to form a basic system. The Group 3 facsimile coding standards require that the transmission of A4 documents over the telephone network (at a rate of 4800 bits per second) take less than one minute, and the two coding schemes described in this thesis were successful in achieving the desired target.

The binary output bitstream resulting from the encoding of speech using delta modulation resembles that of a facsimile signal and thus offers an opportunity for transmission rate reduction using facsimile techniques. Preliminary research that has been carried out, (though more work is required), using Adaptive Block/Run-Length Coding, indicates that a possible transmission rate reduction can be achieved provided that suitable preprocessing is employed.

In the following sections, a recapitulation of the main results obtained during the course of the research is made.

9.2. RECAPITULATION

9.2.1 Adaptive Pel Location Coding

Adaptive Pel Location Coding (APLC) is a simple technique for encoding black and white images where the number of bits transmitted varies as a function of the local distribution of the black and white pels in the document. Operating in one- and two-dimensions, isolated black or white pels are discarded to increase coding efficiency. This is made possible by the fact that most documents are inherently predominantly white. At the same time, the high two-dimensional correlation that exists is exploited.

The scheme is information lossy and as expected, degradations are observed in the decoded image though intelligibility is maintained. The degree of degradation is greater in the one-dimensional case than in its two-dimensional counterpart, but suitable post-processing schemes can be applied to improve image quality. It has been shown that one-dimensional APLC is as efficient as the Modified Huffman Code. However, it is with the use of two-dimensional APLC that significant improvements in compression ratios were achieved compared with the other encoding methods that were examined, particularly when documents with low complexity were used. No comparison with the current two-dimensional CCITT coding standard (Modified READ code) was made mainly because, at the time the work was carried out, the scheme was still under review.

9.2.2 Pre- and Post-Processing

In Chapter V, two preprocessing schemes, resulting in more efficient coding, were described. Greater efficiency is achieved by removing redundant information like notches and pinholes or random isolated pels, caused by improper decision during the scanning process. The first strategy employed, a simple one, uses a set of 3x3 masks which is sequentially applied to the source image. The value of the central picture element is altered from black to white or vice versa provided the surrounding elements correspond to a certain predetermined pattern. The process carried out preserves connectivity and contour direction, and at the same time, introduces minimal degradation to the original document and does not impair intelligibility. At the receiver, another set of 3x3 restoration masks is employed in order to restore the 'naturalness' of the received documents. A theoretical analysis of the preprocessing scheme, beginning with the concept of connectivity and connected numbers, was followed by entropy measurements indicating the efficiency of such a scheme. The results obtained showed an improvement of about 24 percent compared with the original and about 14 percent compared with Majority Logic Smoothing with Contour Preservation(15). By observing displayed, processed, images, it is evident that the proposed scheme is highly successful in removing redundant information without disturbing the connectivity and the essential details of the image.

Subsampling and interpolation as a means of pre- and post-processing respectively, is the second scheme studied. In this method of preprocessing, severe degradations were introduced with efforts directed towards maintaining features which preserve the intelligibility of the resulting image, and yet produce high compression ratios. These objectives were achieved by first removing notches and pinholes in the original image using the set of masks described in Section 5.2. Single element runs were then doubled to preserve connectivity, after which subsampling was carried out. Different methods of subsampling were investigated including subsampling by the use of the Hadamard and Cosine Transforms but the simplest, and the most effective method was subsampling by taking alternate picture elements. Observation of the subsampled images revealed that connectivity is preserved and that the loss of information is minimal.

Since subsampling is an information lossy approach to preprocessing in which the area of the original document is reduced by a factor of four, enlargement and restoration of the subsampled documents are required at the receiver. Three schemes, namely replication, bilinear and Cubic B-spline interpolation with suitable thresholding, were examined in order to overcome the loss of resolution which occurred during subsampling. The edges of the interpolated images were then 'smoothed out' by a set of restoration masks producing visually acceptable results. In order to assess the degree of degradation introduced by subsampling in the restored documents using the three interpolation schemes, a formal subjective test was carried out. The test revealed, however, that the order of preference for text predominant documents was different from that for documents containing mainly graphics. For test predominant documents, enlargement of the subsampled image using bilinear interpolation was the one most favoured, while for graphics-type documents, replication with proper restoration produced the most 'cosmetically' pleasing effect. The subjective test also revealed a decrease in quality of about 20 percent when the most preferred image was compared with the original. This relatively small reduction in quality is the price to be paid for the high compression ratios achieved. The main drawback of subsampling and interpolation as pre- and post-processing steps, respectively, is the relatively large amount of memory required to store the documents while processing is taking place. However, with the costs of memory at present decreasing faster than that of transmission, the problem is not expected to be a major one.

9.2.3 Data Compression

Two approaches to data compression are employed in this work, both of which are adaptive in their own way. The first, a line-by-line sequential coding scheme called Classified Adaptive Block/Run-Length Coding (CABC), uses preprocessing by masks as a prerequisite to coding, whilst the second, called Adaptive Block/Location Coding (ABLC), an area coding algorithm, is employed in conjunction with subsampling.

