Fast Search Algorithms For Digital Video Coding

Somphob Soongsathitanon

BEng, MSc

A Thesis Submitted for the Degree of
Doctor of Philosophy
in the Faculty of Science, Agriculture and Engineering
University of Newcastle upon Tyne

NEWCASTLE UNIVERSITY LIBRARY

203 02784 5

Thesis L7660

School of Electrical, Electronic and Computer Engineering
University of Newcastle upon Tyne, UK

June 2004

ABSTRACT

Motion Estimation algorithm is one of the important issues in video coding standards such as ISO MPEG-1/2 and ITU-T H.263. These international standards regularly use a conventional Full Search (FS) Algorithm to estimate the motion of pixels between pairs of image blocks. Since a FS method requires intensive computations and the distortion function needs to be evaluated many times for each target block, the process is very time consuming. To alleviate this acute problem, new search algorithms, Orthogonal Logarithmic Search (OLS) and Diagonal Logarithmic Search (DLS), have been designed and implemented.

The performance of the algorithms are evaluated by using standard 176×144 pixels quarter common intermediate format (QCIF) benchmark video sequences and the results are compared to the traditional well-known FS Algorithm and a widely used fast search algorithm called the Three Step Search (3SS). The fast search algorithms are known as sub-optimal algorithms as they test only some of the candidate blocks from the search area and choose a match from a subset of blocks. These algorithms can reduce the computational complexity as they do not examine all candidate blocks and hence are algorithmically faster. However, the quality is generally not as good as that of the FS algorithms but can be acceptable in terms of subjective quality.

The important metrics, time and Peak Signal to Noise Ratio are used to evaluate the novel algorithms. The results show that the strength of the algorithms lie in their speed of operation as they are much faster than the FS and 3SS. The performance in speed is improved by 85.37 % and 22% over the FS and 3SS respectively for the OLS. For the DLS, the speed advantages are 88.77 % and 40% over the FS and 3SS. Furthermore, the accuracy of prediction of OLS and DLS are comparable to the 3SS.

CONTENTS

	Page
Abstract	I
Table of Contents	II
List of Figures	VI
List of Tables	X
Publications	XI
Acronyms	XIII
Acknowledgements	XVI
Chapter 1 introduction	
1.1 Background and Motivation.	1
1.2 Introduction to Digital Video Coding.	3
1.3 International Video Coding Standards.	5
1.4 Contribution of The Thesis.	8
1.5 Structure of The Thesis.	9
Chapter 2 Conceptual Background	
2.1 Introduction.	10
2.2 A Historical Overview of Video Coding.	12
2.3 Video Compression Characteristic.	15
2.3.1 Compression Ratio.	16
2.3.2 Image Quality.	16
2.3.3 Compression Speed.	18
2.4 Various Video Coding Techniques.	19
2.4.1 Spatial redundancy reduction.	20
2.4.2 Temporal redundancy reduction.	23

2.4.3 Entropy Coding. 24

2.5 MPEG Coding Standard. 25

2.5.1 MPEG Frame Structure. 26

2.5.2 MPEG Video Encoder and Decoder. 28

2.5.3 MPEG Data Stream. 30

2.5.4 Motion Estimation and Compensation. 32

2.5.5 MPEG Audio Encoder and Decoder. 34

2.5.6 MPEG Interleaved A/V Data Stream. 36

2.6 Summary 37

Chapter 3 Motion Compensation

3.1 Introduction. 39

3.2 Overview of Inter frame Coding Technique. 40

3.2.1 Sub-sampling. 40

3.2.2 Difference Coding. 41

3.2.3 Block Based Difference Coding. 42

3.2.4 Block Based Motion Compensation. 43

3.3 Motion Compensated Prediction. 45

3.3.1 Frame Segmentation. 46

3.3.2 Search Threshold. 47

3.3.3 Block Matching. 47

3.3.4 Motion Vector Correction. 47

3.3.5 Vector Coding. 48

3.3.6 Prediction Error Coding. 48

3.4 Summary 49

Chapter 4 Motion Estimation Techniques for Video Coding

4.1 Introduction. 50

4.2 The Fundamental Of Motion Estimation. 52

4.2.1 Preservation Constraint. 55

4.2.2 Coherence Constraint. 57

4.3 Estimation Methods. 58

4.4 Background Techniques for Motion Estimation. 59

4.4.1 Linear Regression. 59

4.4.2 Iterative Motion Estimation. 60

4.4.3 Pixel Recursive Algorithms. 62

4.5 The Block-Matching Algorithm (BMA). 63

4.5.1 BMA Principle. 63

4.5.2 Search Techniques. 65

4.5.3 Advance Possibilities. 65

4.6 Summary 65

Chapter 5 Search Algorithms in Motion Estimation

5.1 Introduction. 67

5.2 Full Search Block-Based Motion Estimation. 69

5.3 Fast Search Block-Based Motion Estimation. 71

5.3.1 Three Step Search Algorithm (3SS). 71

5.3.2 New Three Step Search Algorithm (N3SS). 72

5.3.3 One at a Time Algorithm (OTA). 76

5.4 Summary. 78

Chapter 6 Search Algorithms Design and Implementation

6.1 Introduction. 80

6.2 Simulation Environment for the Performance Evaluation. 81

6.2.1 Measurement Parameters. 81

6.2.2 Sources of Video Sequence. 81

6.3 The Matching Criteria. 83

6.4 Characteristic of a good search algorithm. 86

6.5 Simulation of well-known search algorithms. 87

6.5.1 FS algorithm. 89

6.5.2 3SS algorithm. 97

6.6 Summary. 105

Chapter 7 Novel Approaches and Algorithms Testing

7.1 Introduction. 106

7.2 Orthogonal Logarithmic Search (OLS) Algorithm. 108

7.2.1 Implementation of Orthogonal Logarithmic Search Algorithm. 111

7.2.2 Performance of the Orthogonal Logarithmic Search algorithm. 112

7.2.3 Comparative Performance of OLS. 119

7.3 Diagonal Logarithmic Search (DLS) algorithm. 123

7.3.1 Implementation of Diagonal Logarithmic Search Algorithm. 125

7.3.2 Performance of the Diagonal Logarithmic Search algorithm. 127

7.3.3 Comparative Performance of DLS. 133

7.4 Comparison between OLS and DLS algorithms 136

7.5 Summary. 143

Chapter 8 Conclusions and Future work

8.1 Conclusions. 146

8.2 Future Work. 148

References 150

LISTS OF FIGURES

		Page
Figure (2.1)	Block diagram of Video Compression	11
Figure (2.2)	Basic Predictive Coder	20
Figure (2.3)	Types of frames in the MPEG standard	27
Figure (2.4)	The block diagram of the MPEG encoder.	29
Figure (2.5)	The block diagram of the MPEG decoder	29
Figure (2.6)	MPEG Data Stream	30
Figure (2.7)	Motion compensated interpolated implemented in MPEG	33
Figure (2.8)	Block Diagram of MPEG Audio Encoder and Decoder	35
Figure (3.1)	Consecutive frames from the "Claire" Video sequence	41
Figure (3.2)	Typical Motion Compensation Prediction Codec	45
Figure (3.3)	Motion Compensation Scheme	46
Figure (4.1)	The Typical Motion Estimation Processes	51
Figure (4.2)	Aperture Problem	53
Figure (4.3)	The Bicycle Wheel: ambiguity in the correspondence problem because aliasing	54
Figure (4.4)	The optical flow is not always equal to the motion Field.	54
Figure (4.5)	Limit of iterative motion determination:	61
Figure (4.6)	The example of Motion Vector Predication	64
Figure (5.1)	Block in the Current Frame and Search Area in the Previous Frame	70
Figure (5.2)	Search Pattern of Three Step Search	72
Figure (5.3)	Search Pattern of New Three Step Search	73
Figure (5.4)	Comparative PSNR Performance between 3SS and N3SS using "Claire"	74
Figure (5.5)	Comparative PSNR Performance between 3SS and N3SS	

	using "Carphone"	74
Figure (5.6)	Comparative PSNR Performance between 3SS and N3SS using "Foreman"	75
Figure (5.7)	The Operation Time of 3SS and N3SS	76
Figure (5.8)	Search Pattern of One at a Time Algorithm	77
Figure (5.9)	The Subjective Performance of One at a Time Algorithm Using "Carphone"	78
Figure (6.1)	The Comparison of Matching Criteria	85
Figure (6.2)	Motion Compensation Process	89
Figure (6.3)	The Sample of Motion Estimation using FS Algorithm	90
Figure (6.4)	"Claire" Subjective quality of the predicted frame using FS	91
Figure (6.5)	"Foreman" Subjective quality of the predicted frame using FS	92
Figure (6.6)	"Carphone" Subjective quality of the predicted frame using FS	93
Figure (6.7)	The PSNR Performance of "Claire" over 90 frames using FS	94
Figure (6.8)	The PSNR Performance of "Carphone" over 90 frames using FS	95
Figure (6.9)	The PSNR Performance of "Foreman" over 90 frames using FS	95
Figure (6.10)	The Sample of Motion Estimation using 3SSAlgorithm	98
Figure (6.11)	"Claire" Subjective quality of the predicted frame using 3SS	100
Figure (6.12)	"Foreman" Subjective quality of the predicted frame using 3SS	101
Figure (6.13)	"Carphone" Subjective quality of the predicted frame using 3SS	102
Figure (6.14)	The PSNR Performance of "Claire" over 90 frames using 3SS	103
Figure (6.15)	The PSNR Performance of "Carphone" over 90 frames using 3SS	103
Figure (6.16)	The PSNR Performance of "Foreman" over 90 frames using 3SS	104
Figure (7.1)	Search Pattern of the Orthogonal Logarithmic Search Algorithm	110
Figure (7.2)	"Claire" Subjective quality of the predicted frame using OLS	114
Figure (7.3)	"Foreman" Subjective quality of the predicted frame using OLS	115
Figure (7.4)	"Carphone" Subjective quality of the predicted frame using OLS	116

Figure (7.5) The PSNR Performance of "Claire" over 90 frames using OLS 117

Figure (7.6) The PSNR Performance of "Foreman" over 90 frames using OLS 117

Figure (7.7) The PSNR Performance of "Carphone" over 90 frames using OLS 118

Figure (7.8) Comparative PSNR Performance of "Claire" Using FS, 3SS and OLS 120

Figure (7.9) Comparative PSNR Performance of "Carphone" Using FS, 3SS and OLS 120

Figure (7.10) Comparative PSNR Performance of "Foreman" Using FS, 3SS and OLS 121

Figure (7.11) Comparative Speed of Operation using FS, 3SS and OLS 122

Figure (7.12) The Search Pattern of the Diagonal Search Algorithm 124

Figure (7.13) "Claire" Subjective quality of the predicted frame using DLS 128

Figure (7.14) "Foreman" Subjective quality of the predicted frame using DLS 129

Figure (7.15) "Carphone" Subjective quality of the predicted frame using DLS 130

Figure (7.16) The PSNR Performance of "Claire" over 90 frames using DLS 131

Figure (7.17) The PSNR Performance of "Forman" over 90 frames using DLS 132

Figure (7.18) The PSNR Performance of "Carphone" over 90 frames using DLS 132

Figure (7.19) Comparative PSNR Performance of "Claire" Using FS, 3SS and DLS 133

Figure (7.20) Comparative PSNR Performance of "Carphone" Using FS, 3SS and DLS 134

Figure (7.21) Comparative PSNR Performance of "Foreman" Using FS, 3SS and DLS 135

Figure (7.22) Comparative Speed of Operation using FS, 3SS and DLS 136

Figure (7.23) "Claire" Subjective quality comparison between OLS and DLS 138

Figure (7.24) "Foreman" Subjective quality comparison between OLS and DLS 139

Figure (7.25) "Carphone" Subjective quality comparison between OLS and DLS 140

Figure (7.26) Comparative Speed of Operation using OLS and DLS 142

LIST OF TABLES

		Page
Table 1. 1	Video Coding Standards Developed by Different Organisations	7
Table 2.1	Parameters of MPEG Algorithms	26
Table 2.2	Layers of MPEG Video Stream Syntax	36
Table 5.1	Average PSNR of N3SS and 3SS	75
Table 6.1	Characteristic of the Benchmarks Video Sequence	82
Table 6.2	Summarisation of the condition of simulation	87
Table 7.1	Comparison of the average PSNR using OLS and DLS	141
Table 7.2	Summary of the average PSNR using FS, 3SS. OLS and DLS	143
Table 7.3	Comparison of the Processing Time	144

PUBLICATIONS

[1] S. Soongsathitanon, S.S.Dlay, 'A New Orthogonal Logarithmic Search Algorithm For Fixed Block-Based Motion Estimation for Video Coding', *Proceeding of International Symposium on Communication Systems, Networks and Digital Signal Processing (CSNDSP2002)* Co-Sponsored by: *IEEE, IEE, EURASIP, BCS*, July 15-17, 2002, Staffordshire University, UK, pp. 256-259.

[2] S. Soongsathitanon, S.S.Dlay, 'Video Coding For Multimedia Communication Using A Novel Orthogonal Logarithmic Search Algorithm', *Proceeding of IASTED International Conference on Communications and Computer Networks (CCN2002)*, November 4-6, 2002, Massachusetts Institute of Technology (MIT), Cambridge, USA, pp. 498-502

[3] S. Soongsathitanon, S.S.Dlay, 'A Novel Orthogonal Logarithmic Search Algorithm for Low Bit Rate Video Coding', *Proceeding of SPIE/IS&T International Symposium on Image and Video Communications and Processing 2003*, SPIE Vol. 5022/Part2, January 21-24, 2003, Santa Clara, California, USA, pp. 958-965

[4] S. Soongsathitanon, S.S.Dlay, 'A New Motion Estimation for Multimedia Video Compression', *Proceeding of Postgraduate Research Conference in Electronics, Photonics, Communications and Software (PREP2003)* Sponsored by: *IEEE, ESPRC, IOP* , April 14-16, 2003, Exeter University, UK, pp. 208-209

[5] S. Soongsathitanon, S.S.Dlay, 'A Fast Novel Motion Estimation For Video Coding Using Logarithmic Diagonal Search Algorithm', *Proceeding of* 7th *WSEAS*

International Multiconference on Circuits, Systems, Communications and Computers(CSCC2003). July 7-10, 2003, Corfu Island, Greece

[6] S. Soongsathitanon, S.S.Dlay, ' Block-Based Motion Estimation Using A Novel Orthogonal Logarithmic Search Algorithm', *Proceeding of IEE International Conference on Visual Information Engineering (VIE2003).* Conference Publication 495, July 7-9, 2003, University of Surrey, UK, pp. 246-249

[7] S. Soongsathitanon, S.S.Dlay, 'A Novel Fast Search Algorithm For MPEG Standard', *Proceeding of International Symposium on Communication Systems, Networks and Digital Signal Processing (CSNDSP2004)* Co-Sponsored by: *IEEE, IEE, EURASIP, BCS,* July 20-22, 2002, University of Newcastle upon Tyne, UK

[8] S. Soongsathitanon, S.S.Dlay, 'A Fast Novel Motion Estimation For Video Coding Using Logarithmic Diagonal Search Algorithm', Journal: *Recent Advances in Intelligent Systems and Signal Processing.* Publisher World Scientific And Engineering Academy and Society Press, ISBN 960-8052-87-4 ,pp 198-203

ACRONYMS

2D	2-Dimensional
3D	3-Dimensional
3SS	Three Step Search
A/V	Audio/Video
ADPCM	Adaptive Differential Pulse Code Modulation
ATM	Asynchronous Transfer Mode
BMA	Block Matching Algorithm
BMP	Bit-Mapped Graphic Format
CCIR	International Radio Consultative Committee
CCITT	International Telegraph and Telephone Consultative Committee
CD	Compact Disc
CD-DA	Compact Disc Digital Audio
CIF	Common Intermediate Format
DAT	Digital Audio Tape
DCT	Discrete Cosine Transform
DFD	Displaced Frame Difference
DLS	Diagonal Logarithmic Search Algorithm
DPCM	Differential Pulse Code Modulation
DVD	Digital Versatile Disc
EDTV	Enhanced Definition Television
FLC	Fixed Length Coding
FR	Full-reference models
FS	Full Search
GIF	Graphics Interchange Format

GOP	Group-Of-Pictures
HDTV	High Definition Television
HVS	Human Visual System
IDCT	Inverse Discrete Cosine Transform
ISDN	Integrated Services Digital Network
ISO/IEC	International Organisation for Standardisation-International Electrotechnical Commission
ITU-T	International Telecommunications Union - Telecommunication Standardisation Bureau
JPEG	Joint Photographic Experts Group
LZW	Lempel-Zif-Welch
MAD	Mean Absolute Difference
MC	Motion Compensation
ME	Motion Estimation
MPEG	Motion Picture Experts Group
MSE	Mean-Square error
N3SS	New Three Step Search
NCF	Normalised cross-correlation function
NR	Non-reference models
NTD	Number of Threshold Differences
NTSC	National Television System Committee
OLS	Orthogonal Logarithmic Search
OTS	One at a Time Algorithm
PAL	Phase Alternating Line Television System
PCM	Pulse Code Modulation
PESs	Packetised Elementary Streams

PKZIP	Phillip Katz ZIP
PSNR	Peak-Signal-to-Noise-Ratio
PSTNs	Public Switched Telephone Networks
QCIF	Quarter Common Intermediate Format
RGB	Read- Green-Blue
RR	Reduced-reference models
SIF	Source Input Format
S-QCIF	sub-QCIF
TIFF	Tag Image File Format
TS	Transport Stream
VGA	Video Graphics Array
VHS	Video Home System
VLC	Variable Length Coding
VQ	Vector Quantisation
YC_bC_r	Luminance/Chrominance Signals (CCIR-601)
YIQ	Luminance/Chrominance Signals (NTSC System)
YUV	Luminance/Chrominance Signals (PAL System)

ACKNOWLEDGEMENTS

I would like to acknowledge the persons that either directly or indirectly have supported me throughout this research. I would like to express my earnest appreciation to my supervisor Dr. Satnam S. Dlay for his invaluable guidance, time, patience and encouragement throughout this research work. His profound knowledge and experience have guided me through stages of development, of both research and myself as a researcher. He gave me many chances to participate in the international conferences. I would like to thank Thepsatri Rajabhat University and Royal Thai Government who sponsored me for this study and gave me an opportunity to gain plenty of valuable knowledge. Finally, I would like to thank all of my colleagues in Multimedia and Computer Laboratory who always supported me either direct or indirect way.

Chapter 1

INTRODUCTION

1.1 Background and Motivation

In recent years, Video compression has played a vital role in data storage and transmission as it has been the main focus in many applications such as multimedia communications, remote monitoring, videophones, videoconference etc. It has become an interesting and very important area of research. The advent of new technology leads us to the new communication system which involves visual information. Because of the huge amount of information to be stored and transmitted, it needs to be compressed. Without a compression scheme, typical scenes with a resolution of 176×144 pixels at 10 frames per second and with the luminance component quantised using the customary 8-bits/s, that is used in Quarter Common Intermediate Format (QCIF), video sequences would occupy approximately 2.03 Mbits/s. This generates huge amounts of data which would quickly overwhelm all available bandwidth and disk space if it were not compressed. To alleviate this problem, the compression scheme will be implemented by choosing an encoding scheme that can remove redundancy while keeping the important information. At present mobile service provide not only voice communication, but also multimedia communication. The NOKIA mobile system can send and receive 176×144 video pixels at 10 frames per second over the bandwidth 16-26 kbits /s [1].

The study in this thesis is motivated by the possibility of new video data compression schemes of moving images based on motion compensation scheme with faster search algorithms. The advantage of these proposed fast search algorithms is that they can achieve high speeds of operation, acceptable image quality and can lead to an efficient implementation. The rate distortion performance of the scheme could be comparable to the current international standard MPEG (Motion Pictures Experts Group). However, the search for improved techniques for video data compression remains an active area of research. The fundamental goal of a video compression scheme is to reduce the information data while maintaining an acceptable fidelity or image quality. The typical video compression system is composed of two principle parts which are the coder and decoder. The coder obtains the sequence of video frames with fixed resolution as input, and produces codes representing those video frames. Meanwhile, the decoder retrieves the codes back to the original input as decoded outputs or reconstructed frame which are not necessarily identical to the original. Codes are either transmitted over transmission channels or stored on storage devices.

There are two categories of data compression. The first category is *lossless* or reversible compression in which the original data can be completely recovered from the compressed form. It is usually limited to compression ratios of 4:1 or less. The second category of data compression is *lossy* or irreversible technique. The lossy technique provides higher compression ratios than lossless compression, usually between 50 ~ 150 times, but it introduces error or degradation into the data so that the original information can not be perfectly generated. The performance of any coder is determined by the bit rate achieved, which presents the compression ratio and the distortion or degradation between originals and decoded images introduced by the coders. Fortunately, most images contain some amount of redundancy that can sometime be removed when the image is stored or transmitted and replaced when it is reconstructed and this can be exploited in order to reduce the amount of data necessary for their representation. The human eye is insensitive to a wide variety of information. That is, an image can be changed in many ways and can not be detected by humans.

1.2 Introduction to Digital Video Coding

A video sequence fundamentally consists of a sequence of still pictures or frames of a scene taken at various subsequent intervals in time. Each frame represents the distribution of light energy and wavelength over a finite size area and is expected to be seen by a human viewer [2]. In an analogue system for example a video camera produces an analogue signal of an image scanned from left to right and from top to bottom making up a frame [3]. The choice of number of scanned lines per picture is a trade-off between the bandwidth, flicker and resolution. The analogue signal is then converted to digital signal by three basic operations of low pass filtering, sampling and quantisation. A colour signal can be represented in RGB (red, green, blue) colour component format or any of several luminance-chrominance component formats. These include YUV, YC_bC_r (a colour coordinate system closely related to YUV), and YIQ (another luminance-chrominance format preferred for NTSC-standard video) [3]. The typical main component of YUV colour signal consists of a luminance signal, Y representing brightness and two chrominance signal U and V representing colour. This thesis is exclusively concerned with the luminance Y component of the video sequence.

The digital video coding aims to accommodate video within the allocated bandwidth by mean of compression. To achieve this purpose, the two types of redundancy in digital image are taken into accounts which are:

- Spatial redundancy: This redundancy is related to the correlation between neighbouring pixels within a frame.
- Temporal redundancy: The correlation between consecutive frames of a video sequence. Same objects appearing in the previous frame are likely to appear in the current frame.

There are a number of techniques used for exploiting spatial redundancy such as line-by-line coding technique. The samples of line-by-line coding techniques are different pulse code modulation and adaptive pulse code modulation. These techniques [4] exploit the one-dimensional spatial redundancy between pixels in a given line scan, but ignore the potential compression of vertical adjacent pixel correlation. A more efficient coding scheme is the so-called block based methods that encode a group of pixels at a time. These blocks are typically 4×4 or 8×8 pixels in size and are treated as two dimensional vectors which facilitate much higher compression ratios. This compression potential is often utilised by a technique more commonly referred to as vector quantisation (VQ). The extension to block based coding is transform coding [4], where the block is transformed into the spatial domain. The most practical transform coding is the discrete cosine transform (DCT), which provides estimates of the frequency contents of each image region (e.g. each block of 8×8, pixels, in the DCT case). Given that most of the energy is concentrated in a few coefficients, only a few bits will be needed to provide a good approximation to the original image block.

As already mentioned, the temporal redundancy is also exploited due to the similarity of consecutive frames. The consecutive frames of video sequence naturally exhibit similarity, especially in video scenes with low motion. The technique which is widely used for exploiting temporal redundancy is motion compensation (MC) prediction [5-17]. The well-known technique to obtain motion compensation prediction is motion estimation. Motion estimation is a technique that breaks video frames into non-overlapping blocks and then searches the previous frame for blocks that provide the most similarity to each block belonging to an object in the current frame from a block in the same object in the previous frame.

1.3 International Video Coding Standards

Standards are essential for communication. Standards define a common language that different parties can use, so that they can communicate with one another. Video coding standards define the bit stream syntax, which is the language that encoder and decoder use to communicate. Moreover defining the bit stream syntax, standards for video coding are also required to be efficient for the compression of video content. There are two major organisations for the development of standards, the ITU-T (International Telecommunications Union -Telecommunication Standardisation Bureau) and the ISO/IEC (International Organisation for Standardisation-International Electrotechnical Commission). The ISO/IEC has formed the working group, the Moving Picture Experts Group (MPEG) [18], in an attempt to introduce a degree of standardisation. The MPEG group of ISO/IEC has established a series of recognised image codecs. The MPEG1 [18] is designed for such products as Video CD and MP3. Meanwhile the MPEG-2[19] is the standard on which such products as Digital Television set top boxes and DVD are based. The MPEG-4[19] is the standard for multimedia for the fixed and mobile web, while MPEG-7[19] is the standard for description and search of audio and visual content. The newest of a series of MPEG standards is MPEG-21[19]. MPEG-21 started in June 2000 and is still under development. MPEG-21 is an open standards-based framework for multimedia delivery and consumption. The aim of MPEG-21 is to define a multimedia network framework to enable transparent and augmented use of multimedia resources across a wide range of networks and devices. The main MPEG-21 parts have already reached the status of international standards in 2003. A Technical Report and two standards have been produced and three more parts of the standard are at different stages of development. Several Calls for Proposals have already been issued.