CABC is based on the observation that a facsimile signal can be regarded as the output of an Nth order Markov source and that the statistics of the run-lengths conditioned on each of the states of the Markov source are quite different. Exploiting the vertical correlation between the current and the preceding line, the approach taken to reduce redundancy effectively is to apply a 7th order Markov model predictor, in which the prediction errors generated are assumed to emanate from a memoryless binary source. Entropy measurements were made and, further to improve the performance of CABC, the scan-lines containing prediction errors are classified into two states called the Greater Error State and the Lesser Error State. These states are adaptively divided into smaller blocks of which the maximum block size is M, where M is obtained using the probability of correct prediction as a criterion. Initially, the coding strategy and the value of M determined, was decided by developing an equation matched to the probability distribution of the number of prediction errors per scan-line. Its optimality was later confirmed using computer simulation. The compression ratios obtained when CABC is applied to the documents following preprocessing by the set of masks range from 9.4 to 49.4. This represents an improvement of about 13 to 20 percent when compared with the original and about 41 to 61 percent when compared with run-length coding using the Modified Huffman Code. A further improvement in compression ratios, though marginal, was observed when a combination of Huffman code and CABC was employed.

No investigation of facsimile compression would be complete without a study of the effect of transmission errors on the decoded image. As with most two-dimensional coding schemes, the effect of transmission errors on CABC is to cause error propagation. In order to simulate random and short burst errors, single and double errors were introduced, respectively, and the Error Sensitivity Factor (ESF) measured. From the numerical results and visual illustrations obtained, it is obvious that CABC is more resilient to double errors than single errors, and that the former are easier to deal with than the latter.

The second approach to data compression, ABLC, made use of a numerical measure of complexity as a criterion for the coding of subsampled documents. Complexity is chosen as a criterion because it has been mathematically shown (see Chapter VII) to be related to the number of bits necessary for transmission; the lower the complexity, the smaller is the number of bits required. A theoretical analysis of block and location coding based on complexity has been given in this thesis. In order to reduce complexity, the 7th order Markov model predictor was found to be effective for this purpose. Four variations of ABLC were examined and the application of the one giving the best performance (System D, see Section 7.4.4) to seven subsampled CCITT documents using bit assignment Ⓑ yielded compression ratios ranging from 13.35 to 64.06.

The effect of transmission errors on ABLC, in particular on System C (see Section 7.4.3) was also investigated. No attempt was made to measure the Error Sensitivity Factor, rather the effects of single transmission errors occurring at random on the decoded image, and the extent of damage caused due to loss of synchronisation were observed. From visual observations, the effect of errors is quite devastating, and as the error rate is increased, the degradation becomes catastrophic. To limit this catastrophic effect, a unique 'end of IPB' codeword is included, and this improves the decoded image considerably.

The use of subsampling, as a preprocessing step, on CCITT one- and two-dimensional coding standards has also been investigated in this thesis, to see what gain in compression ratio or reduction in transmission time could be achieved if ever subsampling were incorporated into the standards. Results obtained show a remarkable improvement in compression ratio, ranging from 11.49 to 39.13 for Modified Huffman Code (MHC) and from 14.04 to 76.93 for Modified READ Code (MRC). The average transmission times for the seven subsampled CCITT documents are 45.2 seconds for MHC and 23.8 seconds for MRC. This, on average, corresponds to an increase in efficiency of more than 50 percent compared with the original documents.

9.2.4 Speech Coding Using a Facsimile Technique

This work is a slight departure from the main theme of the thesis. Its intention is to assess the efficiency of Adaptive Block/Run-Length Coding when applied to other data sources. As it turns out, the output of a delta modulator is particularly amenable to further coding and, with suitable preprocessing, the application of Adaptive Block/Run-Length Coding can reduce the transmission rate further.

LDM, CFDM and CVSD were the three coding methods studied, and substantial transmission rate reduction were achieved. The two-dimensional DM system, making use of the quasiperiodic nature of speech, was also examined, with somewhat disappointing results. Nevertheless, the idea was novel. The research is still in its preliminary stages, requiring further investigations on buffer problems and the effects of transmission errors.

9.3. CLOSING COMMENTS

The basic objective of the research reported in this thesis, viz. digital encoding of black and white facsimile signals, has been fulfilled. Successful practical implementation of the schemes described might make the research more complete, but this has purposely been omitted due to lack of both finance and time.

The work described in this thesis can also be applied to other areas, for example, coding dithered images and colour facsimile. Although the demand for colour facsimile is not great at the present moment, it will, in the author's opinion, increase rapidly in the not too distant future and become a commercial reality. With the coming of public data network, employing packet switching, another area of research worth considering is the effect of packet losses on the decoded images using the current CCITT coding standards.

Finally, it is of interest to compare the data compression schemes studied with the CCITT coding standards. Using the subsampled CCITT documents instead of the originals, the compression ratios of the various schemes (for MHC, MRC, and CABC, a 12 bit EOL codeword is used, K factor taken to be infinity and MSLT set to 0 ms), are shown in Table 9.1. The values speak for themselves.

It is hoped that the contribution made by this thesis to the current state-of-the-art may have laid a foundation for fruitful research in the future.

DOC. NO.	MHC	MRC	CABC	ABLC - SYSTEM D (32 x 32)	
				Bit Assignment Ⓐ	Bit Assignment Ⓑ
1	34.52	48.34	48.54	47.36	48.19
2	39.13	76.93	64.98	63.53	64.06
3	19.54	32.66	35.20	33.56	34.46
4	11.75	14.14	17.11	12.72	13.35
5	19.37	29.42	32.41	29.73	30.53
6	24.69	53.75	52.66	60.60	60.86
7	11.49	14.04	16.68	14.74	15.34

Table 9.1 - Comparison between the performance of data compression methods studied and the CCITT coding standards.

APPENDIX A

Eqn. (6.17) is obtained as follows:-

The derivation of Eqn. (6.17) uses Figs. 6.5(d) and 6.5(e) as an example, in which the prediction error domain is categorised into the Lesser Error State (LES) and the Greater Error State (GES).

Let p_1 and p_2 be the probabilities of prediction errors which have been classified into the Lesser Error State and the Greater Error State, and p_3 and p_4 are the probabilities of correct prediction in the LES and GES respectively. In the example of Figs. 6.5(d) and 6.5(e), $p_1 = \frac{1}{32}$, $p_2 = \frac{4}{32}$, $p_3 = \frac{13}{32}$, and $p_4 = \frac{14}{32}$. This situation can be represented in a form of a tree as shown in Fig. A1 below.

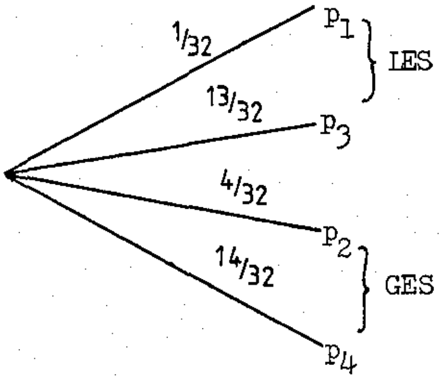


Fig. A1

The entropy is therefore:-

$$H(p_1,p_2,p_3,p_4) = - \sum_{i=1}^{4} p_i \log_2 p_i \qquad (A1)$$

The tree can be represented in another form as shown in Fig. A2:-

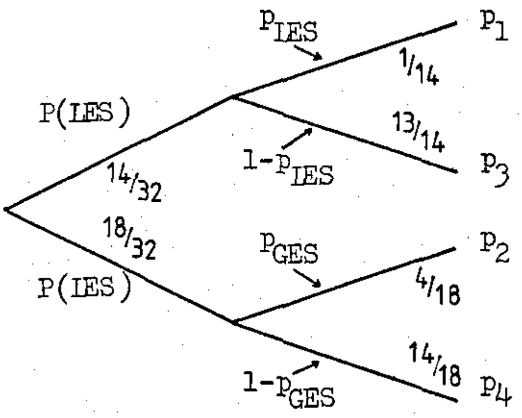


Fig. A2

The entropy now becomes:-

$$H(p_1,p_2,p_3,p_4) = P(LES)\,H(p_{LES},1-p_{LES}) + P(GES)\,H(p_{GES},1-p_{GES}) \qquad (A2)$$

where $H(p_{LES},1-p_{LES})$ and $H(p_{GES},1-p_{GES})$ are the entropies of the Lesser Error State and the Greater Error State respectively. The weighting factors, P(LES) and P(GES) = 1-P(LES), are introduced because the LES and GES only occur for, respectively, P(LES) and P(GES) of the time.

Expanding Eqn. (A2):-

$$\begin{aligned} H(p_1,p_2,p_3,p_4) = &-P(LES)\left[p_{LES}\log_2 p_{LES} + (1-p_{LES})\log_2(1-p_{LES})\right] \\ &-P(GES)\left[p_{GES}\log_2 p_{GES} + (1-p_{GES})\log_2(1-p_{GES})\right] \qquad (A3) \end{aligned}$$

where P(LES) is defined as the probability of the prediction domain being in the Lesser Error State and p_{LES} is the probability of prediction errors in the Lesser Error State. P(GES) and p_{GES} are similarly defined.

From the example of Figs. 6.5(d) and 6.5(e), $p_{LES} = \frac{1}{14}$, $p_{GES} = \frac{4}{18}$, $P(LES) = \frac{14}{32}$ and $P(GES) = \frac{18}{32}$. Fig. A1 is equivalent to Fig. A2 and hence the entropy of Eqn. (A1) is also equivalent to the entropy of Eqn. (A3), where it is the weighted sum of the individual entropies of the Lesser Error State and the Greater Error State.

REFERENCES

1. D.M. COSTIGAN: 'Electronic Delivery of Documents and Graphics' Van Nostrand Reinhold Ltd., 1978, New York.

2. R.B. ARPS: 'Binary Image Compression', Chap.7 in Image Transmission Techniques. New York, Academic Press, 1979, pp. 219-276.

3. C.E. SHANNON: 'The Mathematical Theory of Communication' BSTJ, Vol.27, 1948, pp. 379-423, 623-656.

4. A.E. LAEMMEL: 'Coding Processes for Bandwidth Reduction in Picture Transmission', Rep. R-246-51, PIB-187, Microwave Res. Inst., Polytechnic Institute of Brooklyn, New York, 1951.