MPEG-4 [19] is the most up-to-dated standard targeted for mobile multimedia communication including digital video coding. Nevertheless, some parts of the MPEG-4 standard such as the inter frame coding can be further developed. Having implemented the new algorithms for digital video coding, MPEG-4 is taken into account. The MPEG-4 standard was first introduced in 1994 and originally created as a standard for very low bit rate coding of limited complexity audiovisual material. The scope was later extended to supporting new functions. The main application of MPEG-4 standards is in relation to the audio and video associated with interactive multimedia application over the Internet and various types of entertainment networks. The prime difference between MPEG-4 and the other standards is that the MPEG-4 is focused on "content-based" video functionality, which attempts to separate an image into objects. The bit stream of a frame/scene encoded in MPEG-4 is transmitted over a network in the form of transport stream (TS), consisting of a multiplexed stream of packetised elementary streams (PESs) [20].

In addition, the international telecommunications union (ITU-T) has also introduced a series of recommendations for audio-visual standards. The H.261[21] was introduced in 1993 and is a video codec for audiovisual services at p x 64 kbits/s (p is in the range 1-30). The H.262 and H.263 were introduced later. The H.261 video compression standard is defined for the video telephony and video conferencing services over an integrated services digital network (ISDN). The digitisation format used is either the common intermediate format (CIF) or QCIF. Normally, the CIF is used for video conferencing and the QCIF for video telephony. In 1998 the H.263 [22] was initially recommended for video applications over wireless and public switched telephone networks (PSTNs) in which the bit rate is lower than 28.8 kbits/s. The applications include video telephony, video conferencing, security surveillance, interactive games playing, etc. all of which require the output of video encoder to be transmitted across the network connection in real time as it is output by encoder. The basic structure of H.263 is based on that used in the H.261 except for several modifications. Motion accuracy is increased through the use of half pixel accuracy in stead of full pixel that is used in H.261.

Organisation	Video Coding Standard	Typical Range of Bit Rates	Typical Application
ITU-T	H.261	p×30 kbits/s. p=1....30	ISDN Videophone
ITU-T	H.263	< 64 kbits/s.	PSTN Videophone
ISO	MPEG-1	1.5 Mbits/s	CD-ROM
ISO	MPEG-2	5-20 Mbits/s	HDTV
ISO	MPEG-4	10 kbits/s-10Mbits/s	Multimedia Applications
ISO	MPEG-7	-	Multimedia Content Description Interface
ISO	MPEG-21	-	Multimedia Framework

Table 1. 1 Video Coding Standards Developed by Different Organisations

The formats of video that are mandatory for H.263 are the QCIF and the sub-QCIF(S-QCIF). The summarisation of the standards of both organisations is shown in the Table 1.1. The major differences between these standards lie in the operating bit rates and the application they are targeted for.

1.4 Contribution of The Thesis

The aim of this thesis is to develop and implement new algorithms for motion estimation and prediction. This research has been surveyed various techniques which are widely used in well-known video standards. The extensively used techniques have been investigated and many of these important techniques are simulated and tested. However the core contribution is mainly on the motion estimation which is a vital part of motion compensation scheme. The comparative performance analysis of several search techniques is presented. The performance of image sequence is concentrated on the 176×144 pixel QCIF videophone sequences which is the application of the mobile phone technology. The simulation bench mark video sequences used are Foreman, Carphone and Claire. The well-know search and up-to-date techniques are investigated and the new techniques are proposed. These novel techniques are intended to fulfil the real-time communication because of their low complexity of the algorithms and the quality of the predicted image. The new methods reduce the candidate blocks for searching motion vectors. These novel techniques are compared with the well-known techniques, full-search and three-step-search methods which are adopted by the world wide standards. Moreover the main factors of the search techniques which have the most impact on the performance of the algorithms are also tested. The best matching criteria are most suitable for the search algorithms are proclaimed.

1.5 Structure of The Thesis

This thesis discusses the recent developments in search algorithms with respect to its application in low bit rate video coding.

The structure of this thesis is as follows.

In Chapter 2 the background of video compression and overview of video compression are presented. The characteristics of the video coding are discussed and various video coding methods are introduced. Well-known techniques such as entropy coding, transform coding, motion compensation are included in this chapter. Moreover the MPEG standards are also presented and. the structure and the methodologies used in MPEG standard are explained.

In Chapter 3 a brief overview of inter frame coding is presented and the motion compensation scheme is explained in detail.

In Chapter 4 a survey of motion estimation techniques is given and the problems associated with motion estimation are explained. Also in particular the main approaches of the motion estimation are introduced.

Chapter 5 introduces the conventional full search technique. The sub-optimal search techniques such as three-step-search algorithm are discussed in depth.

In chapter 6 the implementation of the well-known techniques, Full Search Algorthm and Three Step Search, are introduced. The results of the simulation are shown. These simulations lead to the new proposed techniques.

In Chapter 7 the implementation of the novel search algorithms are given and the simulation results are presented. The algorithms are tested and comparisons made with the International Standard, the three step search (3SS), and the full search methods.

Chapter 8 is presents the conclusions and provides a summary of the new motion estimation and prediction algorithms. Finally further work is suggested in this chapter.

Chapter 2

CONCEPTUAL BAGROUND

2.1 Introduction

This chapter provides a brief introduction to the background of video data compression, including a historical overview of video compression and the main video coding characteristics. In addition, various video coding techniques are included. The main techniques for the redundancy reduction such as predictive coding, vector quantisation and transform coding are explained. Finally an overview of the current standard MPEG algorithm is given and a brief explanation of the MPEG structure is described. The parameters which are used in MPEG standard are shown. The motion estimation and compensation explained. Most digital videos contain a high degree of redundancy. The redundancy is best described as unnecessary data carried by a video signal. Since this data is unnecessary, removing it will reduce the bit rate without necessarily affecting the picture. The video compression can be achieved by reducing the redundancy present in a video signal. An efficient compression technique can successfully reduce the redundancies of a video sequence and keep only the necessary amount of information needed to store or transmit them. In addition the pshychovisual redundancy can also be exploited because of the nature of Human Visual System (HVS). The human visual system do not perceive certain picture in details. These picture details can be altered (i.e. reducing the number of bits per sample) or be removed, thus reducing the data rate and will result in imperceptible errors in the reconstructed picture. Digital video compression has applications in many areas such as Videoconferencing, High

Definition TV (HDTV), network communication, video telephony, interactive television, electronic commerce and Web TV .

The basic video compression scheme is shown in Figure (2.1). On the encoder side, the video sequence feeds in a form of analogue signal, and it is converted to digital form, then the pre-processing step is applied which could be sampling or transformation to another domain. Finally, entropy coding such as Huffman code or arithmetic coding is used for transmitting over the communication channel. On the receiver side or the decoder the process is reversed. A digital video frame is a two-dimensional array of pixels each with a value that corresponds to the intensity of video frame. Although the capability of modern storage media and bandwidth of transmissions systems are high, the performance is still unsatisfactory. The video data is dramatically becoming so large that they can not be adequately compressed for transmission or archival with current techniques. In this respect, intensive research activity is still taking place to address this issue.

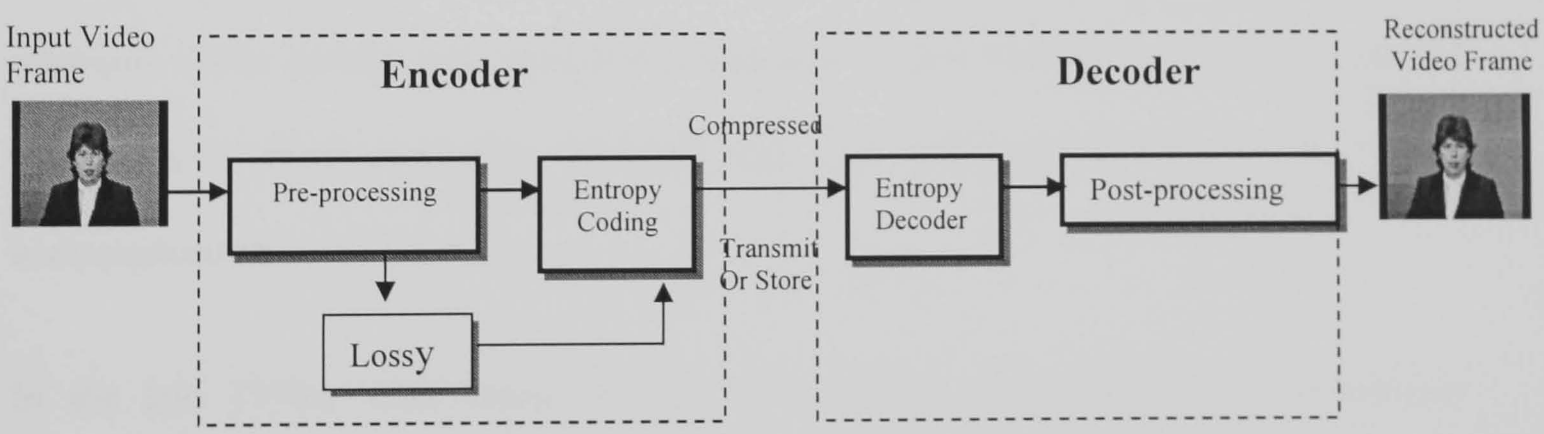


Figure (2.1) Block diagram of Video

2.2 A Historical Overview of Video Coding

The video simply consists of a sequence of frames or pictures. So the principle of video compression is fundamentally based on image compression. Image compression has been studied for more than 40 year and can probably be tracked back to 1838 when the Morse code was introduced for telegraphy. Morse code is an early example of data compression based on using shorter codewords for letters such as "e" and "t" that are more common in English. Nevertheless the modern work on data compression seemingly began in the late 1940s with the development of information theory. In 1946 Gabor introduced the foundation of variable time (space) frequency resolution analysis. Two years later, C.E. Shannon [3] provided the theoretical basis for efficient coding in general by devising a systematic way to assign codewords based on probabilities of blocks. Later an optimal method for doing this was found by David Huffman [3] in 1951. In 1952 Cutler [4] introduced originally the application for the patent on predictive coding and this was followed by Harrison who published this work on the application of linear prediction to television. Later in 1959 C.E. Shannon considered the application of a fidelity criterion to these earlier results on coding. In the 1960s an analogue video system was used but it required a wide bandwidth and the produced postcard-size black-and-white pictures did not add appreciably to the voice communication.

In the late 1970s, with online storage of text files becoming common, software compression programs began to develope, almost all based on adaptive Huffman coding. In 1971 Ericsson demonstrated the first trans-Atlantic video telephone call. In 1977 Abraham Lempel and Jacob Ziv suggested the basic idea of pointer-based encoding. In the mid-1980s, following work by Terry Welch, the so-called LZW algorithm rapidly became the method of choice for most general-purpose compression systems. It was used in programs such as PKZIP, as well as in hardware devices such as modems. In 1982, CCITT (forerunner of the ITU-T) issued the H.120 video coding standard under the European COST211 project which achieved a target rate of 2 Mbits/s for Europe and 1.544 Mbits/s. COST211 video codec was achieved by the method based on Differential Pulse Code Modulation (DPCM). The nature work of DPCM is based on pixel-by-pixels basis. The image quality of this codec gave a very good spatial resolution; however it had a very poor temporal quality. To improve the COST211, the new design of so-called block based codec was introduced. During the late 1980s, the 15 block based videoconferencing proposals were submitted to the ITU-T, 14 of them were based on the Discrete Cosine Transform (DCT) and only one on Vector Quantisation (VQ). However, the subjective quality of video sequences presented to the panel showed hardly any significant differences between the two coding techniques. In parallel to ITU-T's investigation during 1984-88, the Joint Photographic Experts Group (JPEG) was also interested in compression of static images. The JPEG mainly based on DCT for the unit of compression due to the possibility of progressive image transmission. JPEG's decision undoubtedly influenced the ITU-T in favouring DCT over VQ. By now there was a worldwide activity in implementing the DCT in chips and on DSPs.

In the early 1990s, lossy compression methods also began to be widely used. Current image compression standards include: FAX CCITT 3 (run-length encoding, with codewords determined by Huffman coding from a definite distribution of run lengths); GIF (LZW); JPEG (lossy discrete cosine transform, then Huffman or arithmetic coding); BMP (run-length encoding); TIFF (FAX, JPEG, GIF). In 1992 CCITT recommended the H.261 video coding and H.320 for ISDN conferencing. Also during the 1990s, the Motion Picture Experts Group (MPEG) started investigating coding techniques for storage of video, such as CD-ROMs. The aim was to develop a video codec capable of compressing highly active video such as movies onto the hard disks, with a performance comparable to that of VHS quality. In fact, the basic framework of the H.261 generated MPEG, called the MPEG-1 standard, which was capable of accomplishing this task at 1.5 Mbits/s. Since for video storage encoding and decoding delays are not a major constraint, one can trade delay for compression efficiency. For example in the temporal domain a DCT might be used instead of DPCM but with much improved motion estimation, such that the motion compensation removes temporal correlation. This later option was adopted with MPEG-1

These days, MPEG-1 decoders/players are becoming commonplace for multimedia on computers. MPEG-1 decoder plug-in hardware boards (e.g. MPEG magic cards) have been around for a while, and now software MPEG-1 decoders are available with the release of operating systems or multimedia extensions for PC and Mac platforms. Since in all standard video codecs, only the decoders have to comply with proper syntax, software based encoding has added the extra flexibility that might even improve the performance of MPEG-1 in the future.

MPEG-1 was originally optimised for typical applications using non-interlaced video of 25 frames/s, in European format and 29.9 frames/s in North American format. Early versions of MPEG-1 for interlaced video, such as those used in broadcast, were called MPEG1+. A new generation of MPEG, called MPEG-2 was soon adopted by broadcasters. MPEG-2 codes interlaced video at bit rates 4-9 Mbits/s, and is now well on its way to making a significant impact in a range of applications such as digital terrestrial broadcasting, digital satellite TV, digital cable TV, digital versatile disc (DVD) and many others. Since the late 90s television broadcasters have started using MPEG-2 coded digital forms.

A slightly improved version of MPEG-2, called MPEG-3, was to be used for coding of High Definition (HD) TV, but since MPEG-2 could itself achieve this, MPEG-3 standards were folded into MPEG-2.

2.3 Video Compression Characteristic

Unlike normal analogue video used by everyday televisions, the digital video basically consists of a sequence of non-interlaced images, each of which is a two-dimensional frame of picture elements or ***pixels.*** The characteristic of video can be defined as the following variables:

Frame rate: The number of frames displayed per second. The illusion of motion can be experienced at frame rates as low as 12 frames per second, but modern cinema uses 24 frames per second, and PAL television 25 frames per second.

Frame dimensions: The width and height of the image expressed in the number of pixels. Digital video comparable to television requires dimensions of around 640 x 480 pixels.

Pixel depth: The number of bits per pixel. In some cases it might be possible to separate the bits dedicated to luminance from those used for chrominance. In others all the bits might be used to reference one of a range of colours from a known palette.

There are three main characteristics for the measurement to assess the performance of video compression algorithms. These are compression ratio, image quality and compression speed. The compression ratio is the measurement of the capability of the storage or data reduction. The higher compression ratio means the better data reduction can be achieved. Secondly the image quality is a core measurement which aims to compare the decompressed data to the original data. The compression speed or cost that refers to the computational effort required to encoding and decoding processes. These characteristic functions are usually used for judging the performance of the compression technique. The use of these characteristic measurements depends on the application and use of images for particular requirements. In addition these characteristics are used to determine the suitability of the compression techniques to different applications according to acceptable reconstructions and range of compression ratios available and implementation costs. The following sections discuss each of these attributes in more details.

2.3.1 Compression Ratio

Compression Ratio is a measure of the closeness between the original image and compressed image. The compression ratio can be found from a simple formula which is the size of the original divided by the size of the compressed image. This ratio shows the capable of the data compression algorithms how much compression is achieved.

$$Compression\ Ratio = \frac{Size of Original image}{Size of Compressed image} \tag{2.1}$$

The compression ratio can be used for indicating the picture quality, since most of the compression techniques operate over a range of compression rate and decompression quality. Generally, the higher the compression ratio, the poorer the quality of resulting images. The trade–off between compression ratio and picture quality is an important metric to consider when compressing images.

2.3.2 Image Quality

Image quality is one of the significant measures for the image and video compression system. Normally, the compression and decompression process cause the degradation of the reconstructed image. So the image fidelity can be used to assess the degree of degradation. The image quality can be grouped into two quality measures, subjective image quality and objective image quality. Subjective image quality is determined by statistically processing the fidelity rating given by a group of human viewers, while objective image quality is defined by a computational process that does not require human intervention. For the subjective quality assessment, the quality is rated using a discrete or a continuous scale ranging from bad quality to excellent quality. So the subjective quality assessment is very tedious, expensive and cannot be conducted in real time. The subjective quality requires many considerations, standard viewing conditions, criteria for observer and scene selection, assessment procedures, and analysis methods. Many observers are needed and the assessments are lengthy, the procedure is therefore very costly. Moreover, it is very difficult to embed it into a practical video processing system because it cannot be implemented automatically. For this reason the objective quality assessment are used more extensively. The objective assessment can compute the image quality automatically and in a relatively short period of time. This is very important for real world applications.

There are a number of models for objective perceptual video quality and assessments have been introduced over the years. However, the objective methods for image quality estimation can be divided into three types. Full-reference models (FR) where the difference between the original and the distorted sequence is computed. Reduced-reference models (RR) which computes statistics on the distorted sequence and compares them with corresponding stored statistics from the original sequence. Non-reference models (NR) do not use the original sequence at all.

The simplest measures of quality are the Mean-Square error (MSE), the Peak-Signal-to-Noise-Ratio (PSNR). The MSE between two images with a size $M * N$ where X_{ij} is the original image and $\hat{X}_{ij}$ is the reconstructed image are:

$$MSE=\frac{1}{M*N}\sum_{i=1}^{M}\sum_{j=1}^{N}(x_{ij}-\hat{x}_{ij})^2 \tag{2.2}$$

One problem with MSE is that it depends strongly on the image intensity scaling. However, PSNR avoids this problem by scaling the MSE according to the image. It is determined as follows:

$$PSNR = 10\log_{10}\left(\frac{S^2}{MSE}\right) \tag{2.3}$$

Where, S is the maximum intensity value. PSNR is measured in decibels (dB). This measure (PSNR) is also not ideal, but it is commonly use. Its main failure is that the signal strength of the image is estimated as $(S)^2$ (Value square), rather than the actual signal strength of the image.

2.3.3 Compression Speed

Compression and decompression times are defined as the amount of time required for compressing and decompressing a picture or one image frame. These values depend on the following considerations:

1. The complexity of the compression algorithm, where a complex compression technique can produce better quality images, but it could be time consuming which it is not suitable for some real time applications.
2. The efficiency of the software or hardware implementation of the algorithm.
3. The speed of the utilised processor or auxiliary hardware.

Generally, the faster the compression/decompression can be performed, the better. Fast compression time increases the speed with which resulting compressed image can be created. Fast decompression times increase the speed with which the user can display and interact with the reconstructed images.

Speed of compression usually matters much more if the data is to be transmitted rather than stored. The decompression speed is important for storage and retrieval and is vital for reception of transmitted data. Some compression techniques show symmetry for compression and decompression speeds.

2.4 Various Video Coding Techniques

Since 1989 the digital video coding techniques have been widely developed and implemented. Normally the characteristic of these techniques such as compression ratio vary according to the subjective acceptable level of error and the definition of the word compression. The coding techniques can be classified as "lossless" and "lossy" approaches. The lossless approach aims to reduce the amount of data or image for storage or transmission where the reconstructed image quality is required to be identical with the original image. The typical compression ratio of the lossless compression techniques are about 2 to 3. In contrast the lossy coding technique aims to achieve the compression ratio around 50 ~ 150 times, where the good quality of images achieved. This technique of compression allows the image quality degradation and introduces errors into the data, so that the original can not be recovered perfectly. These video coding techniques aim to reduce three fundamental redundancies which are spatial redundancy, temporal redundancy and entropy coding. The coding techniques which reduce the spatial correlation are referred to as intra frame coding, whereas those coding techniques which reduce the temporal correlation are called the inter frame coding. Spatial redundancy reduction can be divided into 2 groups which are predictive coding and transform coding. The details of these techniques are explained in more details in the following sections.

2.4.1 Spatial redundancy reduction

There are a large number of methods to tackle the spatial redundancy. These techniques can be classified as predictive coding, vector quantisation and transform coding.

Predictive Coding

In a pixel, there is a strong spatial correlation between adjacent pixels. To reduce the data information, the predictive coding can be used. The predictive coding aims to removed the spatial redundancy between the successive pixels and encode only the new information. The predictive coding uses pixels or data already received to form an estimate or 'prediction' of the next pixels or data to be transmitted. The predictive coder consists of 3 main components which are predictor, quantiser and code assigner. A basic form of predictive coding is illustrated in figure (2.2).

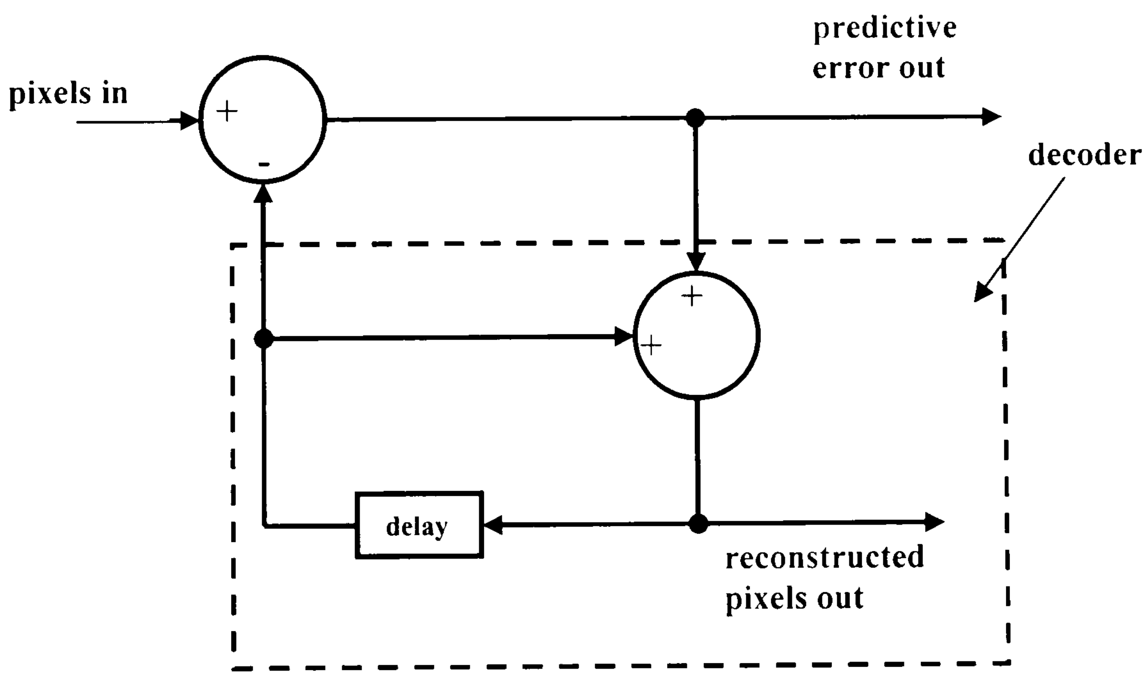


Figure (2.2) Basic Predictive Coder

Pixels are fed into the system in raster scanned order (i.e. pixel by pixel scanned left to right across the image, line by line down the image, and picture by picture in a moving sequence). Differences are formed between each pixel and a nearby one temporarily stored in the delay element. The delay element is fixed and corresponds to selecting the pixel immediately to the left, the pixel immediately above, the difference or 'prediction error' is seldom exactly zero because even in regions with no detail or in stationary areas the camera system usually adds some noise. However, the variance of the prediction error is generally substantially less than that of the original and further techniques such as entropy coding can be used to represent the data with fewer bits than the original pixels. The original pixels can be reconstructed from the prediction errors by a decoder which consists of just the components in the outlined box.

As well as using just one previous pixel as the predictor, more complex schemes employing several previous pixels together may be used, for example a local average of a few spatially adjacent pixels may provide a better estimate of the new one by averaging out noise, or estimating and extrapolating the local slope of the image function. When predicting from one picture to the next in a moving video, the single pixel predictor is of course very good in regions of the image where there is no change (or 'motion'), but often very bad where there is actually motion. Unfortunately the latter is precisely the area where most of data needs to be sent and compression is most needed. This observation led to the introduction of conditional replenishment, whereby in areas of the picture where there is no temporal change at all, it is sufficient for a decoder to simply display the pixels from the previous frame again. In changed areas, new pixels are transmitted using prediction from spatially adjacent pixels. Omitting the redundant transmission of unchanged pixels gives a big gain in compression.