5. P. ELIAS: 'Predictive Coding', IRE Trans. on Information Theory, IT-1, March 1955, pp. 19-33.

6. S. DEUTSCH: 'A Note on Some Statistics Concerning Typewritten or Printed Material', IRE Trans. on Information Theory, IT-3, June 1957, pp. 147-148.

7. W.S. MICHEL: 'Statistical Encoding for Text and Picture Communication', Commun. and Electronics, Vol. 35, March 1958, pp. 33-36.

8. W.S. MICHEL, W.O. FIECKENSTEIN and E.R. KRETZMER: 'A Coded Facsimile System', 1957 WESCON Convention Record, Pt. 2, pp. 84-93.

9. H. WYLE, T. ERB and R. BANOW: 'Reduced-Time Facsimile Transmission by Digital Coding', IRE Trans. on Commun. Systems, CS-9, September 1961, pp. 215-222.

10. J. CAPON: 'A Probabilistic Model for Run-Length Coding of Pictures', IRE Trans. on Information Theory, IT-5, September 1959, pp. 157-163.

11. J.S. WHOLEY: 'The Coding of Pictorial Data', IRE Trans. on Information Theory, IT-7, April 1961, pp. 99-104.

12. R.B. ARPS: 'Biblography on Digital Graphic Image Compression and Quality', IEEE Trans. on Information Theory, Vol. IT-20, No. 1, January 1974, pp. 120-122.

13. R.B. ARPS: 'Bibliography on Binary Image Compression', Proc. IEEE, Vol.68, No. 7, July 1980, pp. 922-24.

14. W.K. PRATT: 'Image Transmission Techniques', Chap. 1, in Image Transmission Techniques, New York, Academic Press, 1979, pp. 1-19.

15. D. TING and B. PRASADA: 'Preprocessing Techniques for Digital Facsimile', ICC '78, Conference Record, Toronto, June 4-7, Vol. 3, pp. 48.5.1-6.

16. D. TING and B. PRASADA: 'Digital Processing Techniques for Encoding of Graphics', Proc. IEEE, Vol. 68, No. 7, July 1980, pp. 757-769.

17. M.G.B. ISMAIL and R.J. CLARKE: 'New Preprocessing Technique for Digital Facsimile Transmission', Electronics Letters, 8th May 1980, Vol. 16, No. 10, pp. 455-456.

18. A.N. NETRAVALI, F.W. MOUNTS and K.A. WALSH: 'Adaptation of Ordering Techniques for Facsimile Pictures with No. Single Element Runs', BSTJ, Vol. 58, No. 4, April 1979, pp. 857-865.

19. V. MARGNER and P. ZAMPERONI: 'Data Reduction for Print and Graphics by Means of Feature Reduction', Nachrichtentech. Z, Vol. 29, No. 7(1976), pp. 525-530.

20. Y. YASUDA and K. KOGA: 'Bandwidth Compression Multiplex Transmission System for Facsimile Signals by Pseuodorandom Interchange of Pixel Order', J. IIEE (Jap.) Vol. 1, No. 1, June 1972, pp. 6-17.

21. T. KAWADE and T. NAKAGAWA: 'Effects of Reducing and Bridging Isolated Picture Elements in Coding of Facsimile Signals', 1976, Nat. Conv. Rec., IECE Japan, No. 1027.

22. M. KUNT and O. JOHNSEN: 'Block Coding of Graphics: A Tutorial Review', Proc. IEEE, Vol. 68, No. 7, July 1980, pp. 770-786.

23. T. FUKINUKI: 'Notchless Bilevel Quantizer for Facsimile and Its Effect on Coding Efficiency', IEEE Trans. on Communications, Vol. COM-26, No. 5, May 1978, pp. 611-618.

24. T.L. BILLINGS: 'Pre-Transmission Enhancement of Facsimile Images', IBM Tech. Disclosure Bulletin, Vol. 21, No. 8, January 1979, pp. 3086-3087.

25. V. MARGNER and P. ZAMPERONI: 'Transmission of Binary Pictures with Constant Rate by Source Encoding using a Morphological Irrelevance Criterion', Frequenz, Vol. 31, No. 7, July 1977, pp. 221-227.

26. M. TAKAGI and T. TSUDA: 'Bandwidth Compression for Facsimile Using Signal Modification', Electronics and Commun. in Japan, Vol. 60-A, No. 2, 1977, pp. 9-16.

27. T. USUBUCHI, S. MIZUNO and K. IINUMA: 'Efficient Facsimile Signal Data Reduction by using a Thinning Process', NTC '77, Conference Record, December 5-7, pp. 49.2.1-6.

28. I.D. JUDD: 'Compression of Binary Images by Stroke Encoding', IEE Proc. on Computers and Digital Techniques, Vol. 2, No. 1, February 1979, pp. 41-48.

29. R. STEFANELLI and A. ROSENFELD: 'Some Parallel Thinning Algorithms for Digital Pictures', J. ACM, Vol. 18, No. 2, 1971, pp. 255-264.

30. W. CHEN, J.L. DOUGLAS and R.D. WIDERGREN: 'Combined Symbol Matching - A New Approach to Facsimile Data Compression', Proc. Soc. Photo-Optical Instrumentation Engineers, Vol. 199, August 1978, pp. 2-9.

31. W. CHEN, J.L. DOUGLAS, W.K. PRATT and W.H. WALLIS: 'A Dual Mode Hybrid Compressor for Facsimile Images', Proc. Soc. Photo-Optical Instrumentation Engineers, Vol. 207, 1979, pp. 226-232.

32. W.K. PRATT, P.J. CAPITANT, W. CHEN, E.R. HAMILTON, R.H. WALLIS: 'Combined Symbol Matching Facsimile Data Compression System', Proc. IEEE, Vol. 68, No. 7, July 1980, pp. 786-796.

33. K. RAMACHANDRAN: 'Coding Method for Vector Representation of Engineering Drawings', Proc. IEEE, Vol. 68, No. 7, July 1980, pp. 813-817.

34. F. DECOULON and M. KUNT: 'An Alternative to Run-Length Coding of Black and White Facsimile', Proc. 1974 International Zurich Seminar on Digital Communications, Zurich, Switzerland, March 1974.

35. F. DECOULON and O. JOHNSEN: 'Adaptive Block Scheme for Source Coding of Black and White Facsimile', Electronics Letters, Vol. 12, No. 3, February 1976, pp. 61-2.

36. D.A. HUFFMAN: 'A Method for the Construction of Minimum Redundancy Codes', Proc. IRE, Vol. 40, Sept. 1952, pp. 1098-1101.

37. H. MEYR, H.G. ROSDOLSKY and T.S. HUANG: 'Optimal Run-Length Codes', IEEE Trans. on Communications, Vol. COM-22, No. 6, 1974, pp. 826-835.

38. U. ROTHGORDT: 'Run-Length Coding Method for Black and White Facsimile with a Ternary Code as an Intermediate Step', Electronics Letters, 6 March 1975, Vol. 11, No. 5, pp. 101-102.

39. U. ROTHGORDT and G. RENELT: 'Intermediate Ternary Code: A Redundancy Reducing Run-Length Code for Digital Facsimile Transmission', Electronics Letters, 24 November 1977, Vol. 13, No. 24, pp. 749-750.

40. G. RENELT: 'Easily Decodable Run-Length Code(EDC) for Source Encoding of Black and White Facsimile Pictures', Electronics Letters, 11 November 1976, Vol. 12, No. 23, pp. 633-634.

41. H.G. MUSMANN and D. PREUSS: 'Comparison of Redundancy Reducing Codes for Facsimile Transmission of Documents', IEEE Trans. on Communications, Vol. COM-25, No. 11 November 1977, pp. 1425-1432.

42. U. ROTHGORDT, G. AARON and G. RENELT: 'One-Dimensional Coding of Black and White Facsimile Pictures', Acta Electronica, Vol. 21, No. 1, 1978, pp. 21-37.

43. R. HUNTER and H. ROBINSON: 'International Digital Facsimile Coding Standards', Proc IEEE, Vol. 68, No. 7, July 1980, pp. 854-867.

44. T. HIGETA et. al. : 'Band Compression for High Speed Facsimile System', FUJITSU, Vol. 22, No. 7, 1972, pp. 57.

45. Y. YAMAZAKI et. al. : 'Run-Length Coding by Simultaneous Coding of Scanning Lines', Paper of Tech. Group IECE Japan, Vol. CS74, No. 116, November 1974.

46. R. OHNISHI et. al. : 'Facsimile Bandwidth Compression Scheme Using Vertical Correlation', Nat. Conv. Rec. Inst. Television Eng. Japan, No. 2-8, 1973.

47. T. USUBUCHI, K. IINUMA and T. ISHIGURO: 'Data Compression of Facsimile Signals by Cascade Division Coding', ICC '75, Conference Record, San Franscisco, June 1975, pp. 7.22.

48. S. FURUKAWA: 'Administrative Standard Facsimile Equipment', J. Image Electron. Eng. Japan, Vol. 6, No. 3, September 1977, pp. 123-128.

49. KALLE INFOTEC U.K. LTD. : 'Redundancy Reduction Technique for Fast Black and White Facsimile Apparatus', CCITT Study Group XIV, Temp. Document, No. 15, April 1975.

50. P.K.S. WAH: 'Simple Adaptive Vector Encoding for Two-Dimensional Redundancy Reduction (SAVER)', Electronics Letters, 12 April 1979, Vol. 15, No. 8, pp. 221-222.

51. Y YASUDA: 'Overview of Digital Facsimile Coding Techniques in Japan', Proc. IEEE, Vol. 68, No. 7, July 1980, pp. 830-45.

52. T.S. HUANG: 'Run-Length Coding and Its Extension', in Picture Bandwidth Compression, Ed. by T.S. Huang and O.J. Tretaik, Gordon and Breach, 1972.

53. T.S. HUANG: 'Coding of Two-Tone Images', IEEE Trans. on Communications, Vol. COM-25, No. 11, November 1977, pp. 1406-1424.