Vector Quantisation

Wavelets and DCT exploit the correlation of adjacent pixels by generating coefficients that have small or zero value if adjacent pixels are similar. Another way to exploit this correlation is vector quantisation (VQ) [23]. A block of pixels or prediction errors is quantised as one unit (a vector), rather than a pixel at a time. Typically, 4×4 sized blocks might be used, and a codebook of hundreds or thousands of possible example block patterns would be searched to find the pattern giving the best match. Only the index number of that entry in the codebook need then be transmitted. VQ has a simple decode operation, being merely a table look-up, but the encoding process can be very demanding because at worst each block will require a full search of the codebook. In practice, various ways of pre-normalising the data and structuring the codebooks can reduce this problem. VQ can also be used to quantise the data resulting from other processes such as DCT or wavelet transformation, and both theoretically and practically should perform better than an equivalent number of independent scalar quantisers. However, there are some problems concerning construction of codebooks to adequately cover the range of vectors to be quantised, and of the processing power required to perform the codebook searching at the encoder.

Transform Coding

Transform coding is a mathematical operation which transforms a set of spatial image pixel values to another set of transform coefficient values. The transform coding is a linear process so the information remains the same. As the result of linear process, the number of coefficients produced is also equal to the number of pixels transformed. The achievement of the compression lies in the fact that the image energy of most natural scenes is mainly concentrated in the low frequency region, and hence into a few transform coefficient. The insignificant coefficients then can be discarded and still retain the good reconstructed image quality. This transform coding then is one of the lossy compression. There are many types of transform coding have been tried for picture coding, such as Fourier, Karhunen-Loeve, Walsh-Hadamard, lapped orthogonal, DCT, and recently, wavelets. However the basic ideas of every transform coding are the same. Among those transform coding, the DCT is one of the most popular transform coding adopted by coding standards because of its ability not to generate spurious spectral components, so coding efficiency is high and "blocking artifacts" [24] are relatively low.

2.4.2 Temporal redundancy reduction

The widely common used coding technique for removing temporal redundancy is motion compensation. This technique is an extension of the prediction method, in which the delay for each pixel is varied dynamically throughout each still image to find the place in the previous image which is the best prediction. The corresponding horizontal and vertical offsets which refer to that best prediction are included in the coded bit stream. Ideally these offsets would be sent for every pixel but the overhead of doing this is much more than the savings obtained from the smaller prediction error. Instead, a single pair of offsets is applied to many pixels, a 16×16 block of them being a common compromise. Motion-compensated prediction works fairly well on videoconference type scenes where the gross movements of people's heads and bodies are represented to a first approximation by simple translations of their positions from the previous image. However, real world objects also undergo rotation, deformation, occlusion and so on, which are not well modelled by the above block-based motion compensation. More complex schemes have been devised to model these higher order changes in image structure, including representing the motion by complex warping functions, and delineating the boundaries between areas with different motions.

2.4.3 Entropy coding

Entropy coding is a general term for lossless data compression methods which rely on the statistics of a set of 'events' to be compressed. In most practical cases this means the overall frequency of occurrence of the various events in a set. The length of the codeword used to convey a particular event is matched to the likelihood of it occurring. Shorter codes are used for the frequent events and longer codes for those appearing less often, hence the term 'variable length coding'. For example, in Morse code a single dot represents the frequently occurring letter 'e' whereas the rarer letter 'q' is encoded as dash, dash, and dot, dash. The optimum length of a codeword for an event of probability p is $\log_2 (1/p)$ [25]. If all possible events in the set are assigned codewords according to this formula, the overall bit rate required to send events from that set with the given probabilities will be minimised. Thus for an event with a probability of occurrence of 1/8 the optimum codeword length is 3 bits. Unfortunately in practical cases the formula will yield non-integer values for codeword lengths, and a means is needed to optimally assign integer length codewords to events. Huffman [26] devised an algorithm to do this, hence the term Huffman coding (which is often wrongly used as a term for variable length coding, which need not, in general, be optimal). Huffman's method assigns integer length codewords to events, but this results in a loss of efficiency since the theoretically optimal codeword lengths are non-integer. Arithmetic coding [6] is a technique which overcomes this by not having a one–to-one mapping of events to codewords, and thus comes closer to the optimum compression. However, it is also more difficult to resynchronise the decoder in the presence of errors. Entropy coding gives very little compression if applied directly to image signals because the distribution of the brightness levels is fairly uniform. However, prediction errors have a very peaked distribution centred about zero and variable length coding is very worthwhile.

2.5 MPEG Coding Standard

The MPEG compression algorithm is designed for compression of full-motion video. The compression method uses inter frame compression and can achieve compression ratios of 200:1 through storing only the differences between successive frames. The MPEG approach is optimised for motion-intensive video applications, and its specification also includes an algorithm for the compression of audio data at ratios ranging from 5:1 to 10:1. The MPEG first-phase standard (MPEG-1) is targeted for compression of 320x240 full motion video at rates of 1 to 1.5 Mbits/s in applications, such as interactive multimedia and broadcast television. MPEG-2 standard is intended for higher resolutions, similar to the digital video studio standard, CCIR 601, EDTV, and further leading to HDTV. It specifies compressed bit streams for high-quality digital video at the rate of 2-80 Mbits/s. The MPEG-2 standard supports interlaced video formats and a number of features for HDTV. The MPEG-2 standard also addresses scalable video coding for a variety of applications, which need different image resolutions, such as video communications over ISDN networks using ATM [27, 28]. The MPEG-4 standard is intended for compression of full-motion video consisting of small frames and requiring slow refreshments. The data rate required is 9-40 kbits/s, and the target applications include interactive multimedia and video telephony. This standard requires the development of new model-based image coding techniques for human interaction and low bit rate speech coding techniques [27]. Table 2.1 illustrates various motion video formats and corresponding MPEG parameters.

FORMAT	VIDEO PARAMETERS	COMPRESSED BIT RATE
SIF	352 × 240 at 30 Hz	1.2-3 Mbits/s MPEG-1
CCIR 601	720 × 486 at 30 Hz	5-10 Mbits/s MPEG-2
EDTV	960 × 486 at 30 Hz	7-15 Mbits/s MPEG-2
HDTV	1920 × 1080 at 30 Hz	20-40 Mbits/s MPEG-2

Table 2.1 Parameters of MPEG Algorithms

The MPEG algorithm is intended for both asymmetric and symmetric applications. Asymmetric applications are characterised by frequent use of the decompression process, while the compression process is performed once. Examples include movies-on-demand, electronic publishing, and education and training. Symmetric applications require equal use of the compression and decompression processes. Examples include multimedia mail and videoconferencing.

When the MPEG standard was conceived, the following features were identified as being important: random access, fast forward/reverse searches, reverse playback, audio-visual synchronisation, robustness to errors, ease of editing, format flexibility, and cost-trade-off. The MPEG standard consists of three parts: synchronisation and multiplexing of video and audio; video; and audio.

2.5.1 MPEG Frame Structure

In the MPEG standard, frames in a sequence are coded using three different algorithms, as illustrated in Figure (2.3).

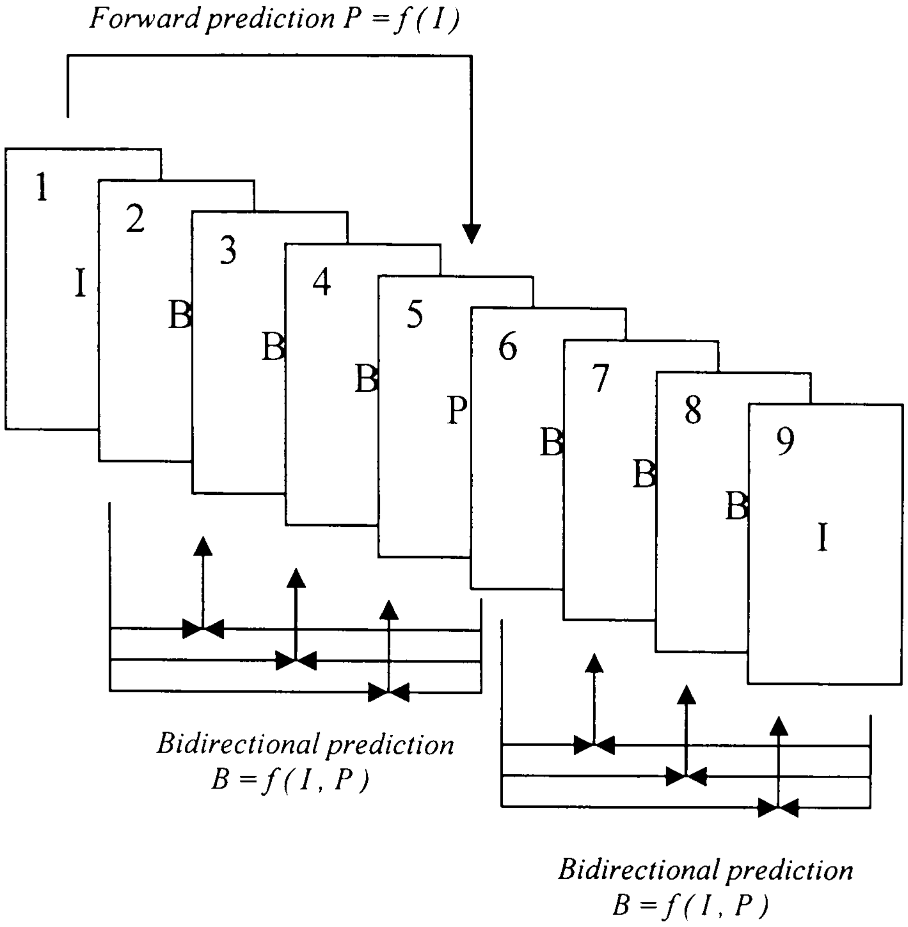


Figure (2.3) Types of frames in the MPEG standard

I frames (intra images) are self-contained and coded using a DCT-based technique similar to JPEG. **I** frames are used as random access points to MPEG streams, and they give the lowest compression ratios within MPEG.

P frames (predicted images) are coded using forward predictive coding, where the actual frame is coded with reference to a previous frame (**I** or **P**). This process is similar to H.261 predictive coding, except the previous frame is not always the closest previous, as in H.261 coding. The compression ratio of **P** frames is significantly higher than of **I** frames.

B frames (bidirectional or interpolated images) are coded using two reference frames, a past and a future frame (which can be **I** or **P** frames). Bidirectional or interpolated coding provides the highest amount of compression.

Note that in Figure (2.3), the first three **B** frames (2, 3 and 4) are bidirectionally coded using the past frame **I** (frame 1) and the future frame **P** (frame 5). Therefore, the decoding order will differ from the encoding order. The **P** frame 5 must be decoded before **B** frames 2, 3, and 4, and **I** frame 9 before **B** Frames 6, 7, and 8. If the MPEG sequence is transmitted over the network, the actual transmission order should be {1, 5, 2, 3, 4, 9, 6, 7, 8}.

The MPEG application determines a sequence of **I**, **P**, and **B** frames. If there is a need for fast random access, the best resolution would be achieved by coding the whole sequence as **I** frames (MPEG becomes identical to MJPEG). However, the highest compression ratio can be achieved by incorporating a large number of **B** frames. The following sequence has been proven to be very effective for a number of practical applications [27]:

(I B B P B B P B B) (I B B P B B P B B)...

In the case of 25 frames/s, random access will be provided through nine still frames (**I** and **P** frames), which is about 360 ms [27]. On the other hand, this sequence will allow a relatively high compression ratio. If we take a simple example and assume that the compression ratio for **I** frames is 1:10, for **P** frames is 1:40, and for **B** frames is 1:90, an average compression ratio for this MPEG sequence can be found by.

$$Average\ Compression\ Ratio = \frac{1}{N}\sum_{i=1}^{3}\left(n_i \times Compression\ Ratio\right) \text{ with } N = n_1+n_2+n_3 \quad (2.4)$$

$$= \frac{1\times 10}{9} + \frac{2\times 40}{9} + \frac{6\times 90}{9}$$

$$= 70$$

2.5.2 MPEG Video Encoder and Decoder

The block diagram of the MPEG encoder is given in Figure (2.4), while the MPEG decoder is shown in Figure (2.5).

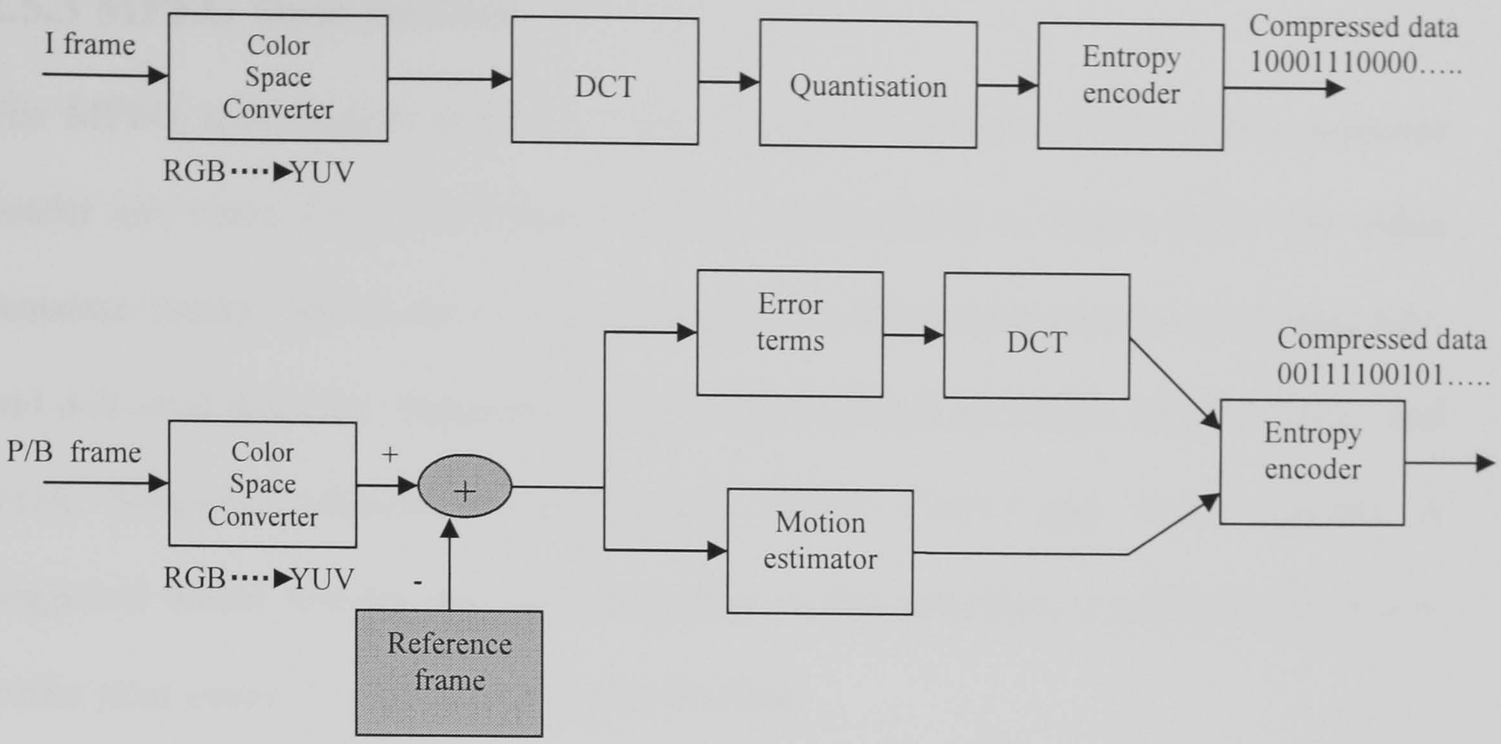


Figure (2.4) the block diagram of the MPEG encoder.

I frames are created similar to JPEG encoder pictures, while **P** and **B** frames are created in terms of previous and future frame. The motion vector is estimated, and the difference between the predicted and actual blocks (error terms) are calculated. The error terms are then DCT encoded and the entropy encoder is used to produce the compact code.

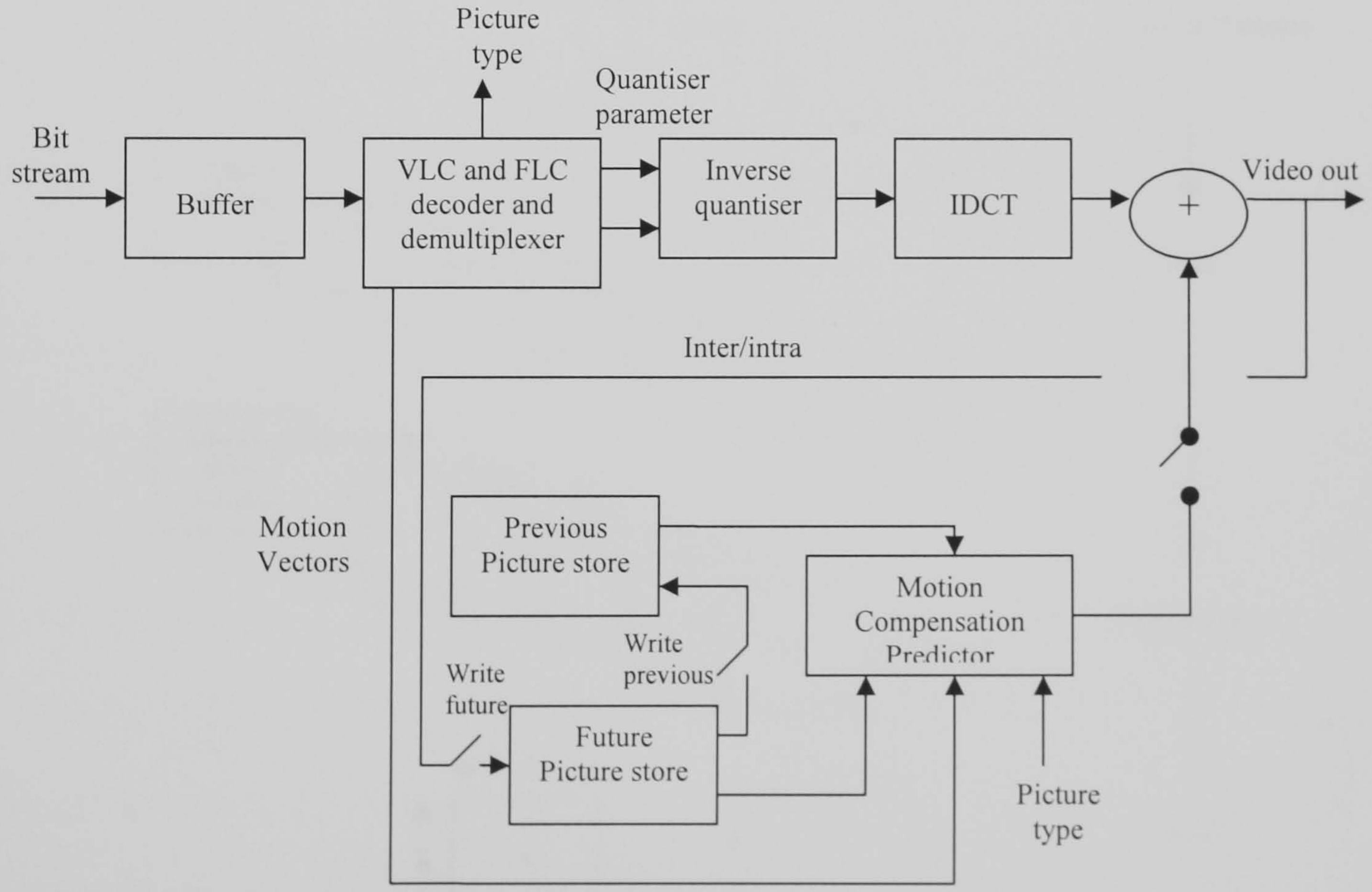


Figure (2.5) the block diagram of the MPEG decoder

2.5.3 MPEG Data Stream

The MPEG specification defines a "video sequence" composed of a video sequence header and many Group-Of-Pictures (GOP), as illustrated in Figure (2.5). The video sequence header defines the video format, picture dimensions, aspect ratio, frame rate, and delivered data rate. Supported video formats include CCIR601, HDTV (16:9), and VGA. Supported chroma formats include "4:2:0" (YUV) and "4:4:4" (RGB). A suggested buffer size for the video sequence is also specified, a number intended to buffer jitter caused by differences in decode time.

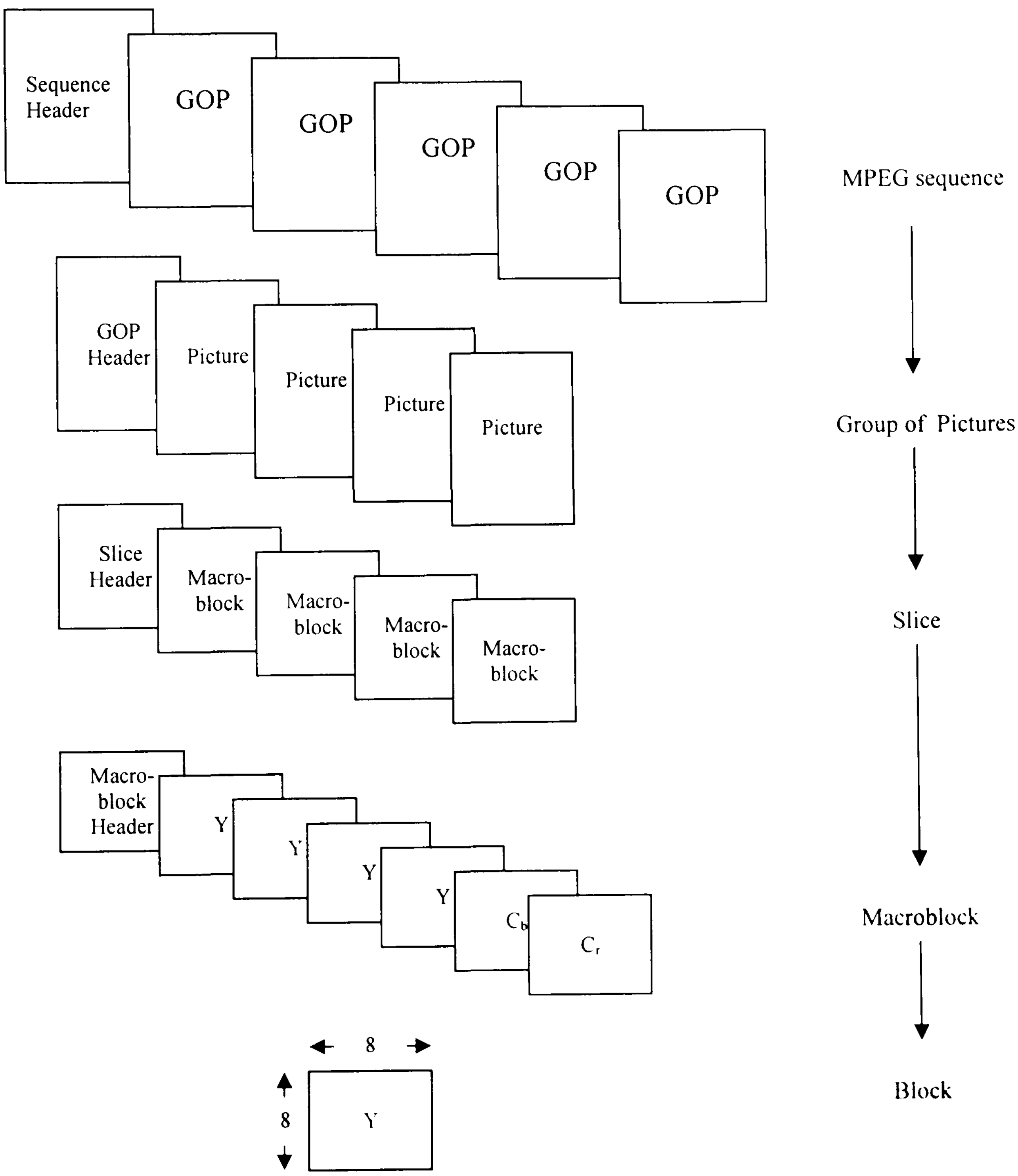


Figure (2.6) MPEG Data Stream

A GOP contains pictures that may be encoded into one of three supported compression formats. The GOP header contains a starting time for the group, and can therefore be used as a point of random access. Each frame within the GOP is numbered, and its number coupled with the GOP start time and the playback frame rate determines its playback time. Each picture is subdivided into "slices" and then into "macroblocks." A macroblock is composed of four 8x8 blocks of luminance data, and typically two 8x8 blocks of chrominance data, one C_r and one C_b.

I Picture Format

These pictures are encoded by transformation into DCT space, quantisation of the resultant coefficients, and entropy coding of the result. Transformation into DCT space is performed by an 8x8 DCT. Quantisation is performed by reference to a user-loadable quantisation table modified by a scale factor. This mechanism supports adaptive quantisation at the cost of additional complexity - although 30% improvement in compression is claimed [29]. After quantisation, the resulting coefficients are reordered in zig-zag order, run-length coded, variable-length coded, and entropy coded. The resulting data stream should roughly show JPEG levels of compression.