54. Y. YAMAZAKI, Y. WAKAHARA and H. TERAMURA: 'Digital Facsimile Equipment "Quick-FAX" using a New Redundancy Reduction Technique', NTC '76, Conference Proceedings, Nov. 29-Dec. 1, 1976, pp. 6.2.1-6.2.5.

55. T. YAMADA: 'Edge Difference Coding Scheme for Facsimile Redundancy Reduction', Paper of Technical Group on Image Engineers, Inst. of Electronic and Communication Engineers of Japan, IE76-79, December 1976.

56. T. YAMADA: 'Edge Difference Coding - A New, Efficient Redundancy Reduction Technique for Facsimile Signals', IEEE Trans. on Communications, Vol. COM-27, No. 8, August 1979, pp. 1210-1217.

57. JAPAN: 'Proposal for Draft Recommendation of Two-Dimensional Coding Scheme', CCITT SG. XIV, Doc. No. 42, Aug. 1978.

58. J.W. WOODS: 'Two-Dimensional Delta-Mod Facsimile Coding', IEEE Trans. on Communications, Vol. COM-26, No. 6, June 1978, pp. 936-939.

59. J.L. MITCHELL and G. GOERTZEL: 'Two-Dimensional Facsimile Coding Scheme', ICC '79, Conference Record, Pt. I, Boston, USA, 10-14 June 1979, pp. 8.7.1-8.7.5.

60. D. PREUSS: 'Comparison of Two-Dimensional Facsimile Coding Scheme', ICC '75, Conference Record, San Francisco, June 1975, pp. 7.12-7.16.

61. A.N. NETRAVALI, F.W. MOUNTS and E.G. BOWEN: 'Ordering Techniques for Coding of Two-Tone Facsimile Pictures', BSTJ, Vol. 55, No. 10, December 1976, pp. 1539-1552.

62. F.W. MOUNTS, A.N. NETRAVALI and K.A. WALSH: 'Some Extensions of the Ordering Techniques for Compression of Two-Level Facsimile Pictures', BSTJ, Vol. 57, No. 8, October 1978, pp. 3057-3067.

63. F.W. MOUNTS, E.G. BOWEN and A.N. NETRAVALI: 'An Ordering Scheme for Facsimile Coding', BSTJ, Vol. 58, No. 9, November 1979, pp. 2113-2128.

64. A.N. NETRAVALI AND F.W. MOUNTS: 'Ordering Techniques for Facsimile Coding - A Review', Proc. IEEE, Vol. 68, July 1980, pp. 796-807.

65. Y. UENO and F. ONO: 'Data Compression for Facsimile Transmission using the Classified Pel Pattern Method', Mitsubishi Electron. ADVANCE, Vol. 3, No. 3, 1978, pp. 21-23.

66. Y. UENO, F. SEMASA, S. TOMITA and R. OHNISHI: 'Comparison of Facsimile Data Compression Methods', ICC '78, Conference Record, Pt. III, June 4-7, pp. 48.4.1-6.

67. J.M. WHITE: 'Recent Advances in Thresholding Techniques for Facsimile', Journal of Applied Photographic Engineering, Vol. 6, No. 2, April 1980, pp. 49-57.

68. T.S. HUANG and A.B.S. HUSSAIN: 'Facsimile Coding by Skipping White', IEEE Trans. on Communications, Vol. Com-23, December 1975, pp. 1452-1466.

69. E. KAWAGUCHI and T. ENDO: 'On a Method of Binary-Picture Representation and its Application to Data Compression', IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. PAMI-2, No. 1, Jan 1980, pp. 27-35.

70. W.F. SCHREIBER, T.S. HUANG and O.J. TRETAIK: 'Contour Coding of Images', IRE Western Electron Show and Conv., Conference Record, August 1968, pp. 8.3.1-8.3.6.

71. T.H. MORRIN: 'Recursive Contour Coding of Nested Objects in Black and White Images', ICC '75, Conference Record, June 16-17, pp. 7.17-7.21.

72. A.N. ASCHER and G. NAGY: 'A Means of Achieving a High Degree of Compaction on Scan-digitized Text', IEEE Trans. on Computers, Vol. C-23, No. 11, November 1974, pp. 1174-1179.

73. J.O. LIMB: 'Design of Dither Waveforms for Quantized Visual Signals', BSTJ, Vol. 48, No. 7, September 1969, pp. 2555-2582.

74. J.F. JARVIS and C.S. ROBERTS: 'A New Technique for Displaying Continuous Tone Images on a Bilevel Display', IEEE Trans. on Communications, Vol. COM-24, August 1976, pp. 891-898.

75. J.F. JARVIS, C.N. JUDICE and W.H. HINKE: 'A Survey of the Techniques for the Display of Continuous Tone Pictures on Bilevel Display', Computer Graphics and Image Processing, Vol. 5, No. 1, March 1976, pp. 13-40.

76. A.N. NETRAVALI, F.W. MOUNTS and J.D. BEYER: 'Techniques for Coding Dithered Two Level Pictures', BSTJ, Vol. 56, No. 5, May-June 1977, pp. 809-819.

77. O. JOHNSEN: 'A New Code for Transmission of Ordered Dithered Pictures', BSTJ, Vol. 60, No. 3, March 1981, pp. 379-389.

78. O. JOHNSEN and A.N. NETRAVALI: 'An Extension of the CCITT Facsimile Codes for Dithered Pictures', BSTJ, Vol. 60, No. 3, March 1981, pp. 391-404.

79. T. USUBUCHI, T. OMACHI and K. IINUMA: 'Adaptive Predictive Coding for Newspaper Facsimile', Proc. IEEE, Vol. 68, No. 7, July 1980, pp. 807-813.

80. U. ROTHGORDT: 'Localization of Disturbances in Digitally Transmitted Facsimile Pictures', 1979 Picture Coding Symposium, Ipswich, England, 9-11 July, pp. 13.4.

81. J. PONCIN and J. BOTREL: 'The problems of Errors in Digital Facsimile Transmission', Ann Telecommunication, Vol. 34, No. 7-8, 1979, pp. 367-388.

82. Y. ETO, Y. HIRANO and T. FUKINUKI: 'Facsimile Signal Decoding which Localises the Effect of Run-Length Code Errors', IEEE Trans. on Communications, Vol. COM-25, No. 10, October 1977, pp. 1170-1173.

83. W. CHEN: 'Channel Error Effects and Error Protection for the Combined Symbol Matching Facsimile Coding System', Proc. Soc. Photo-Optical Instrumentation Engineers, Vol. 249, Advances in Image Transmission II, San Diego, 1980, pp. 161-168.

84. D. WIGGERT: 'Error Control Coding and Application, Artech House Inc., Dedham, 1978.

85. R.E.F. CLINT: 'Burst Error Protection for Digital Facsimile using a Microprocessor', 1979 Picture Coding Symposium, Ipswich, England, 9-11 July.

86. W.K. PRATT, W. CHEN and C. READER:'Block Character Coding', Proc. Soc. Photo-Optical Instrumentation Engineers, Vol. 66, August 1976, pp. 222-228.

87. C. ARCELLI, L. CORDELLA and S. LEVIALDI: 'Parallel Thinning of Binary Pictures', Electronics Letters, 3 April 1975, Vol. 11, No. 7, pp. 148-149.

88. A. ROSENFELD: 'Connectivity of Digital Pictures', J. ACM, Vol. 17, No. 1, January 1970, pp. 146-160.

89. K. MAITREE: 'Elimination of Redundancy', Proc. IEEE, Vol. 67, No. 8, August 1979, pp. 1166-1168.

90. K.N. NGAN and R.J. CLARKE: 'Lowpass Filtering in the Cosine Transform Domain', ICC '80, Conference Record, Seattle, Pt II, June 8-12, pp. 31.7.1-31.7.5.

91. W.K. PRATT: 'Digital Image Processing', Wiley-Interscience Publication, 1978.

92. R.C. GONZALEZ and P. WINTZ: 'Digital Image Processing', Addison-Wesley Publishing Company, Inc., 1977.

93. A. ROSENFELD and A.C. KAK: 'Digital Picture Processing' Academic Press, 1976.

94. T.N.E. GREVILLE: 'Theory and Application of Spline Functions', New York, Academic Press, 1969.

95. I.J. SCHOENBERG: 'Contribution to the Problems of Approximation of Equidistant Data by Analytic Functions', Quart. Appl. Maths, 4 (1946), Part A, pp. 45-99, Part B, pp. 112-141.

96. L.L. HOROWITZ: 'The Effects of Spline Interpolation on Power Spectral Density', IEEE Trans. Acoustic, Speech and Signal Processing, Vol. ASSP-22, February 1974, pp. 22-27.

97. T. PAVLIDIS: 'Optimal Piecewise Polynomials L_2 Approximation Functions of One and Two Variables', IEEE Trans. on Computers, Vol. C-24, January 1975, pp. 98-102.

98. M.L. LIOU: 'Spline Fit Made Easy', IEEE Trans. on Computers, Vol. C-25, May 1976, pp. 522-527.

99. L.E. OSTRANDER: 'The Fourier Transform of Spline Function Approximations to Continuous Data', IEEE Trans. on Audio Electroacoustic, Vol. AU-19, March 1971, pp. 103-104.

100. A. CAPRIHAN: 'Finite Duration Digital Filter Design by the Use of Cubic Splines', IEEE Trans. on Circuits System, Vol. CAS-22, March 1975, pp. 204-207.

101. H.C. ANDREWS and C.L. PATTERSON III: 'Digital Interpolation of Discrete Images', IEEE Trans. on Computers, Vol. C-25, No. 2, February 1976, pp. 196-202.

102. H.S. HOU and H.C. ANDREWS: 'Least Squares Image Restoration Using Spline Basis Functions', IEEE Trans. on Computers, Vol. C-26, September 1977, pp. 856-873.

103. H.S. HOU and H.C. ANDREWS: 'Cubic Splines for Image Interpolation and Digital Filtering', IEEE Trans on Acoustic, Speech and Signal Processing, Vol. ASSP-26, No. 6, December 1978, pp. 508-517.

104. C.L. JACOBSON: 'Digital Facsimile Standards', ICC '78, Conference Record, Toronto, Canada, 4-7 June Pt. III, pp. 48.2.1-48.2.3.