P Picture Format

The **P** (Predicted) picture format introduces the concept of motion compensation. Each macroblock is coded with a vector that predicts its value from an earlier **I** or **P** frame. The decoding process copies the contents of the macroblock-sized data at the address referenced by the vector into the macroblock of the **P** frame currently being decoded. Five bits of resolution are reserved for the magnitude of the vector in each of the x and y directions, meaning that 1024 possible data blocks may be referenced by the predicted macroblock. However, eight possible magnitude ranges may be assigned to those five bits, meaning as many as 8192 macroblocks might have to be evaluated to exhaustively determine the best vector. Each evaluation might require testing as many as 384 pixels, and a further complexity is seen in performing fractional interpolation of pixels (vector motions as small as 1/2 pixel are supported). Finally, the difference between the prediction and the macroblock to be compressed may be encoded in like fashion to **I** frame encoding above.

B Picture Format

The **B** (Bidirectional prediction) picture format is calculated with two vectors. A backward vector references a macroblock-sized region in the previous **I** or **P** frame, the forward vector references a macroblock-sized region in the next **I** or **P** frame. For this reason, **I** and **P** frames are placed in the coded stream before any **B** frames that reference them. The macroblock-sized regions referenced by the motion compensation vectors are averaged to produce the motion estimate for the macroblock being decoded. As with **P** frames, the error between the prediction and the frame being encoded is compressed and placed in the bit stream. The error factor is decompressed and added to the prediction to form the **B** frame macroblock.

Many demanding technical issues are raised by the MPEG specification. These include fast algorithms for the DCT, fast algorithms for motion vector estimation, algorithms for adaptive quantisation, and decompression in environments that allow some errors.

2.5.4 Motion Estimation and Compensation

The coding process for **P** and **B** frames includes the motion estimator, which finds the best matching block in the available reference frames. **P** frames always use forward prediction, while **B** frames always use bidirectional prediction, also called motion-compensated interpolation, as illustrated in Figure (2.7)[30,31].

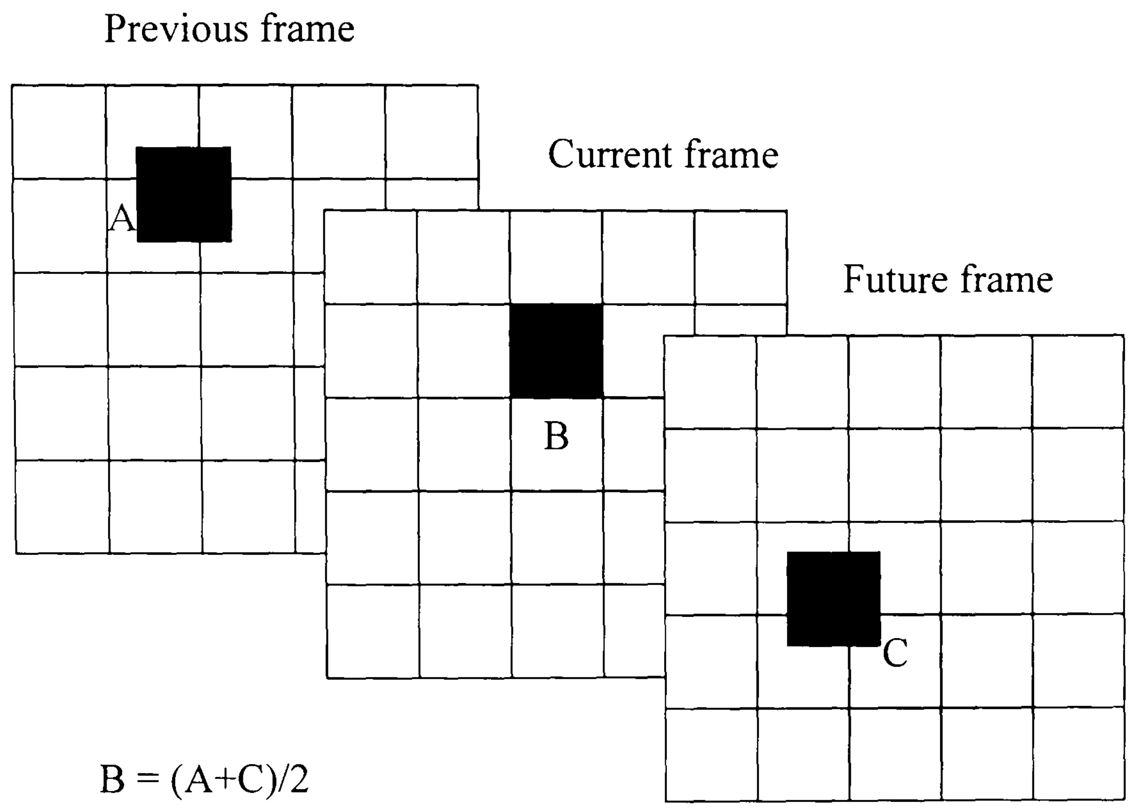


Figure (2.7) Motion compensated interpolated implemented in MPEG. Each block in the current frame is interpolated using the blocks from a previous and a future frame.

B frames can use forward, backward prediction, or interpolation. A block in the current frame (**B** frame) can be predicted by another block from the past reference frame (**B** = **A** → forward prediction), or from the future reference frame (**B** = **C** → backward prediction), or by the average of two blocks (**B** = (**A** + **C**)/2 → interpolation). Motion estimation is used to extract the motion information from the video sequence.

The MPEG standard does not specify the motion estimation technique; however, block matching techniques are likely to be used. There is a number of block-matching techniques has been developed for motion estimation such as the full search algorithm, and the three-step-search algorithm. However the technique which is widely adopted in most applications is the three-step search.

2.5.5 MPEG Audio Encoder and Decoder

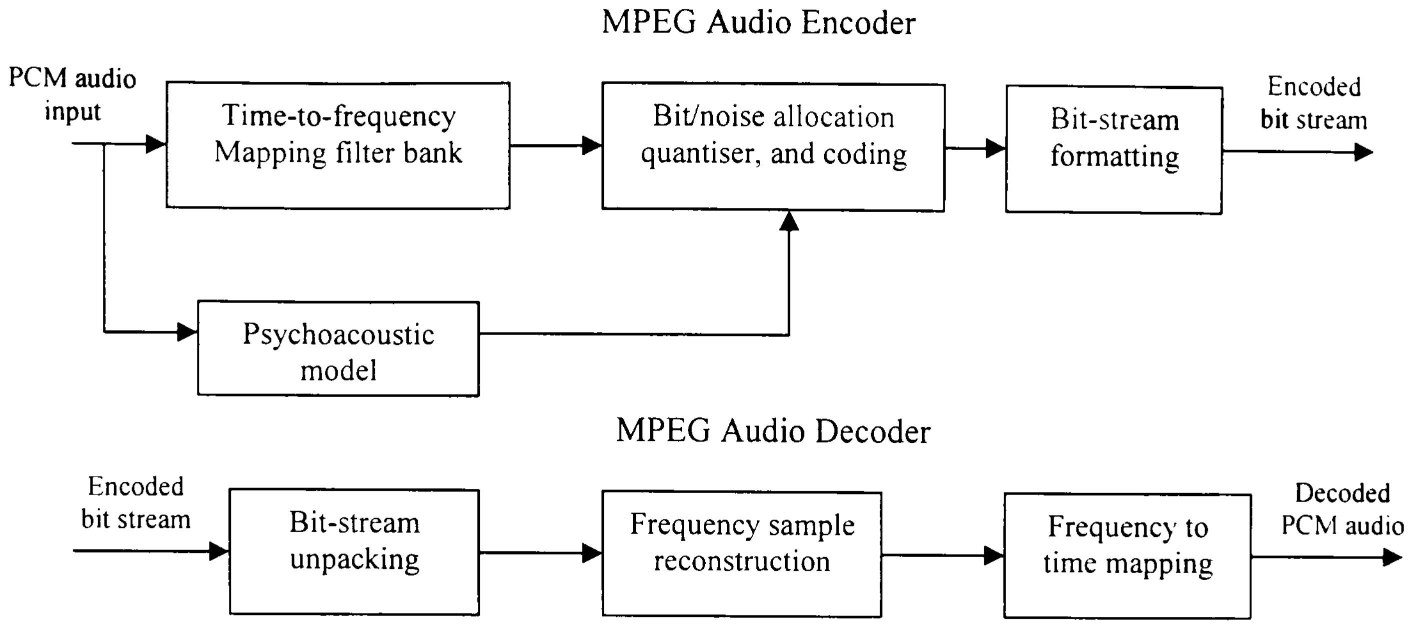
The MPEG standard also covers audio compression. MPEG uses the same sampling frequencies as compact disc digital audio (CD-DA) and digital audio tapes (DAT). Besides these two frequencies, 44.1 KHz and 48 KHz. 32 KHz are also supported. all at 16 bits. The audio data on a compact disc, with two channels of audio samples at 44.1 KHz with 16 bits/sample, require a data rate of about 1.4 Mbits/s [32]. Therefore, there is a need to compress audio data as well. Existing audio compression techniques include m-law and Adaptive Differential Pulse Code Modulation (ADPCM), which are both of low complexity, low compression ratios, and offer medium audio quality. The MPEG audio compression algorithm is of high complexity, but offers high compression ratios and high audio quality. It can achieve compression ratios ranging from 1:5 to 1:10.

The MPEG audio compression algorithm is comprises the following three operations:

i. The audio signal is first transformed into the frequency domain, and the obtained spectrum is divided into 32 non-interleaved subbands.

ii. For each subband, the amplitude of the audio signal is calculated, and the noise level determined by using a "psychoacoustic model." The psychoacoustic model is the key component of the MPEG audio encoder and its function is to analyse the input audio signal and determine where in the spectrum the quantisation noise should be masked.

iii. Finally, each subband is quantised according to the audibility of quantisation noise within that band.



Figure(2.8) Block Diagram of MPEG Audio Encoder and Decoder

The input audio stream simultaneously passes through a filter bank and a psychoacoustic model. The filter bank divides the input into multiple subbands, while the psychoacoustic model determines the signal-to-mask ratio of each subband. The bit or noise allocation block uses the signal-to-mask ratios to determine the number of bits for the quantisation of the subband signals with the goal to minimise the audibility of the quantisation noise. The last block performs entropy (Huffman) encoding and formatting the data. The decoder performs entropy (Huffman) decoding, then reconstructs the quantised subband values, and transforms subband values into a time-domain audio signal. The MPEG audio standard specifies three layers for compression: layer 1 represents the most basic algorithm and provides the maximum rate of 448 kbits/s, layers 2 and 3 are enhancements to layer 1 and offer 384 kbits/s and 320 kbits/s, respectively. Each successive layer improves the compression performance, but at the cost of the greater encoder and decoder complexity. A detailed description of audio compression principles and techniques is reported elsewhere [27,32, 33,34]

2.5.6 MPEG INTERLEAVED A/V DATA STREAM

The MPEG standard specifies a syntax for the interleaved audio and video data streams. An audio data stream consists of frames, which are divided into audio access units. Audio access unit consists of slots, which can be either four bits at the lowest complexity layer (layer 1), or one byte at layers 2 and 3. A frame always consists of a fixed number of samples. Audio access unit specifies the smallest audio sequence of compressed data that can be independently decoded. The playing times of the audio access units of one frame are 8 ms at 48 KHz, 8.7 ms at 44.1 KHz, and 12 ms at 32 KHz [27]. A video data stream consists of six layers, as shown in Table 2.2.

At the beginning of the *sequence layer* there are two entries: the constant bit rate of a sequence and the storage capacity that is needed for decoding. These parameters define the data buffering requirements. A sequence is divided into a series of GOPs. Each GOP layer has at least one **I** frame which is the first frame in GOP, so random access and fast search are enabled. GOPs can be of arbitrary structure (**I**, **P**, and **B** frames) and length. The GOP layer is the basic unit for editing an MPEG video stream. The *picture layer* contains a whole picture (or a frame).

Syntax Layer	**Functionality**
Sequence Layer	Context Unit
Group of Pictures Layer	Random Access Unite: Video Coding
Picture Layer	Primary Coding Unit
Slide Layer	Resynchronise Layer Unit
Macroblock Layer	Motion Compensation Unit
Block Layer	DCT unit

Table 2.2 Layers of MPEG Video Stream Syntax

This information consists of the type of the frame (**I**, **P**, or **B**) and the position of the frame in the display order. The bits corresponding to the DCT coefficients and the motion vectors are contained in the next three layers: *slice*, *macroblock*, and *block* layers. The block is a (8x8) DCT unit, the macroblock is a (16x16) motion compensation unit, and the slice is a sting of macroblocks of arbitrary length. The slice layer is intended to be used for re-synchronisation during a frame decoding when bit errors occur.

2.6 Summary

In this chapter, the history overview of video coding was explained at the beginning. The history overview guided us to the processes of the development from the earliest stage up until the current stage the video can be compressed. This history shows the evolution of the video coding from basic techniques to advanced techniques. Later the video compression characteristics were explained. The common terms which are frame rate, frame dimensions and pixel depth were described. These terms are commonly used in the video coding. The compression ratio, image quality and compression speed were also explained. The compression speed or the speed of operation is the main characteristic which is used to justify the capability of the search algorithms. Also the image quality assessment is one of the main metrics to justify how good the search algorithms can achieve in terms of the capability of the prediction. The image quality assessment can be classified into the subjective quality and objective quality assessment. The widely used objective quality assessment is PSNR. In addition to PSNR, the subjective quality should be also assessed to support the performance of the algorithms.

In section 2.4, various video coding techniques were explained and the main video coding techniques were briefly explained. Most of these techniques are used in the well-known standards such as MPEG. Finally the MPEG standard was explained in details. The MPEG standard is used as a main reference for motion compensation in this research. The motion estimation is one of the methods used in MPEG standard to exploit the temporal redundancy. The motion estimation takes place in the P frames and B Frames in MPEG standard where P frame is forward predicted and B is bidirectionally predicted. The motion compensation takes place in Macroblock layer in the motion compensation unit. The details of the motion compensation will be discussed later in Chapter 3.

Chapter 3

MOTION COMPENSATION

3.1 Introduction

In recent years, Video compression has played a vital role in data storage and transmission as it has been the main focus in many applications such as multimedia communications, remote monitoring, videophones, videoconference etc. The effective video compression mostly employs the hybrid coding configuration which involves both intra frame and inter frame coding. The intra frame compression is a vital means of exploiting the spatial redundancies and inter frame compression is used for exploiting the temporal redundancies. Due to the slow movement of video sequences such as 'head-and shoulders', the two consecutive frames will not change dramatically. So the current frame can be predicted from the translation of the previous frame. The method that is widely used in video transmission is Motion Compensation prediction (MC). The vital part of Motion Compensation is Motion Estimation, which is used for extracting the motion activity that exists between the frames. Block Matching Algorithms (BMA) are popular methods for Motion Estimation because of their simplicity and ease of implementation. Most of current video communication systems and standards such as ISO MPEG-1/2 and ITU-T H.263 employ BMA for motion estimation. This chapter provides the overview of the inter frame compression techniques, motion compensation and motion scheme.

The overview of the inter frame compression techniques includes the most commonly used techniques which are sub-sampling, difference coding, block-based difference coding and block-base motion compensation are explained. This chapter also included the principle of motion compensation. The implementation of the motion compensated prediction is explained. The details of motion compensation scheme are pointed out. Each steps of motion compensation is described.

3.2 Overview of Inter frame Coding Technique

Inter frame coding technique is the compression technique that is designed for sequence of video frames, rather than a single image. It assumes that parts of the current frame can be modelled as a translation of the previous frame, due to the similarity of the two consecutive frames known as temporal redundancy. Inter frame compression aims to exploit these similarities between successive frames to reduce the amount of data required for storing or transmitting over the network. There is a large number of inter frame compression techniques have been introduced in the past. The complexity of the inter frame techniques is various and depends on the application. However the aims of the most inter frame techniques are the same. They attempt to more efficiently describe the sequence by reusing parts of frames the receiver already has, in order to construct new frames.

3.2.1 Sub-sampling

Sub-Sampling is the most classic method of the compression. Sub-sampling aims to reduce the amount of data to be transmitted by transmitting only some of the frames. Sub-sampled digital video might, for example, contain only every third frame. Neither the viewer's brain nor the decoder notice the missing frames at the receiving end due to the fast movement of the video at 30 frames/s.

3.2.2 Difference Coding

Difference coding, or *conditional replenishment*, is a very simple inter frame compression method and easy to implement. This method just compares the two consecutive frames and finds the difference between these two frames. Only pixels that have changed are updated and transmitted instead of all the pixels. This method can reduce the data transmitted because only a fraction of the number of pixel values is transmitted. The images below are successive frames from the "Claire" video sequence and illustrates how, frequently, very little changes from one frame to the next.

Figure (3.1) Consecutive frames from "Claire" Video sequence

If every changed pixel must be updated, then this coding is lossless compression. In the difference coding, there is an overhead transmitted along with the pixels values. This overhead contains the information to indicate which pixels are to be updated. If the number of pixels to be updated is large, then this overhead can adversely affect compression. To alleviate this problem two modifications are introduced. However, these modifications cause some loss. First modification is based on the assumption that there are several pixels which have their intensity changed slightly. So when coding is allowed to be lossy, only pixels that change significantly need be updated. Thus, not every pixel change will be updated. The second modification is the difference coding that operates at the block level instead of the pixel level. This modification leads to the technique called block-based difference coding.

3.2.3 Block-Based Difference Coding

If the frames are divided into non-overlapping blocks and each block is compared with its counterpart in the previous frame, then only blocks that change significantly need be updated. If, for example, only those blocks of the Claire frame that contain head and shoulder were updated, the resulting image might be an acceptable substitute for the original. Updating whole blocks of pixels at once reduces the overhead required to specify where updates take place. The 176x144 pixels (QCIF) in the Claire frame can be split into 396 8x8 pixel blocks. Significantly fewer bits are required to address one of 396 blocks than one of 25344 individual pixels. If pixels are updated in blocks, however, some pixels will be updated unnecessarily, especially if large blocks are used. Also, in parts of the image where updated blocks border parts of the image that have not been updated, discontinuities might be visible and this problem is worse when larger blocks are used. Clearly the choice of block size must be an informed one so as to achieve the best balance between image quality and compression. Block-Based Difference Coding can be further improved upon by compensating for the motion between frames. Difference Coding, no matter how sophisticated, is almost useless where there is a lot of motion. Only objects that remain stationary within the image can be effectively coded. If there is a lot of motion or indeed if the camera itself is moving, then very few pixels will remain unchanged. Even a very slow pan of a still scene will have too many changes to allow difference coding to be effective, even though much of the image content remains from frame to frame. To solve this problem it is necessary to compensate in some way for object motion.

3.2.4 Block-Based Motion Compensation

Block-based motion compensation, like other inter frame compression techniques, produces an approximation of a frame by reusing data contained in the frame's predecessor. This is completed in three stages. First, the frame to be approximated, the *current frame*, is divided into uniform non-overlapping blocks. Then each block in the current frame is compared to areas of similar size from the preceding or *past* frame in order to find an area that is similar. A block from the current frame for which a similar area is sought is known as a *target block*. The location of the similar or *matching block* in the past frame might be different from the location of the target block in the current frame. The relative difference in locations is known as the *motion vector*. If the target block and matching block are found at the same location in their respective frames then the motion vector that describes their difference is known as a *zero vector*. Current frame to be coded is divided into blocks. Motion vectors indicate where changed blocks in the current frame have come from. Unchanged blocks are marked by dots. Finally, when coding each block of the predicted frame, the motion vector detailing the position (in the past frame) of the target block's match is encoded in place of the target block itself. Because fewer bits are required to code a motion vector than to code actual blocks, compression is achieved. During decompression, the decoder uses the motion vectors to find the matching blocks in the past frame (which it has already received) and copies the matching blocks from the past frame into the appropriate positions in the approximation of the current frame, thus reconstructing the image. The effectiveness of compression techniques that use block-based motion compensation depends on the extent to which the following assumptions hold.

- Objects move in a plane that is parallel to the camera plane. Thus the effects of zoom and object rotation are not considered, although tracking in the plane parallel to object motion is considered.
- Illumination is spatially and temporally uniform. That is, the level of lighting is constant throughout the image and does not change over time.
- Occlusion of one object by another, and uncovered background are not considered.

Bidirectional motion compensation uses matching blocks from both a past frame and a *future frame* to code the current frame. A future frame is a frame that is displayed after the current frame. Considering the chess board example, suppose that a player is fortunate enough to have a once lost queen replace a pawn on the board. If the queen does not appear on the board before the current move then no block containing the queen can be copied from the previous state of play and used to describe the current state. After the next move, however, the queen might be on the board. If in addition to the state of play immediately before the current move, the state of play immediately following is also available to the receiver, then the current image of the chess board can be reproduced by taking blocks from both the past and future frames. Bidirectional compression is much more successful than compression that uses only a single past frame, because information that is not to be found in the past frame might be found in the future frame. This allows more blocks to be replaced by motion vectors. Bi-directional motion compensation, however, requires that frames be encoded and transmitted in a different order from which they will be displayed.

3.3 Motion Compensated Prediction

The most widely used technique to exploit temporal redundancy of video signals is motion compensation prediction. The basic behind motion compensation is to estimate the motion of objects and to use this information to build a prediction for successive frames. The process involves the estimation of the displacement between consecutive frames, which is called motion estimation (ME). The resulting motion information is then exploited in efficient inter frame predictive coding (MC). The Typical MC Predictive codec is depicted in Figure (3.2).

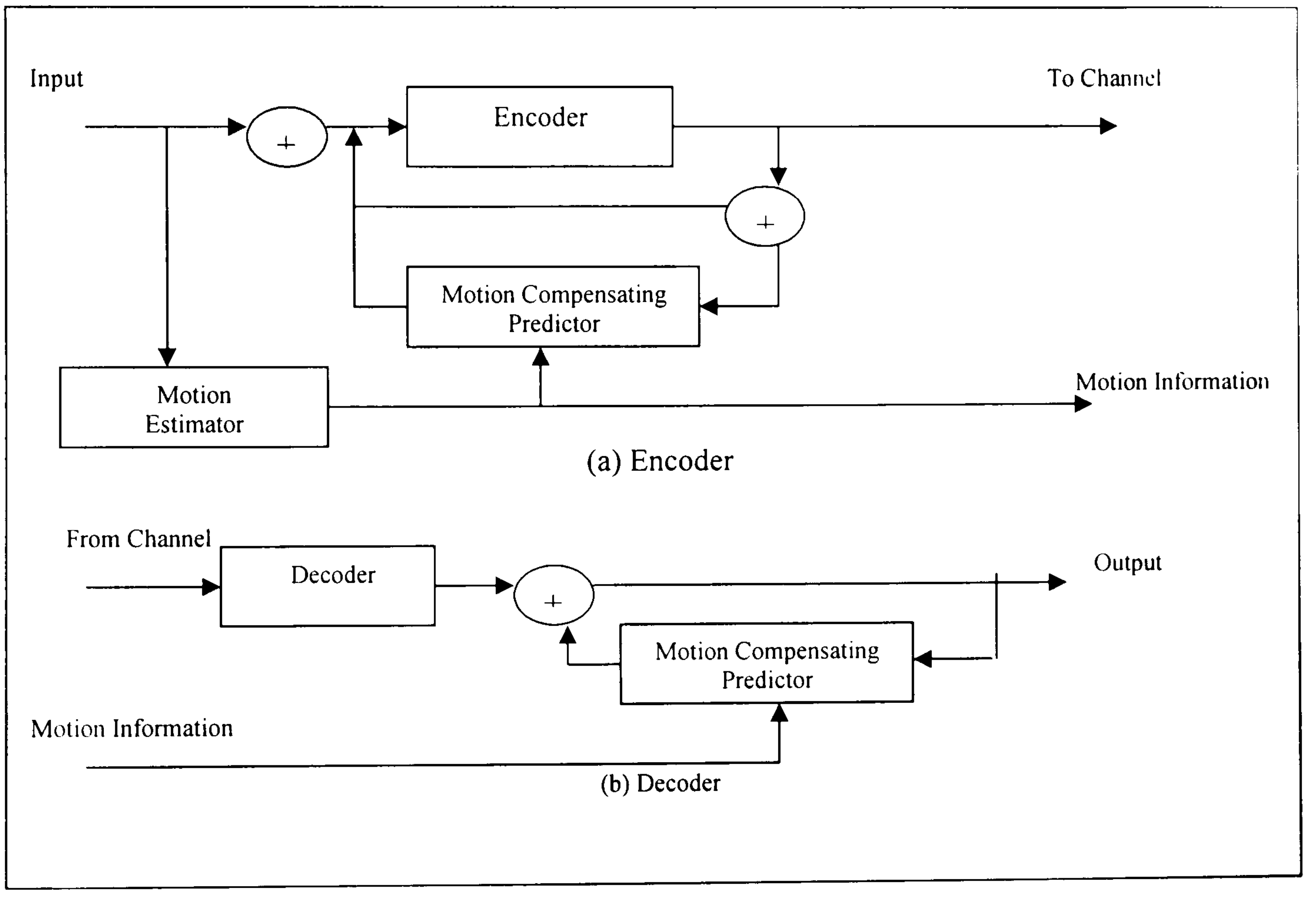


Figure (3.2) Typical Motion Compensation Prediction Codec

Motion Compensation process is implemented by several different stages. The Motion Compensation Scheme [35] [36] is shown in Figure (3.3) The Motion compensated video compression basically consists of the following stages.