105. The International Telegraph and Telephone Consultative Committee, Orange Book, Vol. VII.

106. D. BODSON and R. SCHAPHORST: 'Compression and Error Sensitivity of Two-Dimensional Facsimile Coding Techniques', Proc. IEEE, Vol. 68, No. 7, July 1980, pp.846-853.

107. IBM EUROPE: 'Proposal for a Two-Dimensional Coding Scheme', CCITT SG. XIV, Doc. No. 64, January 1979.

108. 3M COMPANY: 'Results of Studies on Previously Proposed Two-Dimensional Coding Schemes', CCITT SG. XIV, Doc. No. 74, March 1979, corrigendum, July 1979.

109. AT&T: 'Proposal for a Two-Dimensional Facsimile Coding Scheme', CCITT SG. XIV, Doc. No. 81, March 1979.

110. BRITISH POST OFFICE: 'Proposal for an Optional Two-Dimensional Coding Scheme for Group 3 Facsimile Apparatus', CCITT SG. XIV, Doc. No. 77, March 1979.

111. FEDERAL REPUBLIC OF GERMANY: 'Two-Dimensional Coding Scheme', CCITT SG. XIV, Doc. No. 82, March 1979.

112. XEROX CORPORATION: 'Proposal for an Optional Two-Dimensional Coding System for Group 3 Apparatus', CCITT SG. XIV, Doc. No. 84, April 1979.

113. IBM EUROPE: 'Uncompressed Mode Enhancement to IBM Coding Scheme', CCITT SG. XIV, Doc. No. 80, March 1979.

114. D.M. COSTIGAN: 'The Principles and Practice of Facsimile Communications', New York, Chilton, 1971.

115. Y. YASUDA: Private Communication at NTC '80, Houston, Texas, in December 1980.

116. N. ABRAMSON: 'Information Theory and Coding', New York, McGraw Hill 1963.

117. J.B. THOMAS: 'An Introduction to Statistical Communication Theory', Wiley 1969.

118. R.G. GALLAGER: 'Information Theory and Reliable Communication', Wiley 1968.

119. R.W. HAMMING: 'Coding and Information Theory', Prentice-Hall 1980.

120. M.G.B. ISMAIL and R.J. CLARKE: 'Facsimile Compression Using a Classified Adaptive/Run-Length Coding', NTC '80, Conference Record, Houston, Texas, Vol. 2, 1-4 December, pp. 36.8.1-36.8.5.

121. M.G.B. ISMAIL and R.J. CLARKE: 'Adaptive Block/Location Coding of Facsimile Signals using Subsampling and Interpolation for Pre- and Post-processing', IEEE Trans. on Communications, Vol. COM-29, No. 12, December 1981, pp. 1925-1934.

122. R. STEELE: 'Delta Modulation Systems', Pentech Press, London 1975.

123. C.K. UN and H.S. LEE: 'A Study of the Comparative Performance of Adaptive Delta Modulation Systems', IEEE Trans. on Communications, Vol. COM-28, No. 1, January 1980 pp. 96-101.

124. M.R. WINKLER: 'High Information Delta Modulation', IEEE International Convention Record, Pt. 8 (1963), pp. 260-265.

125. N.S. JAYANT: 'Adaptive Delta Modulation with a One Bit Memory', BSTJ, Vol. 49, March 1970, pp. 321-342.

126. A.K. KYAW and R. STEELE: 'Constant Factor Delta Modulator', Electronics Letters, Vol. 9, 22 February 1973, pp. 96-97.

127. R.F. RICE: 'Practical Universal Noiseless Coding', Proc. Soc. Photo-Optical Instrumentation Engineers, Application of Digital Image Processing III (1979), Vol. 207, pp. 247-267.

128. J. ZIV: 'Coding of Sources with Unknown Statistics - Part I: Probability of Encoding Errors', IEEE Trans. on Information Theory, Vol. IT-18, No. 3, May 1972, pp. 384-389.

129. L.D. DAVISSON: 'Universal Noiseless Coding', IEEE Trans. on Information Theory, Vol. IT-19, November 1973, pp. 783-795.

130. R.F. RICE and J.R. PLAUNT: 'Adaptive Variable-Length Coding for Efficient Compression of Space Craft Television Data', IEEE Trans. on Communication Technology, Vol. COM-19, No. 6, December 1971, pp. 889-897.

131. N.J. MILLER: 'Pitch Detection by Data Reduction', IEEE Trans. on Acoustic, Speech and Signal Processing, Vol. ASSP-23, February 1975, pp.72-79.

132. L.R. RABINER, M.J. CHENG, A.E. ROSENBERG and C.A. MCGONEGAL: 'Comparative Performance Study of Several Pitch Detection Algorithms', IEEE Trans. on Acoustic, Speech and Signal Processing, Vol. ASSP-24, October 1976, pp. 399-418.

133. V.R. DHADESSAGOOR, C. ZIEGLER and D.L. SCHILLING: 'Delta Modulators in Packet Voice Network', IEEE Trans. on Communication, Vol. COM-28, No. 1, January 1980, pp. 33-51.