- Frame Segmentation
- Search Threshold
- Block Matching
- Motion Vector Correction
- Vector Coding
- Prediction Error Coding

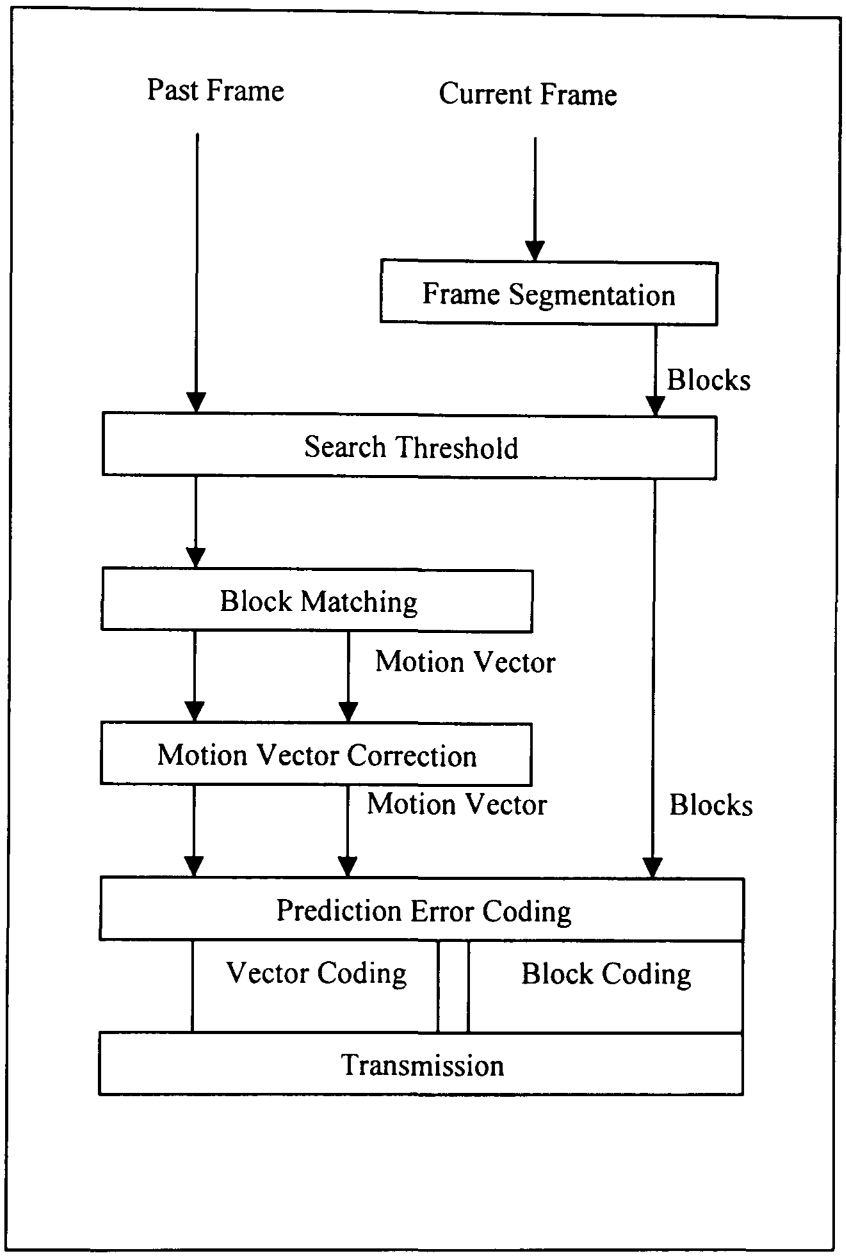


Figure (3.3) Motion Compensation Scheme

3.3.1 Frame Segmentation

Motion compensation is implemented by segmenting the current frame into perfectly tiling blocks. Ideally the frame dimensions are multiples of the block size and square blocks are most common. For architectural reasons block sizes of integer powers of 2 are preferred and so block size of 8 and 16 pixels predominate. Both MPEG and H.261 video standards use blocks of 16×16 pixels

3.3.2 Search Threshold

If the difference between the target block and the candidate block at the same position in the past frame is below some threshold then it is assumed that no motion has taken place and a zero vector is returned. Most video codecs employ a threshold in order to determine if the computational effort of a search is warranted.

3.3.3 Block Matching

Block matching is the most time consuming part of the encoding process. Each targeted block of the current frame is compared with the blocks in the past frame to find the best matching block. When the current frame is reconstructed by the receiver this matching block (from the past frame) is used as a substitute for the block from the current frame. The substitute block must be as similar as possible to the one it replaces. Thus a matching criterion, or distortion function, is used to quantify the similarity between the target block and candidate block.

3.3.4 Motion Vector Correction

The motion vector will be calculated after the best substitute, or matching block, has been found for the target block. The motion vector describes the location of the matching block from the past frame with reference to the position of the target block in the current frame. Nevertheless the motion vector might not correspond to the actual motion in the scene due to noise, weaknesses in the matching algorithm. Motion vector correction will be used after the vectors have been calculated in an attempt to correct them. The most widely used motion vector correction is smoothing techniques.

3.3.5 Vector coding

Once determined, motion vectors must be assigned bit sequences to represent them. As much of the compressed data will consist of motion vectors, the efficiency with which they are coded has a great impact on the compression ratio. In fact up to 40% of the bits transmitted by a codec might be taken up with motion vector data. Fortunately, the high correlation between motion vectors and their non-uniform distribution makes them suitable for further compression. This compression must be lossless. There are many lossless compression algorithms that are suitable for vector coding such as adaptive Huffman, Lempel-Ziv coding, variable length codes. The arithmetic and Huffman techniques performed best and the adaptive techniques using short term statistics performed better than those using long term statistics. The ISO/IEC video compression standard known as MPEG specifies variable length codes to be used for motion vectors. The zero vector, for example, has a short code, because it is the most frequently occurring.

3.3.6 Prediction Error coding

Although the battery of techniques described thus far can code video signals very successfully, they rarely generate perfect replicas of the original frames. Thus the difference between a predicted frame and the original uncompressed frame might be coded. Generally this is applied on a block by block basis and only where portions of the coded frame are significantly different from the original. Transform coding is most frequently used to achieve this and completely lossless coding is rarely a goal.

3.4 Summary

The motion estimation is the main contribution of this research. It is a part of motion compensation technique and one of the inter frame coding techniques. In the beginning of this chapter, the overview of the inter frame coding techniques was explained then the motion compensated prediction was broadly described. The main steps of motion compensated prediction consist of frame segmentation, search threshold, block matching, motion vector coding and prediction error coding. The explanation of each step was shown. The principle of motion compensation is to find the best matching block. The current frame can be predicted from the previous frame known as reference frame by using that matching block. Fundamentally, the current frame is divided into macroblocks. The size of macroblocks is a good trade-off between accuracy and computational complexity. Each macroblock is compared to macroblock in the reference frame using some matching criteria, and the best matching macroblock is selected. The most consuming time process is block matching. The block matching involved with search techniques and the matching criteria or distortion function. Both of these factors effect the operation time of motion compensation technique. To reduce the time consuming, both factors have to be taken into consideration. To understand the motion estimation, the nature of the moving scene is investigated and the problems of motion estimation will be pointed out in Chapter 4. The overview the motion estimation techniques will also be further explained.

Chapter 4

MOTION ESTIMATION TECHNIQUES FOR VIDEO CODING

4.1 Introduction

Motion Estimation (ME) is a vital part of many well-known video compression standards such as MPEG1-2 and H.261/263. It is a core technique of the motion compensation. The motion estimation is a method that aims to extract the motion information from a sequence of time-varying video sequence. It plays an important role in video compression as it leads to compression gains. The difficulty of this approach lays undoubtedly in estimating accurately the motion between two frames. This is the goal of motion estimation. The motion estimation processes fundamentally analyses the performance on the encoder side. Two consecutive video frames are compared and the algorithm to find the motion between the two frames. It results in motion description: dense or discrete motion field and affine parameters. Then the transcription phase takes the result of the motion estimation and tries to describe it in the most compact representation. The aim of this step is efficient coding (so as to reach the required bit rate). This step is reversed at the decoder in order to get the initial motion estimation parameters back. It can be done with or without losses, according to the type of transcription. The typical motion estimation processes are shown in the figure (4.1).

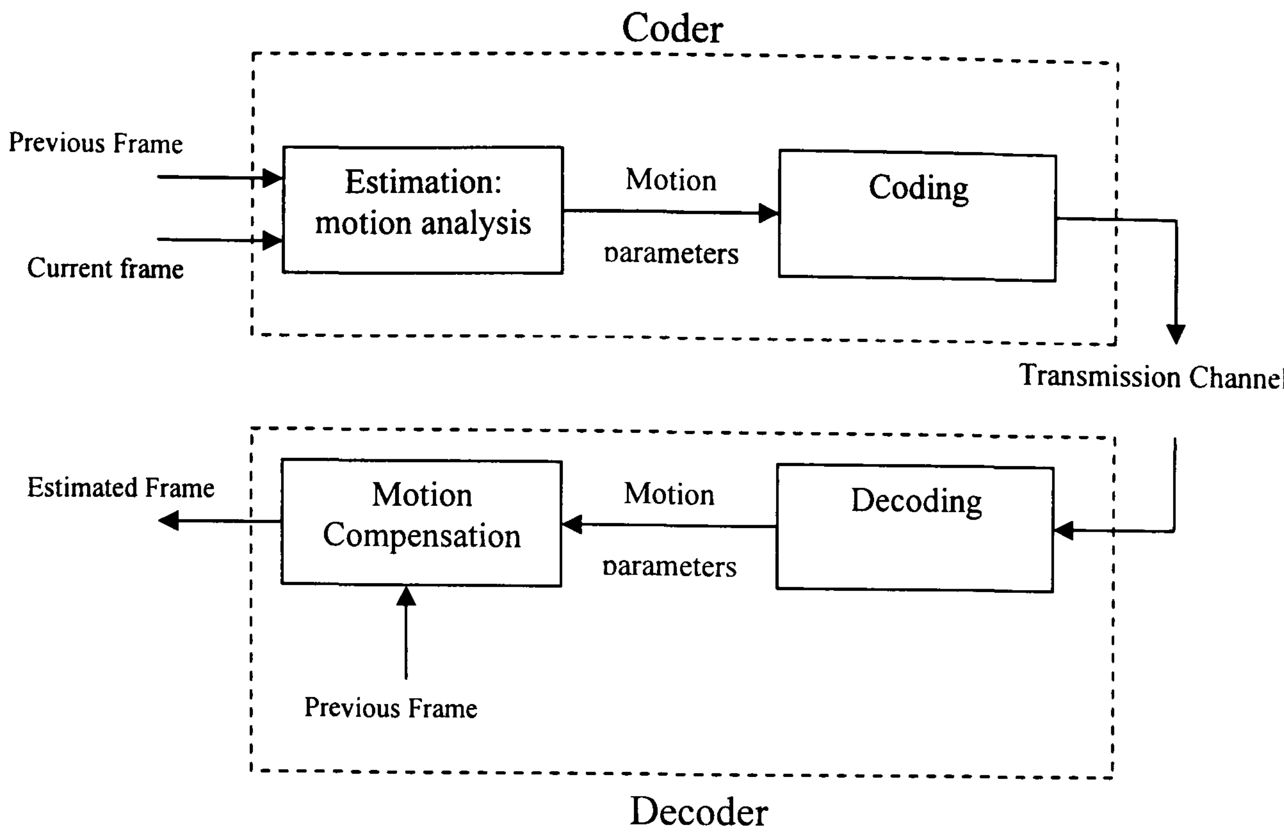


Figure (4.1) the Typical Motion Estimation Processes

The motion estimation algorithms have been studied and developed for various applications in the field of image processing. The most important applications are image sequence analysis, image sequence interpolation, restoration and image sequence coding (video coding). The objective of motion estimation is very different depending on the applications. In image sequence analysis, the motion information is used to extract useful features of the image sequence. In image sequence interpolation and restoration, adaptive filtering exploits the motion information in order to avoid blurring of moving objects. Finally, in image sequence coding motion information allows the reduction in the temporal redundancies between successive frames. As the above applications are very different in nature, they have led to very different motion estimation algorithms. For example, in image sequence coding the motion estimation is used for predictive coding. Thus the fact that it represents the motion in the scene is not an intrinsic goal. Therefore, the classical denomination of motion estimation is maybe not completely appropriate. Furthermore the motion information should usually be transmitted along with the image sequence as overhead information, unless the decoder is able to reconstruct the motion field on-line as in pixel recursive motion estimation algorithms [37]. However, as this constraint restricts severely the motion estimation technique the prediction should be causal, in the coming discussion the assumption that motion vectors have to be transmitted to the decoder is made. Consequently, the motion estimation algorithms should provide a good prediction as well as low coding cost of side information. The fact of obtaining a very precise motion field, in the sense of the motion present in a sequence, is secondary. This chapter provides details about the motion estimation techniques for video coding. The most classical techniques are presented to highlight the contribution put forward by the present work. The most important motion estimation algorithms are reviewed together with the main concepts behind other techniques such as optical flow, pixel recursive motion estimation. A number of motion estimation techniques are described in details.

4.2 The Fundamental of Motion Estimation

A video sequence consists of the two-dimensional frames of brightness resulting from the projection of a three-dimensional (3D) scene onto a two-dimensional (2D) plane. The 3D motion information of the objects also projected onto the 2D image plane. The presence of motion manifests itself on the image planes by changes of intensity values of pixels along the times axis. Therefore the 2D motion estimation is not actually real motion. The apparent motion which occurred in the 2D image plane is referred as optical flow. To compute 3D and 2D motion sequence, there are some well-known unsolved problems that occur. These problems are:

- The aperture problem which is illustrated on figure (4.2). Any operation that sees the moving edge through a local window A can only compute the component of motion perpendicular to the edge. It means that in figure (4.2), any of the vectors e, f or g would be convenient. The optical flow is therefore not uniquely determined by the local information in the changing image. The problem is not sufficiently constrained: it is ill-posed.

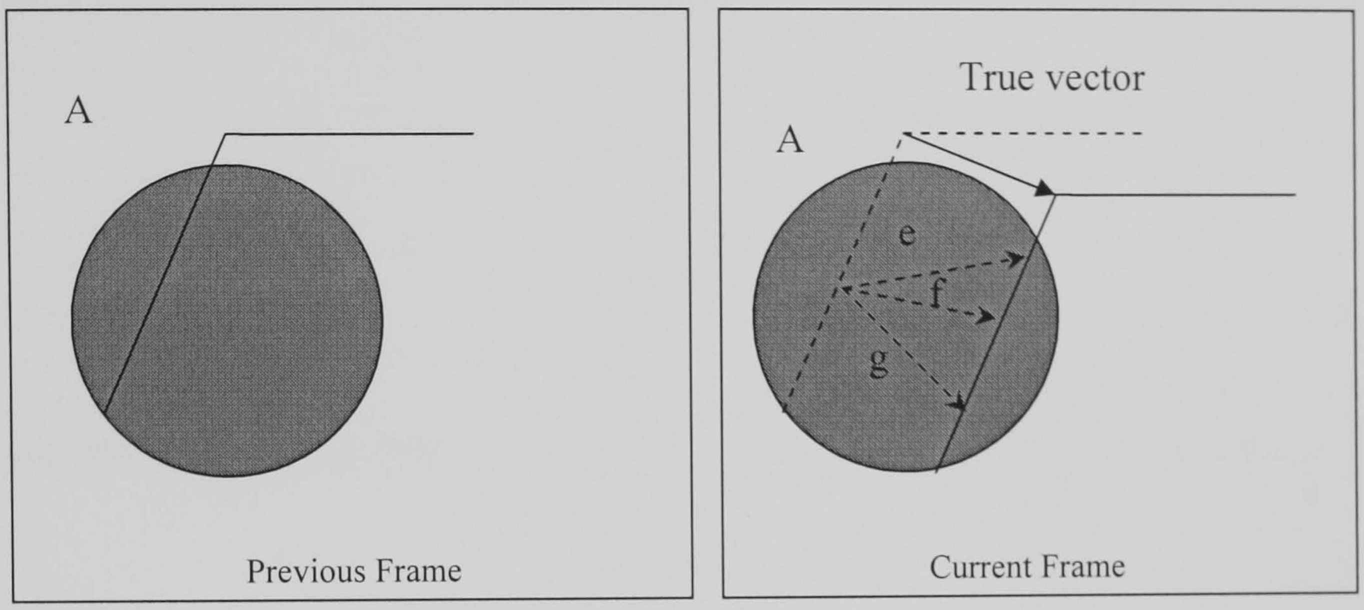


Figure (4.2) Aperture Problem

- The correspondence problem which is depicted in figure (4.2) prevents estimation algorithms from correctly putting in relation the intensity values of successive frames, and results from the spatiotemporal sampling achieved during digital image acquisition. Indeed, it is not always possible to respect the Nyquist frequency [38], particularly in the case of high spatial frequencies undergoing fast motions. A typical illustration of this kind of temporal aliasing is the "wheel" (figure 4.3): if the angular velocity of the wheel is greater than ($\pi / n \times$ frame rate), where n is the number of rails, there will be an ambiguity in the correspondence process and the wheel seems to turn in the opposite direction.

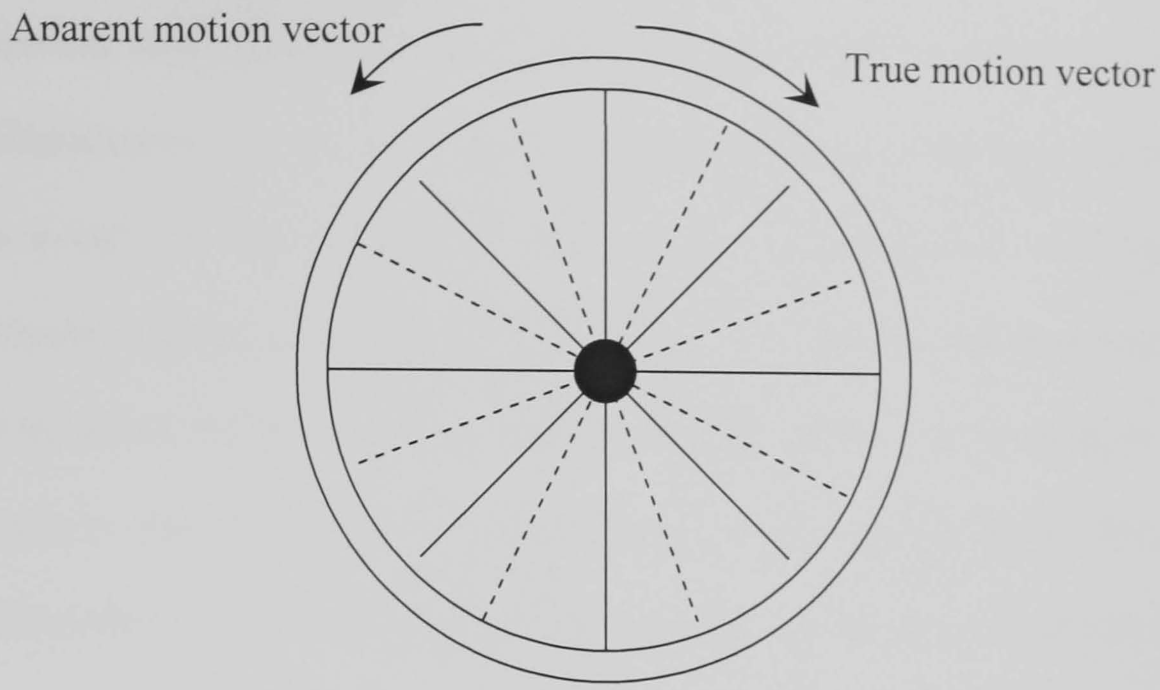
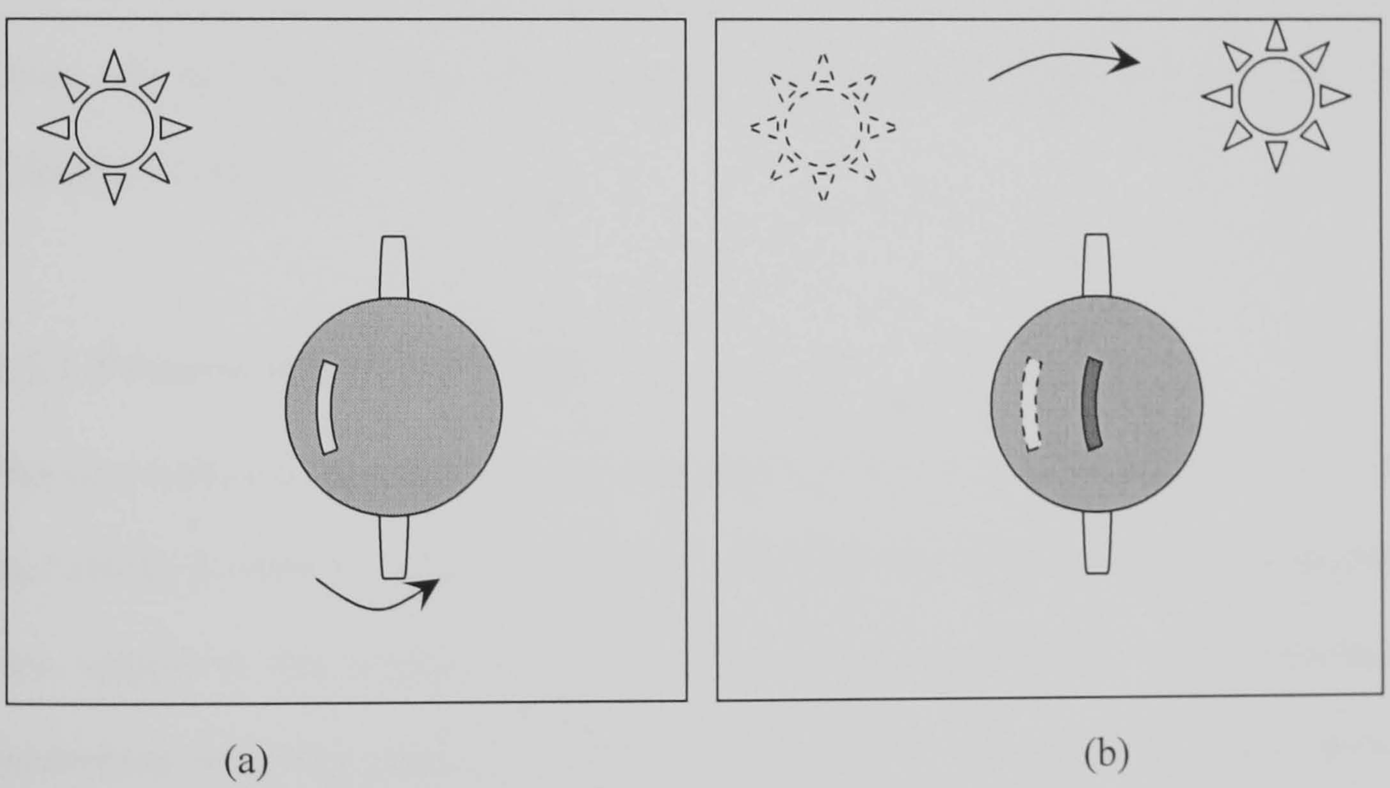


Figure (4.3) The Bicycle Wheel: ambiguity in the correspondence problem because aliasing



(a) (b)

Figure (4.4) the optical flow is not always equal to the motion Field.

(a) Null optical flow during non-null motion field. (b) Reverse situation.

- Because motion is estimated by establishing correspondences between successive images intensities, any noise (camera noise, quantisation, noise, etc.) will cause additional difficulties. Moreover, the illumination changes will be interpreted as motion effects and will distance the optical flow from the true motion field. Figure 4.4 (a) shows a smooth sphere rotating under constant illumination: the projected image does not change, yet the true motion field is non-null. On figure 4.4(b), a fixed sphere is illuminated by a moving source: shadow changes engender an optical flow, even if the true motion field is null.

- Occlusions between moving objects such as appearance or disappearance of objects parts (because of (un)covering), create regions where the observed intensities of previous frame do not have any correspondence in current frame. And the partial loss of the depth information does not provide enough information to recover the true motion.

To solve these problems, the constraints have to be added so to reach a unique solution. There are two main types of constraints which are Preservation Constraint and Coherence constraint.

4.2.1 Preservation Constraint

This constraint considers that, if a luminous point of the scene is visible at time t- 1, it is also visible at time t. Moreover, it assumes that the luminance of a pixel is invariant with respect to the motion, i.e. that any temporal modification of the luminance distribution over the pixels is directly attributable to the pixel's motion. Such a hypothesis is correct when the scene illumination is constant and uniform, and when the objects reflectance is Lambertian [39]. The preservation constraint has two classical formulations.

DFD-based formulation. The Displaced Frame Difference (DFD) expresses the difference between the luminance of the image at time t and the luminance of the image at time t + dt having undergone some displacement (dx; dy):

$$DFD(x, y; dx, dy; t) = I(x + dx; y + dy; t + dt) - I(x; y; t) \tag{4.1}$$

Where $I(x; y; t)$ is the luminance of image luminance at point (x,y) at time t and $I(x + dx; y + dy; t + dt)$ is the luminance of image at point $(x+dx, y+dy)$ at time t+dt. The preservation constraint consists in assuming that a motion vector $(dx; dy)$ that nullifies the DFD exists. If such a vector does not exist, the aim of the motion estimation is to determine the vector that minimises the DFD. The methods that use such a minimisation process are called correlation-based. Let u and v are local velocity ($u = \frac{dx}{dt}$ and $v = \frac{dy}{dt}$).

To compute the value of the DFD over a precise region, the criteria that are most frequently used are:

- the absolute value of the DFD over all region pixels with respect to local velocity [u v]

$$\sum_{region(i,j)} \left| DFD(i, j; u, v) \right| \tag{4.2}$$

  Where i and j are integer numbers (1,2,3,.....)

- the squared value of the DFD over all region pixels with respect to local velocity [u v]

$$\sum_{region(i,j)} \left(DFD(i, j; u, v) \right)^2 \tag{4.3}$$

  Where i and j are integer numbers (1,2,3,.....)

- Both these criteria can be divided by the total number of pixels taken into account, and are then respectively called the Mean Absolute Difference (MAD) and the Mean Square Error (MSE).

Differential formulation. If one considers the image function I continuous and possessing a derivative, a Taylor expansion (limited to the first order) provides:

$$I(x+dx, y + dy; t + dt) = I(x; y; t) + I_x(x; y; t)dx + I_y(x; y; t)dy + I_t(x; y; t)dt \qquad (4.4)$$

Where $I_x(x; y; t)$, $I_y(x; y; t)$ *and* $I_t(x; y; t)$ indicates the partial derivative of I with respect to *x*, *y* and *t* respectively. According to the rule of preservation constraint, it assume that the brightness of every points of moving or static object does not change in time. Because of this assumption:

$$I(x+dx, y + dy; t + dt) = I(x; y; t) \qquad (4.5)$$

Then

$$I_x(x; y; t)dx + I_y(x; y; t)dy + I_t(x; y; t)dt = 0 \qquad (4.6)$$

Devided by dt then

$$I_x(x; y; t)u + I_y(x; y; t)v + I_t(x; y; t) = 0 \qquad (4.7)$$

$$u = \frac{dx}{dt}, \; v = \frac{dy}{dt} \qquad (4.8)$$

Where *u* and *v* are local velocity. The optical flow equation only allows one to compute the component of motion in the direction of the spatial gradient and requires additional hypotheses to suppress all uncertainties.

4.2.2 Coherence Constraint

This constraint assumes the cohesion of all the elements of a unique object. It is valid if the motion variation between the neighbouring elements of an area is limited. It can be implicitly expressed in two different ways thanks to the neighbourhood information:

- Either with a region-based approach (all pixels of the region obey the same motion parameters);

- Either by the adoption of iterative or recursive solving methods that propagate the estimate of the neighbour pixels.

The coherence constraint can also be explicitly expressed when restrictions are formulated about the motion nature (a priori information), or when regularisation is ensured by smoothing criteria.

A remark should be made concerning the implicit formulation in terms of regions: on the one hand, segmentation is required in order to determine the various regions on which the coherence constraint should be applied. On the other hand, this segmentation should take the motion information into account so as to respect motion transitions. This problem arises as motion estimation requires segmentation, which requires motion estimation. Consequently, emerging techniques try to jointly solve the two problems.

4.3 Estimation Methods

Estimating methods aim to estimate the motion between successive pictures. The important objective behind this is to minimise a function that expresses some of the constraints presented in the last section. The minimisation can be achieved in several ways and can be grouped into three main approaches:

Differential methods: which are based on gradient measures. Direct differential methods aim at nullifying, the gradient of the function to be minimised while indirect differential methods converge towards a solution according to the gradient direction. Iterative and pixel recursive motion estimation algorithms are part of this class of methods.

Matching methods: are based on an explicit search for the best matching between two structures. The search for the best matching generally involves trying all the solutions of the search space. Block-Matching belongs to this class.

Stochastic methods: use random choices to drive the exploration of the parameters space. They include Bayesian estimation, Markov models and genetic algorithms.

4.4 Background Techniques for Motion Estimation

The approaches to motion estimation can be implemented in various ways. However the following background techniques for motion estimation are commonly used in the well known standards.

4.4.1 Linear Regression

Linear regression uses both the preservation constraint of equation (4.7) and a translational model of displacement for determined regions. The resolution over all the region pixels is achieved by a least square method, and provides the following solution:

$$(\hat{u},\hat{v}) = \arg\min_{(u,v)} \sum_{(i,j)\in R} (I_x(x,y,t)u + I_y(x,y,t)v + I_t)^2 \qquad (4.9)$$

Where $I_x(x; y; t)$, $I_y(x; y; t)$ *and* $I_t(x; y; t)$ design the partial derivative of I with respect to x, y and t respectively. The value of i and j are integer numbers in the region R. Usually, the difference between the two frames serves as temporal gradient and spatial gradients are digitally computed on the previous image. Small displacements can be measured with this method.

4.4.2 Iterative Motion Estimation

A gradient-based motion estimation was proposed in [40] so as to determine the optical flow. It was one of the very first methods established to solve equation (4.7). In order to correctly constrain the problem, Horn and Schunck have added an a priori smoothness condition on the resulting optical flow: the value of the gradient module had to be as small as possible. The problem then moves to a cost function minimisation, with the function expressed as:

$$\iint((I_x u + I_y v + I_t)^2 + \lambda(u_x^2 + u_y^2 + v_x^2 + v_y^2)^2)dx.dy \tag{4.10}$$

Where u_x, u_y, v_x and v_y are the partial first derivatives of the two optical flow (u; v) components, and λ is a (Lagrange) constant that balances the importance of the error in the motion equation and the penalty of departure from smoothness. A solution provided to this minimisation problem is:

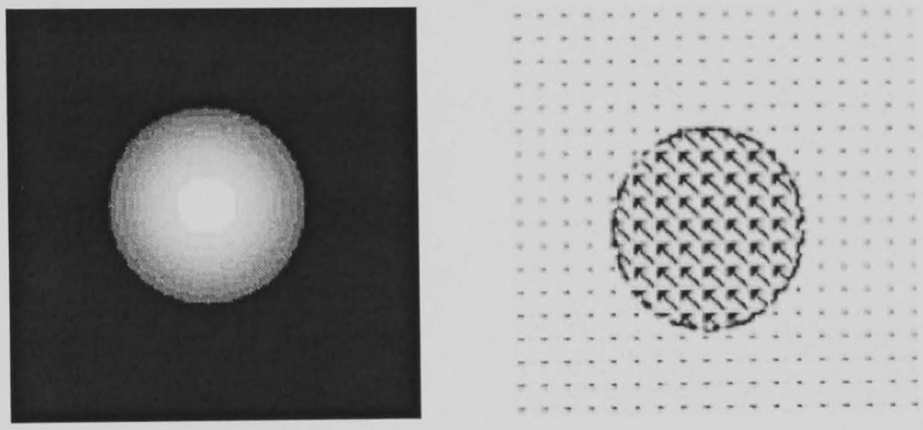
$$\hat{u} = u_m - I_x\left[\frac{I_x u_m + I_y v_m + I_t}{\lambda + I_x^2 + I_y^2}\right] \tag{4.11}$$

$$\hat{v} = v_m - I_y\left[\frac{I_x u_m + I_y v_m + I_t}{\lambda + I_x^2 + I_y^2}\right] \tag{4.12}$$

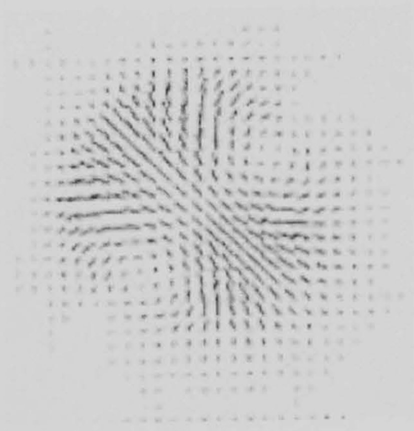
Where u_m and v_m are local average of u and v

Final determination of the optical flow can then be based on an iterative Gauss-Seidel method, refining $\hat{u}_i$ and $\hat{v}_i$ using $\hat{u}_{i-1}$ and $\hat{v}_{i-1}$ (i is the iteration number) until a certain convergence criterion is reached.

A sample of gradient-based motion estimation is as illustrated in figure (4.5). This technique provides a dense motion field (one motion vector for every pixel) the coding cost is very high. This is the major reason why it is practically never used for coding but for analysis instead.



(a) (b)



(c)

Figure (4.5) Limit of iterative motion determination: (a) Sphere at time t

(b) Applied motion field on I (t -1) (c) Detected optical flow

4.4.3 Pixel Recursive Algorithms

Pixel recursive [37] motion estimation algorithms estimate 2D motion recursively on a pixel basis: given an initial estimation for every point, $d_i = (u_i; v_i)$, a correction is carried out according to the resulting DFD:

$$d_{i+1} = d_i + \Delta d_i \tag{4.13}$$

with $\Delta d_i = (\delta u_i; \delta v_i)$ the update term of iteration i. The iteration can be executed along a scan line or from line to line or from frame to frame; the technique is then respectively denoted pixel recursive estimation with horizontal, vertical or temporal recursion. The basic assumption of this technique is that the DFD converges locally to zero when the estimated motion converges to the actual movement of the object point. The aim is thus to recursively minimise the squared value of the DFD using a steepest-descent (gradient) method:

$$d_{i+1} = d_i - \varepsilon DFD(x, y; u, v) \nabla d_i (DFD(x, y; u, v)) \tag{4.14}$$

Where ∇d_i is the gradient operator with respect to d_i and ε is a positive constant. Noting that

$$\nabla d_i (DFD(x, y; u, v)) = \nabla I \cdot [u \ v] \tag{4.15}$$

Where ∇I is the spatial image gradient, we obtain:

$$d_{i+1} = d_i - \varepsilon (DFD(x, y; u, v)) \nabla I \cdot [u \ v] \tag{4.16}$$

ε is a regulating parameter that achieves quick but sometimes oscillating convergence if it is high, or slow but accurate estimate if it is small. More advanced techniques use a variable ε to improve both the convergence speed and the solution accuracy. Evaluation of pixel recursive methods is very similar to the evaluation of iterative methods. In fact, the pixel recursive methodology has been applied to inter frame coding using the scheme of figure (4.6), which implies a high computation cost. In addition to the problem of properly computing the gradient, pixel recursive methods also estimate a motion field generally too smooth (like iterative methods).

4.5 The Block-Matching Algorithm (BMA)

Since its introduction by Jain and Jain in 1981, the Block Matching Algorithm (BMA) has emerged as the ME technique achieving the best compromise between complexity and quality: a fast estimation procedure allows obtaining a block-based motion field that is transmitted at low-cost. An appropriate choice of the block size offers one a compromise between adaptation to small moving objects (performed by small blocks) and robustness against noise (performed by large blocks). These properties have granted the BMA to be included in most video standards like H.263 [41], MPEG-1, 2 [42] and MPEG-4 [43].

4.5.1 BMA Principle

The principle of the BMA is to apply a translational motion model to sub-blocks of the image. For every block, the matching measure is based on the Displaced Frame Difference (DFD). The BMA in inter frame coding make the following assumption.

- Objects move in translation in a plane that is parallel to the camera plane, that is, the effect of camera zoom and object rotation are not considered.

- Illumination is spatially and temporally uniform, that is, the level of lighting is constant throughout the image and does not change over times.
- Occlusion of one object by another, and uncovered background are neglected.

The assumption neglects the problem of illumination changing over time which induces optical flow. The occlusion problem of the third assumption refers to the uncovered background problem. For the area of an uncovered background in the reference frame, no optical flows can be found in the reference frame. Although these assumptions do not always hold for the real world video sequence, they are still used by many motion estimation techniques.

The BMA scheme's source frame is segmented into distinct blocks and the Motion Vector of each block is obtained using block matching methods around a specified search window in the previous frame. The predicted frame is then obtained from the motion compensated previous frame of translated blocks. The sample of motion vector prediction is shown in figure (4.6).

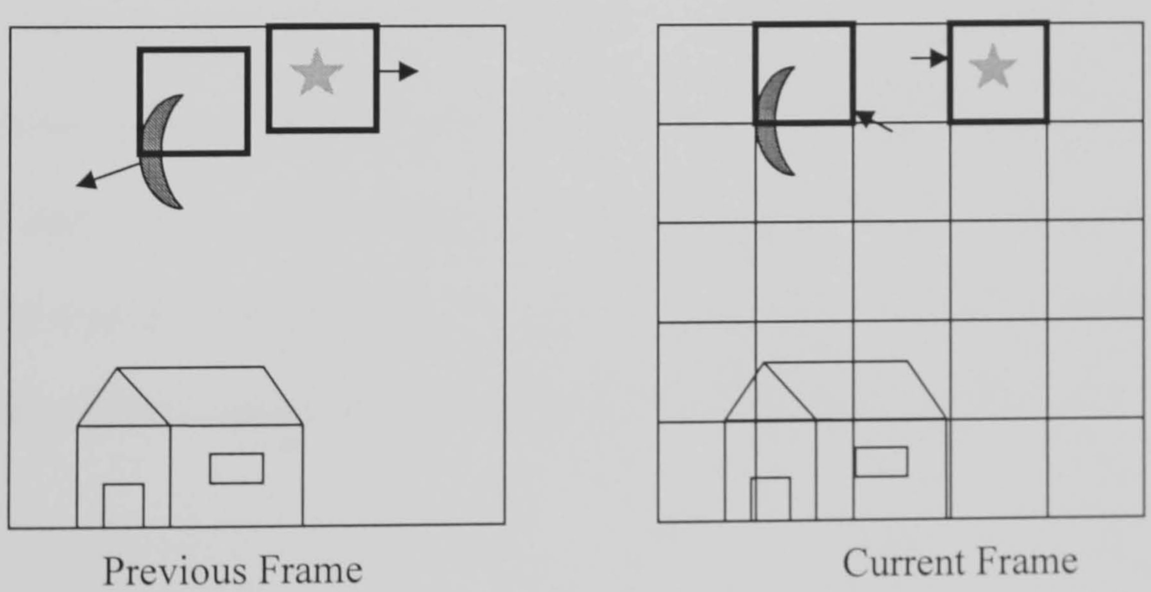


Figure (4.6) the example of Motion Vector Predication

From figure (4.6), the current frame is first divided into blocks. Each block will then search the previous frame to find the best matching block using a search algorithm with a matching criteria. After the best matching block is found the motion vectors can be determined. As shown in figure(4.6), the block which contains a star in the current frame can be also found in the previous frame by moving to the left.

4.5.2 Search Techniques

The location of the best match within the search area is based on an error criterion (e.g. the MAD requires intensive computation when it is performed using a FS method). Therefore, several complexity-reduced algorithms have been proposed for a faster search for the minimum DFD such as the three step search algorithm (3SS) and the new three step search algorithms (N3SS). The details of the search techniques will be explained in chapter 5.

4.5.3 Advanced Possibilities

Computation of dense motion field is based on the BMA. In this case, the minimisation of the error function is computed for each image pixel. In order to reduce the probability of false matches caused by noise, the matching criterion (e.g. the MSE) relies on a square window around the pixel. Sub-pixel accuracy is possible when the BMA is computed backwards. It means that blocks of image t are searched for in image t -1 with a step of half a pixel (which requires interpolation functions of course) or with lower fraction of a pixel. Such half-pixel accuracy is often used so as to compensate for the interpolation effects engendered by the small displacements of the camera.

4.6 Summary

In this chapter, the ill-posed problems were pointed out including aperture, correspondence, ambiguity and occlusion problems. These problems occur since the moving of the objects in video sequence is not the real movement. The 2-D motion information is actually resulted from the projection of real 3-D motion information on the 2-D plane. These problems are considered to be unsolved problems. Having understood these problems, the assumption such as preservation constraint and coherence constraint were introduced. These assumptions are behind the motion estimation techniques. In this chapter, the background techniques also were explained including optical flow, pixel recursive and block matching. The optical flow and pixel recursive are usually used for the analytic techniques and the block matching is used for the compression. The principle of the BMA is to apply a translational motion model to sub-blocks of the image. For every block, the matching measure is based on the matching criteria or distortion functions. The details of matching criteria or distortion functions will be explained later on in chapter 6.

Chapter 5

SEARCH ALGORITHMS IN MOTION ESTIMATION

5.1 Introduction

Video compression is vital for efficient storage and transmission of digital signal. The hybrid video coding techniques based on predictive and transform coding are adopted by many video coding standards such as ISO MPEG-1/2 and ITU-T H.261/263, owing to its high compression efficiency. Motion compensation is a predictive technique for exploiting the temporal redundancy between successive frames of video sequence. Block matching techniques are widely used in motion estimation method to obtain the motion compensated prediction. By splitting each frame into macroblocks, motion vectors of each macroblock are obtained by using block matching algorithms (or motion estimation algorithms). Block matching is a correlation technique that searches for the best match between the current image block and candidates in a confined area of the previous frame. The size of the block affects the performance of the motion estimation. Small block sizes afford good approximations to the natural object's boundaries; they also provide good approximation to real motion, which is approximated by a piecewise translation movement. However, small block size produce a large amount of raw motion information, which increases the number of transmission bits or the required data compression complexity to condense this motion information.

From a performance viewpoint, small blocks also suffer from the object (block) ambiguity problem and the random noise problem. Large blocks, on the other hand, may produce less accurate motion vectors since a large block may likely contain pixels moving at different speed and in different directions. In entertainment and teleconferencing typical picture size ranges from 240 lines by 352 pixels to 1080 lines by 1920 pixels (HDTV). Block sizes of 8×8 or 16×16 are generally considered adequate for these applications. The motion vector of each macroblock can be found by the most obvious and simplistic method, FS algorithm. All possible displacements in the search window are evaluated using block matching criteria (cost function). The block matching criteria has a direct impact on the performance of an algorithm, so it is necessary to choose a proper matching function in the process of searching for the optimal point. The advantage of FS is that we can find the absolute optimal solution. However, its high in computational complexity which makes it impossible for real-time implementation. Because the computational complexity of video compression, the compression efficiency and the compression quality is determined by the motion estimation algorithm, development of Fast Motion Estimation Algorithm for real-time application becomes an important research issue. The computational complexity of a motion estimation technique can then be determined by three factors:

1. search algorithm
2. Cost function/matching criteria
3. Search range parameter

To reduce the computational complexity, these factors have to be taken into account. Many approaches have been proposed in the research community. However, these approaches can be grouped into three main categories which are:

- those that reduce the candidate blocks for searching motion vectors, such as the three-step search, Cross Search and the fast full search;

- those that reduce the calculated pixels for computing the distortion measure, such as the normalised partial distortion search algorithm and subsampling block matching;
- those that reduce the current blocks for employing block matching, such as subblock matching.

Even though the FS method requires high computation, it is still a favourite for an application that needs to be very accurate such as those in medical imaging. However some applications such as mobile communication, where accuracy is not a serious problem, the fast search algorithms are the best choice.

In this chapter, the conventional full search algorithm and the fast search algorithms, Three Step Search (3SS), New Three Step Search (N3SS) and One At A Time (OTA) will be explained. Motion compensation using the OTA algorithm and the N3SS algorithm are compared with the 3SS algorithm and their performance are analysed.

5.2 Full Search Block-Based Motion Estimation

In this BMA, each block within a given search window is compared to the current chosen block and the best match is chosen. The best block matching is obtained by one of the block matching criterion or cost function. The search is carried out in the entire past frame. This method is considered to give the best video quality and ease of implementation. However, this method requires intensive computational complexity. In order to reduce the computational complexity of full search algorithm, other techniques such as sub-sampling, coarse quantisation, vector quantisation and transform coding are normally used. These approaches are helpful to upgrade the processing speed of FS. But the video quality degrades simultaneously.

If the block size is M×M pixels and the maximum displacement in the horizontal and vertical direction are W pixels then the search area will be of size (M+2W) (M+2W), as is depicted in Figure (5.1). With one pixel accuracy the search area will contain (2W+1) (2W+1) distinct, but overlapping blocks.

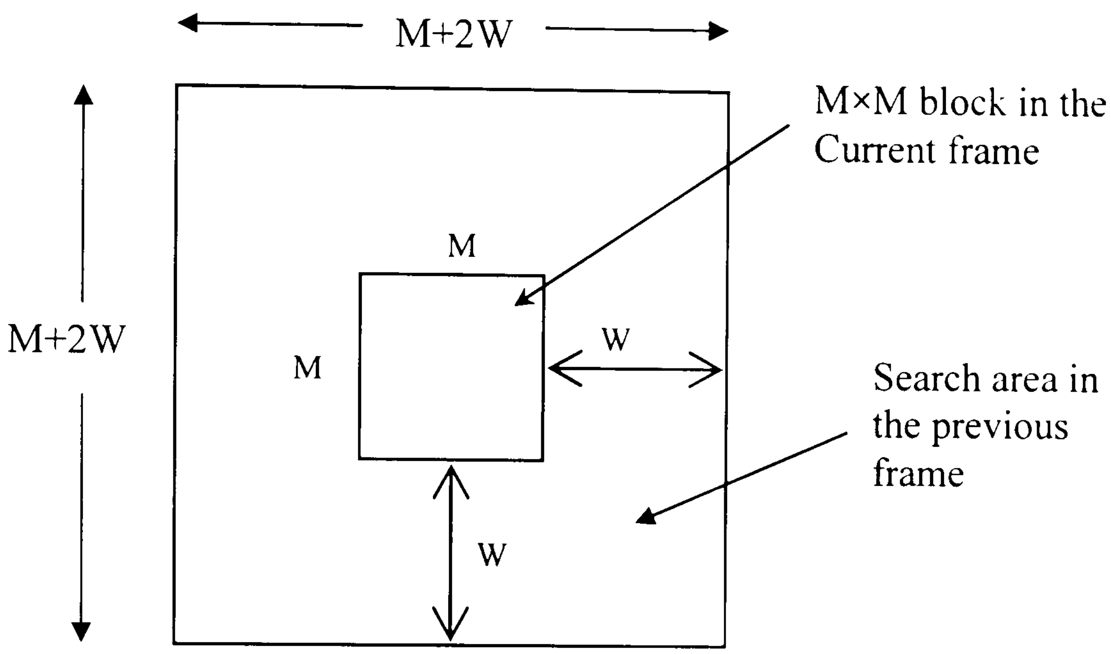


Figure (5.1) Block in the Current Frame and Search Area in the Previous Frame

It is clearly that the longer the allowable displacement, the greater the probability of finding a good match. The number of the candidate blocks n within the search area in the previous frame increases with the function: $n = 4W^2+4W+1$, as the displacement W increase. This can result in a very large number of candidate blocks being compared with the block in the current frame. The method of considering every candidate block in the search area as a potential match is known as the FS method.

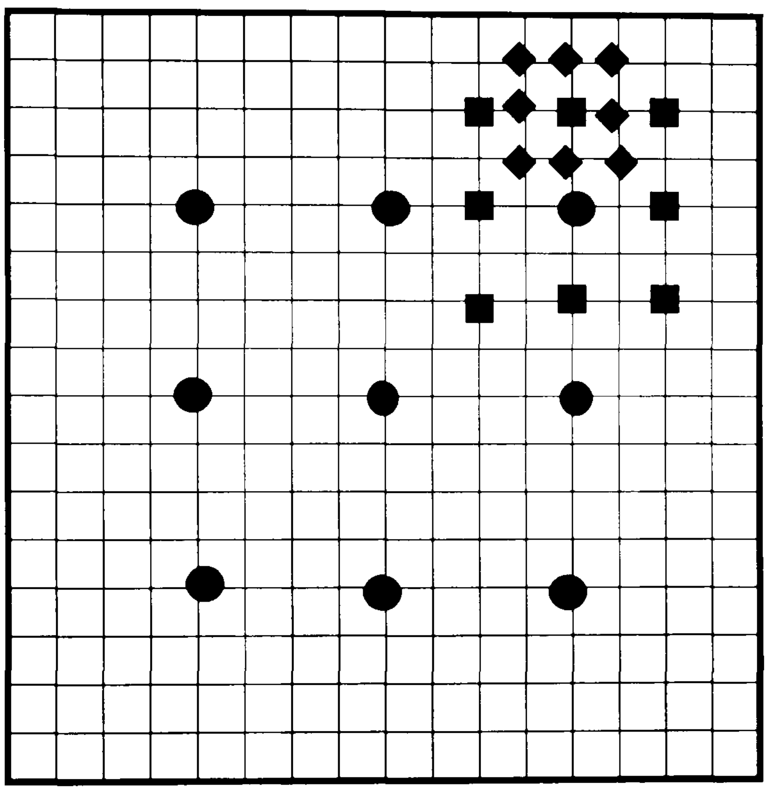
5.3 Fast Search Block-Based Motion Estimation

The computational complexity of the FS algorithm is too high for real time application. Because of this acute problem of complexity, several fast search algorithms have been developed [56-105]. The fast search algorithm aims to reduce the severe problem of the speed of operation. However, the disadvantage of the fast search algorithms lies in the quality of the reconstructed frame. Most of the fast search algorithms effect the performance of the reconstructed frame. Thus the good fast search algorithms have to be compromised between the speed of operation and the quality of the reconstructed frame. The first fast search was introduced in 1981 by Jain and Jain [45] and is known as the 2-D logarithmic search. Since then there have been a number of fast search algorithm that have been developed. Almost at the same time as Jain and Jain's development the three step search (3SS) was proposed by Koga [71]. The 3SS is slightly better in performance than the 2-D logarithmic. The 3SS is considered to be the landmark for the fast search algorithms. The basic operation of the three step search is the fundamentals of most of the new approaches available nowadays. The 3SS is well-known and relevant to MPEG standards.

5.3.1 Three Step Search Algorithm (3SS)

The basics of the three step search are very simple to understand and to implement. The three-step search is similar to the 2-D logarithmic. The major difference between 3SS and 2-D logarithmic are the use of matching criteria and the searching pattern. The 3SS is a fine-coarse search mechanism with logarithmic decreasing in step size is picked. Nine checking points are matched including the centre block and the eight blocks at a distance equals to each step size. The centre moves to the point with the minimum distortion and the step size is halved. The procedure is repeated until the final step.

which yields the motion vector. If the search region sr = 7, the number of the checking points required is (9+8+8) = 25. For larger search regions a selective procedure of the step size should be applied, in order for the three steps to remain. The 3SS is one of the most popular fast BMA and it is also recommended in H.261 owing to its simplicity and effectiveness. One problem that occurs with the 3SS is that it uses a uniformly allocated checking point pattern in the first step, which becomes in efficient when small ME is required.

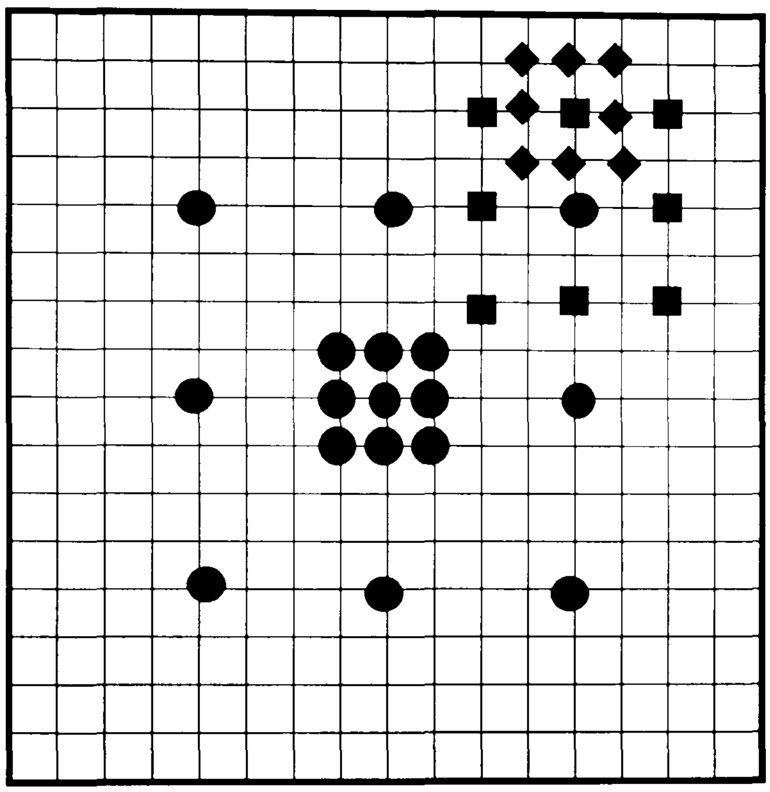


● Points for Stage1 ■ Points For Stage2 ◆ Points for Stage 3

Figure (5.2) Search Pattern of Three Step Search

5.3.2 New Three Step Search Algorithm (N3SS)

This algorithm aims to improve the performance of the conventional three step search. The N3SS algorithm was first introduced by R. Li, B. Zeng, and M.L. Liou in 1994 [72]. The fundamental of this algorithm are very similar to the 3SS algorithm. The difference is that it picks eight more blocks near the centre block for comparison as shown in the figure (5.3). The basic idea of this method is the fact that most small motion videos are centre biased. These eight more blocks are helpful for small motion estimation. Other stages are just the same as that of the 3SS. This method is considered to be very simple and robust. The quality of the reconstructed image is slightly better than 3SS.



● Points for Stage1 ■ Points For Stage2 ◆ Points for Stage 3

Figure (5.3) Search Pattern of New Three Step Search

However, the improvement in the image quality is not very significant as shown in figure (5.4)-(5.6). The N3SS algorithm is implemented and tested by using a Pentium 4 processor, 2.40 GHz.

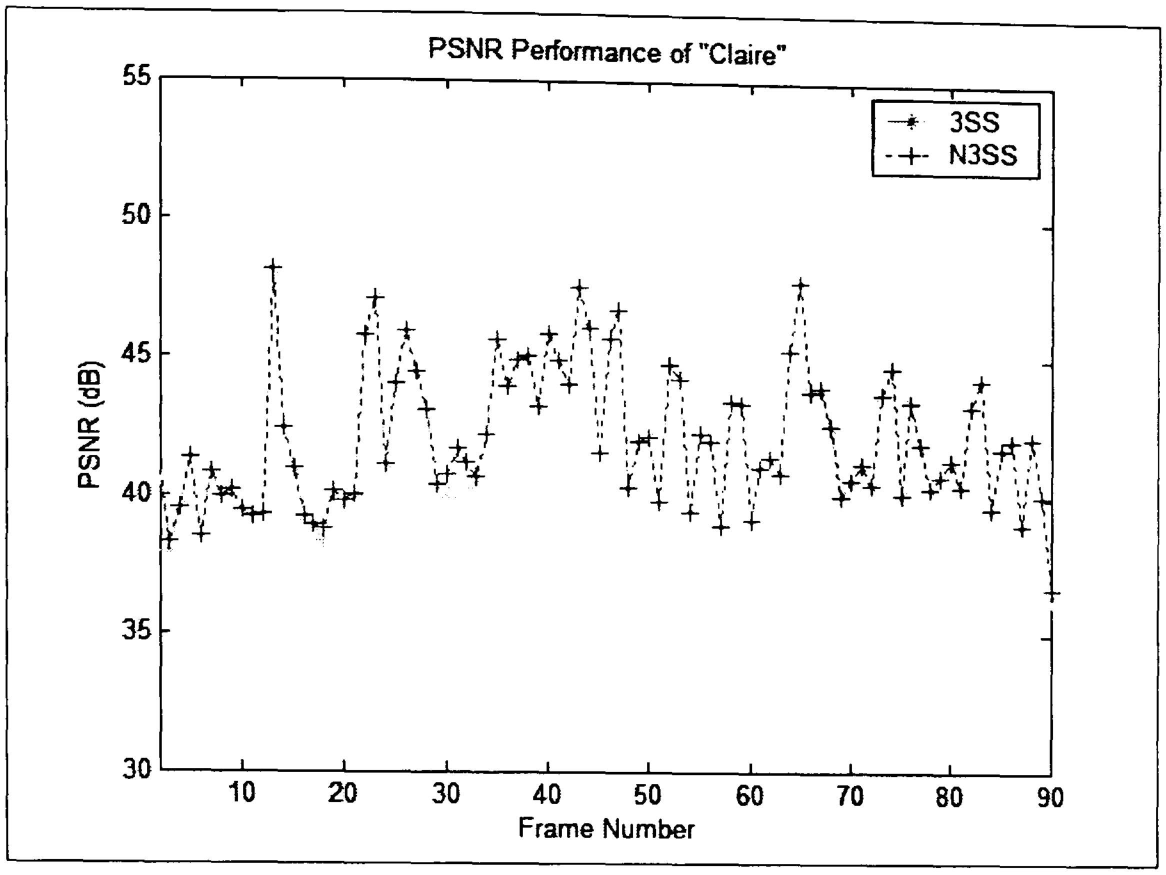


Figure (5.4) Comparative PSNR Performance between 3SS and N3SS using "Claire"

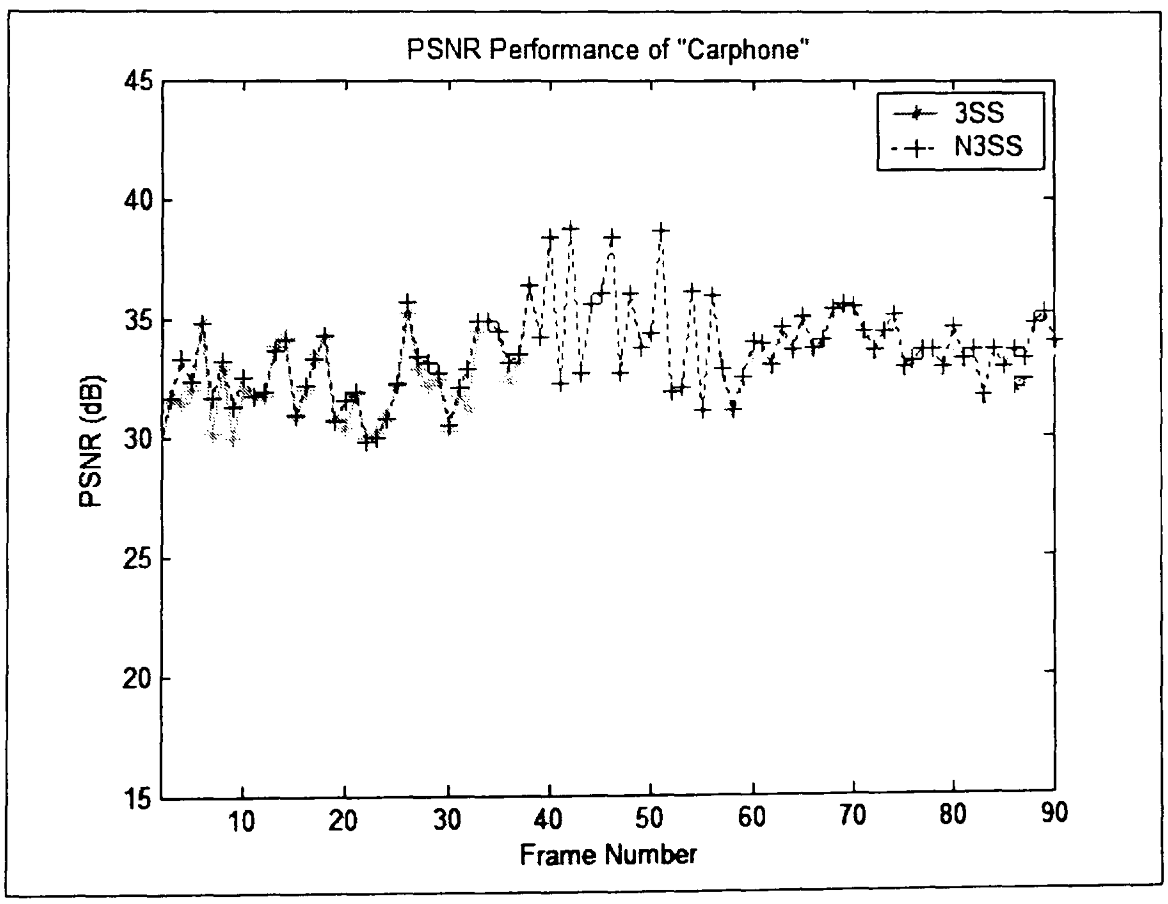


Figure (5.5) Comparative PSNR Performance between 3SS and N3SS using "Carphone"

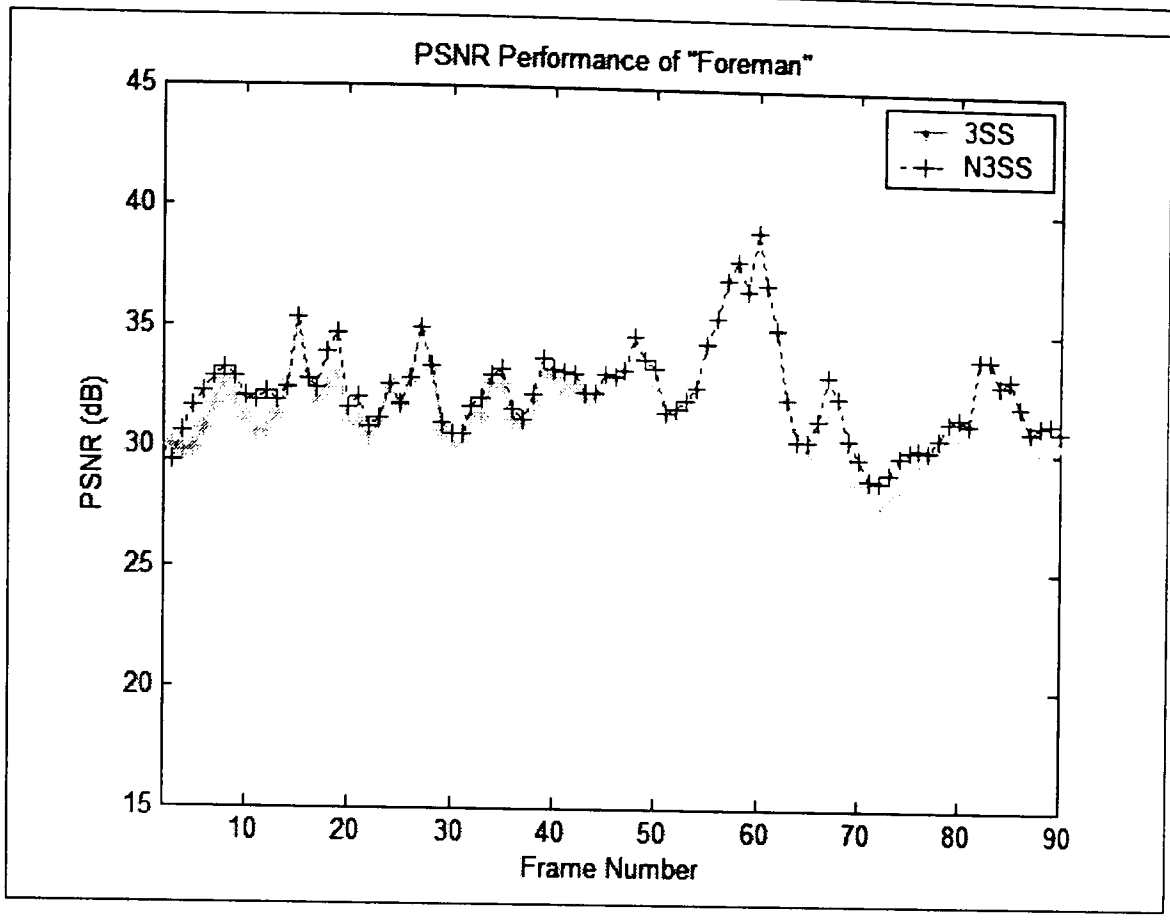


Figure (5.6) Comparative PSNR Performance between 3SS and N3SS using "Foreman"

The N3SS and 3SS algorithms are tested using the first 90 frames of "Claire", "Carphone" and "Foreman" video sequences. Each current frame is predicted from the previous frame. The average performance of the 3SS can be summarised in Table 5.1.

	Average PSNR (dB)		
	Claire	Carephone	Foreman
N3SS	41.65	33.16	32.02
3SS	41.62	32.82	31.53

Table 5.1 Average PSNR of N3SS and 3SS

The main disadvantage of this method is obvious because of the additional test of eight more blocks for first stage. The numbers of comparisons increase and therefore result in an increase in the time consumped [106].The average time taken by the N3SS and the 3SS are shown in the figure (5.7)

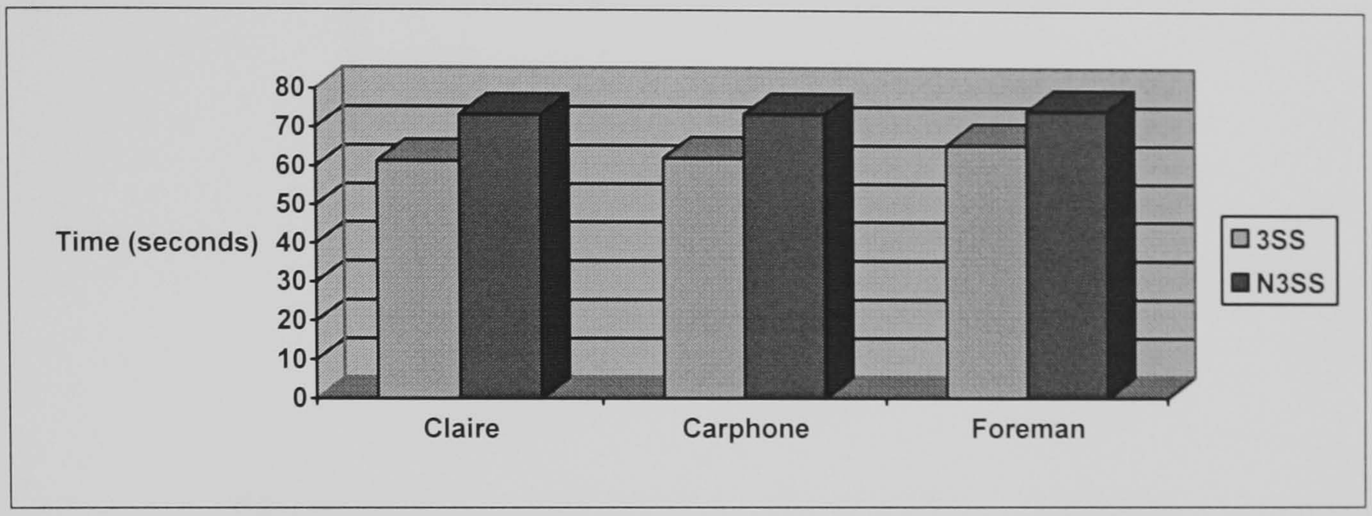


Figure (5.7) the Operation Time of 3SS and N3SS

Even though the quality of reconstruction improves, the complexity of algorithm increases dramatically. The procedure of motion estimation along 90 frames of video sequence requires approximately 63 seconds for 3SS and 73.72 seconds for the N3SS. The Average searching points are 25 and 33 for the 3SS and the N3SS respectively [106]. Therefore, taking everything into account, the 3SS is still favoured over the N3SS as it is widely adopted by the international standard [18][20][31].

5.3.3 One at a Time Algorithm (OTA):

This algorithm is another simple and efficient way to find the right location of corresponding block [73]. Initially, we pick three blocks horizontally at the centre of search area as shown in the figure (5.8) [109] and compare them with a particular block matching criterion. If the centre block has the minimum distortion compared with the previous frame, then we begin our search on the vertical direction.

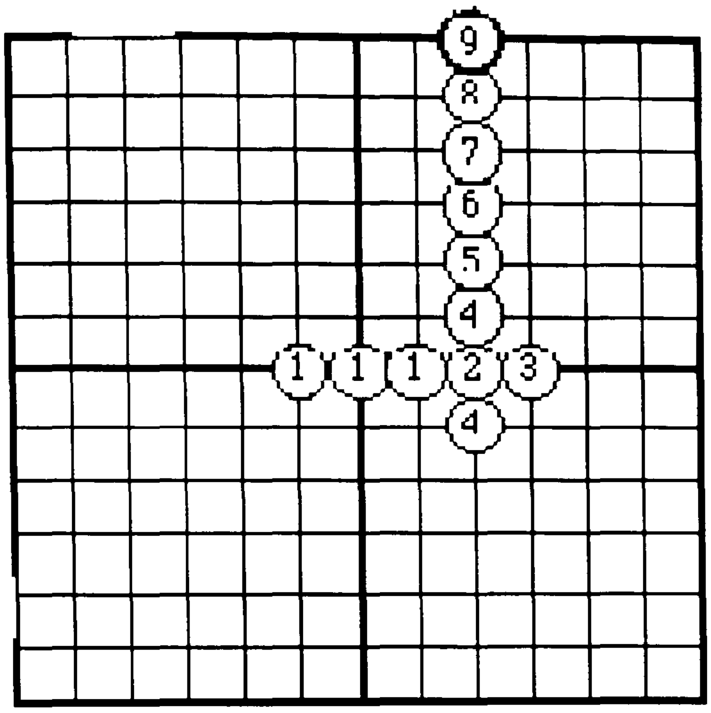


Figure (5.8) Search Pattern of One at a Time Algorithm

If the minimum distortion does not appear at the centre location, we continue looking in that direction till we find the block with minimum distortion. Then the search turns to the vertical direction. The direction of the search is the same as the way we search in the horizontal direction. Finally, the location of the right block and the motion vector are determined. This method is very simple in computation and the processing time is very reduced. However the quality of matching is not very good as shown in figure (5.9) [107][108]. From figure (5.9) the current frame is the second frame in a row of "Carphone" video sequence. The motion vectors are derived from the first frame of the sequence. The reconstructed frame is shown in the bottom left corner.

From the results shown in figure (5.9), the subjective quality of "Carphone" using the OTA is hard to see. The predicted frame is not good. The "Carphone" video sequence is considered to have slow movement. The prediction of the video frame is simple but the OTA cannot even achieve good reconstructed pictures.

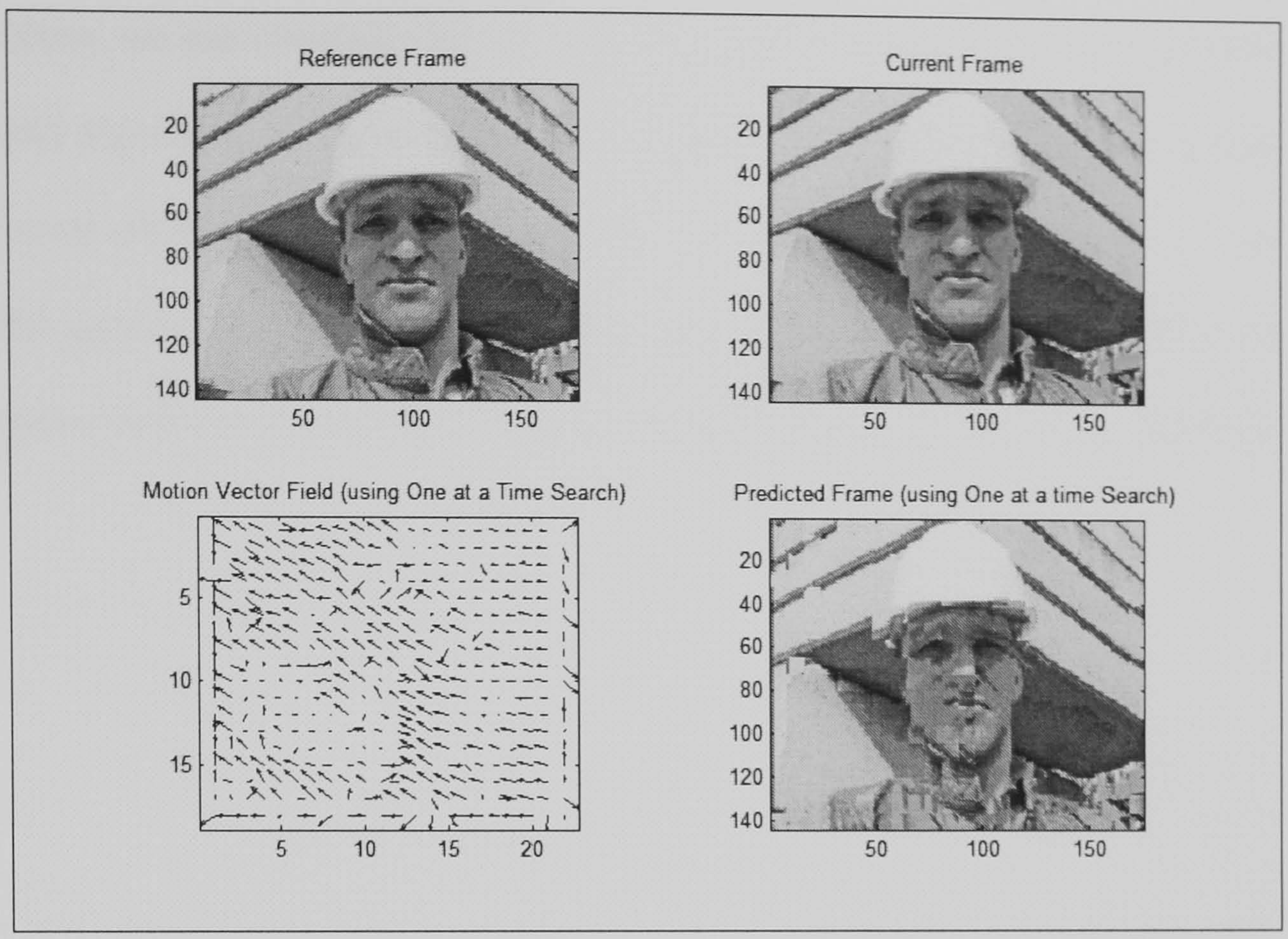


Figure (5.9) the Subjective Performance of One at a Time Algorithm Using "Carphone"

5.4 Summary

Four algorithms were discussed in this chapter: FS, 3SS, N3SS and OTA. These algorithms have their own characteristics and all have their own advantages and disadvantages. The Full-Search Algorithm achieves the best PSNR performance however, the computational complexity of the FS algorithm is extremely high. So this technique is not suitable for real-time applications. From the simulation results so far, the new three step search seems to be more efficient than the 3SS in terms of the reconstructed images. Nevertheless the complexity of N3SS is not as good as the 3SS. The OTA is another method which is proposed. The implementation of OTA is simple but the quality of the prediction is poor. The prediction is considered worse than 3SS and N3SS. Having considered the advantages and disadvantages of those four algorithms, we can conclude that the 3SS is considered as the best choice among them. The 3SS algorithm is robust and the implementation is simple. The 3SS algorithm is a good trade off between the quality of prediction and the speed of operation. Moreover the 3SS algorithm is still considered highly and is used for the well-known international standards. Most researchers use the 3SS as a benchmark to compare new algorithms.

Chapter 6

SEARCH ALGORTIHMS DESIGN AND IMPLEMENTAION

6.1 Introduction

To design the algorithms, many aspects that have impact on the performance of algorithm have to be carefully justified. One of the factors which affect the performance of the algorithms is the matching criteria. Thus the choice of matching criteria is one of the crucial factors which need to be considered. Some matching criteria require a greater computation time so these matching criteria will affect the speed of the algorithms. However, some high computational matching criteria achieve good matching result. The trade-off between the speed of operation and the quality has to be calibrated. The choice of matching criteria varies with the application. To implement the searching algorithm, the well-known benchmarks such as Foreman, Carphone and Claire are likely to be used for the simulation purpose. Each of these video sequences has their own characteristics. Every algorithm in this thesis is tested by using these well-known video sequences.

This chapter explains the methodology used to design and implement the block matching algorithms. To design the algorithms, simulation needs to be considered carefully. Several factors need to be taken into account. This chapter also explains the main factors that need to be considered when designing the algorithm.

The characteristics that need to be assessed and the performance of the algorithm are discussed. In addition, simulations of the well-known methods are given.

6.2 Simulation Environment for the Performance Evaluation

6.2.1Measurement Parameters

- **Predicted Image Quality**

The image quality is the main performance which every algorithm aims to achieve. The performance of search algorithms cannot be simulated without this quality assessment. The image quality assessment which is used in all of the algorithms in this thesis is Peak-Signal-Noise-Ratio (PSNR). In addition to PSNR, the subjective quality is taken into account. Although some of the reconstructed images cannot achieve a good PSNR, but the subjective quality is acceptable.

- **The Time of Operation**

This parameter is defined as the amount of time required to compress and decompress a picture for one image frame. Generally, the faster the compression/decompression can be performed, the better. Fast compression time increases the speed with which resulting compressed image can be created. Fast decompression time increases the speed with which the user can display and interact with the reconstructed images.

6.2.2 Sources of Video Sequence

There are 7 classical (benchmark) video sequences widely used for the simulation purposes, these are **Claire, Miss America, Salesman, Suzie, Carphone, Trevor and Foreman.** Each video sequence has its own characteristics. The characteristics of the video sequence are shown in the table 6.1.

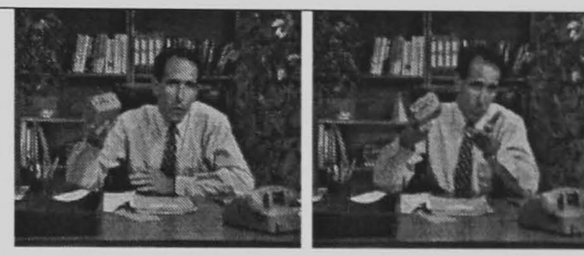
1. Claire	Similar characteristics with the sequence Miss America	
2. Miss America	Still camera, with slight movement of the person's head	
3. Salesman	The man sitting still at the table, with his flexible gesture kept changing.	
4. Suzie	A woman making a phone call, with a big movement from frame 50 to 70 (she puts down the headphone and swings her long hair), the headphone passing in and out of the camera lens	
5. Carphone	A little bit more complicated, with the car running, the counter-marching trees can be seen through the window. At the 55th frame, his mouth opens dramatically.	
6. Trevor	The whole image is split into 6 sub-ones, with their own separate behaviours. At the 61st frame, camera is suddenly switched to another speaker, who does not belong to the gang of former 6.	
7. Foreman (251-350)	Containing all the characteristics of the videos above, while from frame 251 to 250, with camera zooming and camera moving to an absolutely new scene	

Table 6.1 Characteristic of the Benchmarks Video Sequence

The video sequences can be divided into 3 main groups according to the nature of the movement.

(1) Quasi-stationary sequences with small movements: E.g. Claire, Miss America and Salesman. The person in the video is almost still, but with small movement on part of his/her body.

(2) Quasi-stationary sequences with objects moving in and out and relative bigger movements E.g. Suzie, Carphone and Trevor. Suzie has an object (headphone) passing in and out of the camera. Carphone has his mouth dramatically opened, and the camera is switched on suddenly in the sequence Trevor. Some objects are involved with big movement, either rushing into the scene as a new object or moving out.

(3) Quasi-stationary sequences with camera movement and camera zoom: E.g. Foreman (frame 251-340). Camera is held still until frame 250, and then it is slightly zoomed out and moved from the speaker to the wall. The final scene is totally different from the first 250 frames. There are, of course, objects moving in and out of the camera. For simulation purpose, only 3 video sequences which are taken from each group are adequate. The most favourite chosen video sequences in this thesis are Claire, Foreman and Carphone.

6.3 The Matching Criteria

The matching criteria are an essential part of the performance of the search algorithms. The choice of a matching criteria is crucial for the reason that the block matching sometimes requires the distortion function to be evaluated several thousands of times. Several matching criteria are proposed for the research algorithm so far such as Mean Square (MSE), Mean Absolute differences (MAD), Normalised cross-correlation function (NCF) and Number of Threshold Differences (NTD). The selection of the matching function has a direct impact on the computational complexity and the coding efficiency. Let m_x, m_y represent a motion vector candidate inside the search region and the m and n are the dimensions of the blocks. The value of i and j are integer numbers (1,2,3,....) and the f_n and f_{n-1} are the intensity values of the current frame and previous frames respectively.The Matching criteria can be found from the following equations:

i. **Normalised cross-correlation function (NCF):**

$$NCF_{(x,y)}(m_x,m_y)=\frac{\sum_{i=1}^{m}\sum_{j=1}^{n} f_n(x+i,y+j)f_{n-1}(x+i-m_x,y+j-m_y)}{\left[\sum_{i=1}^{m}\sum_{j=1}^{n} f_n^{\,2}(x+i,y+j)\right]^{1/2}\left[\sum_{i=1}^{m}\sum_{j=1}^{n} f_{n-1}^{2}(x+i-m_x,y+j-m_y)\right]^{1/2}} \tag{6.1}$$

ii. **Mean Absolute Differences(MAD):**

$$MAD_{(x,y)}(m_x,m_y)=\sum_{i=1}^{m}\sum_{j=1}^{n} \left|f_n(x+i,y+j)-f_{n-1}(x+i-m_x,y+j-m_y)\right| \tag{6.2}$$

iii. **Mean Square Error(MSE):**

$$MSE_{(x,y)}(m_x,m_y)=\sum_{i=1}^{m}\sum_{j=1}^{n}(f_n(x+i,y+j)-f_{n-1}(x+i-m_x,y+j-m_y))^2 \tag{6.3}$$

iv. **Number of Threshold Differences(NTD):**

$$NTD_{(x,y)}(m_x,m_y)=\sum_{i=1}^{m}\sum_{j=1}^{n}N(f_n(x+i,y+j),f_{n-1}(x+i-m_x,y+j-m_y)) \tag{6.4}$$

Where

$$N(f_n(x+i,y+j),f_{n-1}(x+i-m_x,y+j-m_y))=\begin{cases}1 & if\ \left|f_n(x+i,y+j)-f_{n-1}(x+j-m_x,y+j-m_y)\right|>T_0\\ 0 & if\ \left|f_n(x+i,y+j)-f_{n-1}(x+i-m_x,y+j-m_y)\right|\le T_0\end{cases}$$

is the counting function with threshold T_0

To estimate the motion vectors, we normally maximise the value of NCF and minimise the value of the other three matching criteria. NCF requires very high computation so the other matching functions are regarded as more practical, and they perform almost equally well for real images. NCF is the matching proposed to improve the performance of the PSNR. Nevertheless the MSE and MAD are likely to be used in most of the research environments because of their low computational complexity and simple implementation [73].

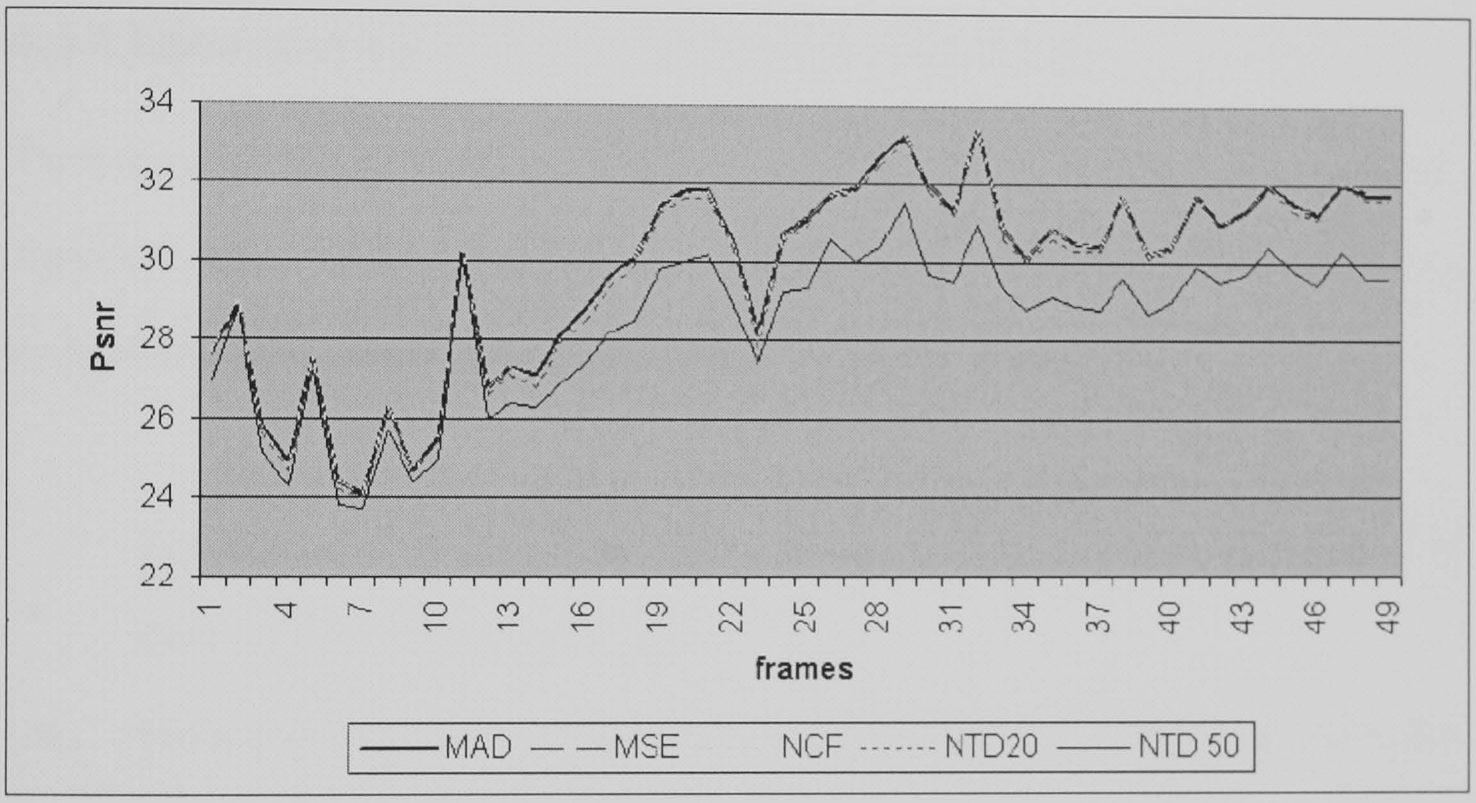


Figure (6.1) the Comparison of Matching Criteria

The performance of the matching criteria is compared in the figure 6.1. This test is simulated by using the Carphone video sequence. The first 50 frames of video sequence are simulated by using the MPEG encoder. This simulation aims to compare the performance of 3SS with the different matching criteria. The PSNR can be achieved by most of the criteria except NTD50. The PSNR of the predicted frame using MSE, MAD and NCF is almost the same. Therefore any one of these three matching criteria can be chosen. Nevertheless, if the consideration of the computational complexity is taken into account, the MAD is the best choice among those three matching criteria. The calculation of the MAD is simple as it is a straightforward mathematical operation. The MSE is slightly more complex than the MAD as it needs the square mathematical operation calculated on every pair of pixels in the search area. The NCF is the worst matching criteria among them because the computation is very complicated as it needs several mathematical operations.

6.4 Characteristics of a good search algorithm

There are a set of requirements for a good search algorithm, even though any realistic algorithm cannot achieve all these goals simultaneously. From this set of requirements some will have to be inadequately fulfilled so as to enhance the others.

i. Convergence – the search algorithm should converge to the optimum point in finite steps for a finite search region.

ii. Fewer search points – the total number of search points necessary for finding the optimum point should be as few as possible. This measure could be evaluated either in the worst case or in the average case.

iii. Fewer search steps – the total number of search steps necessary for finding the optimum point should be as few as possible.

iv. Noise immunity – the search algorithm should converge independently of the noise in the data.

6.5 Simulation of the well-known search algorithms

The simulation of the well-known search algorithms have been tested and implemented on the Pentium4, 2.4GHz computer. The algorithms are created by using MATLAB. The metric time and Peak Signal to Noise Ratio (PSNR) are used to evaluate the algorithms. The subjective quality of prediction is also assessed to support the performance. The subjective quality is used to justify if the prediction is perceptible or not. The benchmarks video sequences are used for the simulation purpose. The condition of the simulation can be summarised in the table 6.2

Maximum Displacement	7 (-7 ~ 0 ~ +7)
Cost Function	MAD
Block Size	8 by 8
Source Video Sequence	QCIF 176 * 144 video sequence
Performance assessment	PSNR and Operation Times

Table 6.2 Summarisation of the condition of simulation

The procedure starts from the video sequence are fed into the system. The video sequence is in the qcif format file which each file consists of the three main components, YUV. Nevertheless only Y component or luminance component is taken into consideration because the motion estimation fundamentally uses only intensity value of pixels to calculate the matching criteria or distortion functions. The video sequence will be extracted into single frames. These frames will be further passed to the block matching process. The steps of block motion estimation can be summarised as below:

Step1: The video sequence is divided into single frames. The two consecutive frames are fed into the block base motion estimation scheme.

Step2: The first frame came in is considered as the past frame and the second frame came in is considered as the current frame.

Step3: The current frame segmented into perfectly tiling blocks and the block size is 8×8 pixels.

Step4: Each targeted block of the current frame is compared with the blocks in the past frame to find the best matching block. The matching blocks must be as similar as the targeted block. The matching criteria or distortion function is used to quantify the similarity between the target block and candidate block.

Step5: The search algorithm is used to design a pattern how the targeted block is compared with the candidate blocks. There are several search algorithms have been proposed. The search algorithms will be explained in details later in this chapter.

Step6: After the best matching block is found. The motion vector can be also found by considering the best matching block compared with the target block. Motion vector will give information about the translational model of the best matching block.

Step7: The current frame is predicted by using the intensity value of the block from the previous frame with the translational model.

The most important part of the motion compensation is the search algorithm. The search algorithm must efficiently find the best matching blocks. The simulation of the algorithms in this research followed steps as described above. The main difference of each algorithm is search pattern in step5. The process can be viewed in the diagram in figure (6.2).

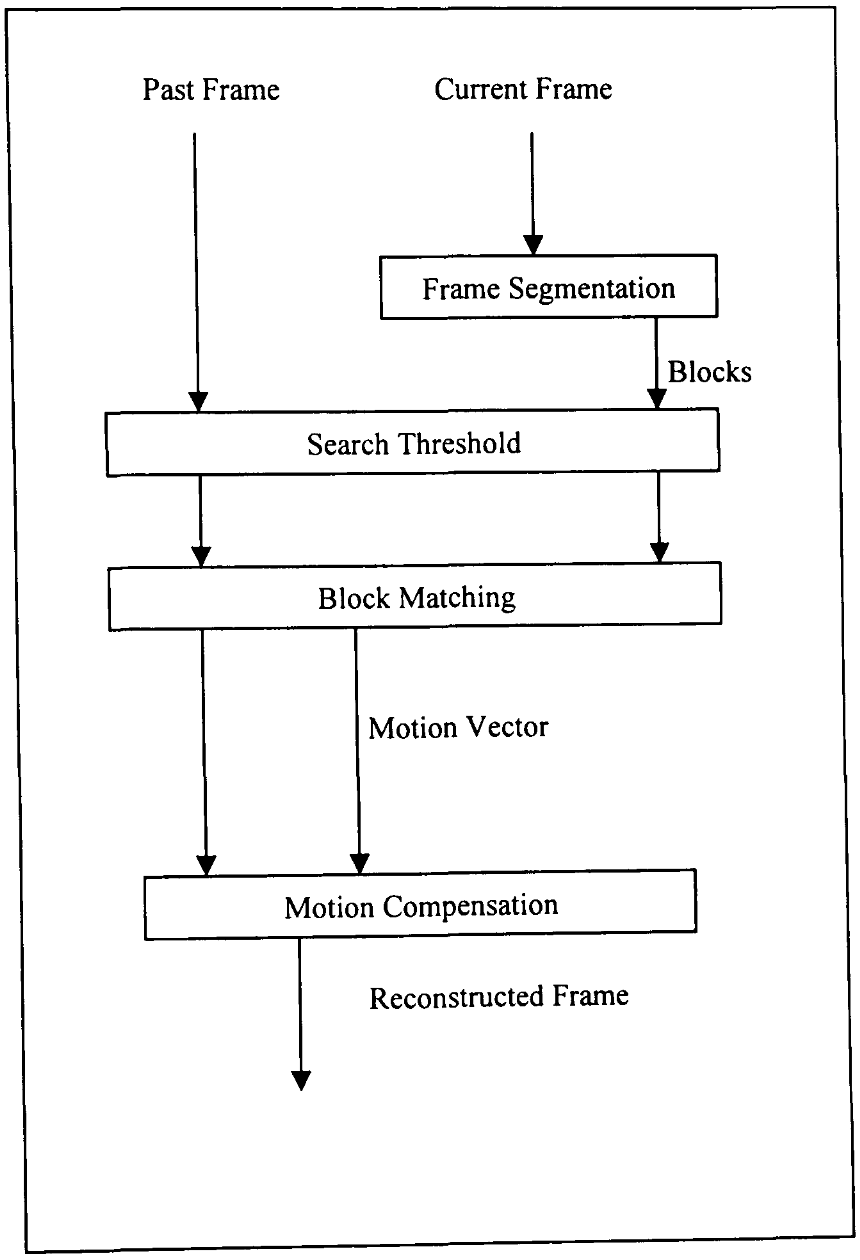


Figure (6.2) Motion Compensation Process

6.5.1 FS Algorithm

FS algorithm is considered to give the best performance in term of the quality of reconstructed image. However, the time of the operation of this method is extremely high. The Sample of Motion Estimation using FS Algorithm is shown in the figure (6.2)

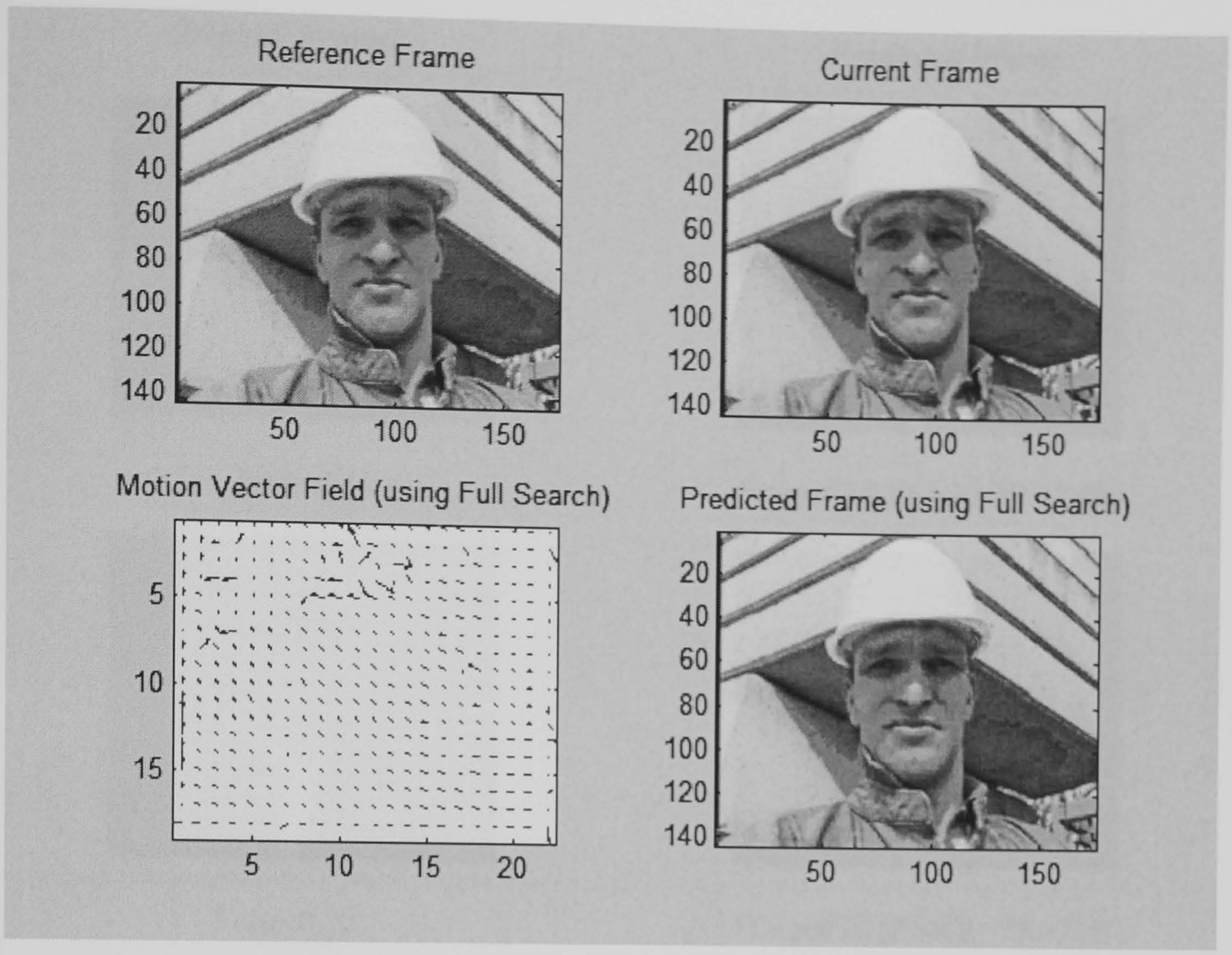


Figure (6.3) The Sample of Motion Estimation using FS Algorithm

As shown in figure (6.3), the two upper images are the third and the fourth original frames of the video sequence. The lower left diagram shows the motion vectors between the two original images. The lower image is the prediction of the fourth frame by using ME algorithm. The prediction of the fourth frame is derived from the third original frame by adding the information of the motion vectors. The arrows in the lower left diagram show the direction where each block of the fourth frame is moving from. If there is no movement of the block between the frames, dots will occur instead of arrow. From figure (6.3), the prediction of the forth frame is as good as the original frame in term of subjective quality.

Test sequence: "Claire"

Original Frame *Predicted Frame*

Frame#10

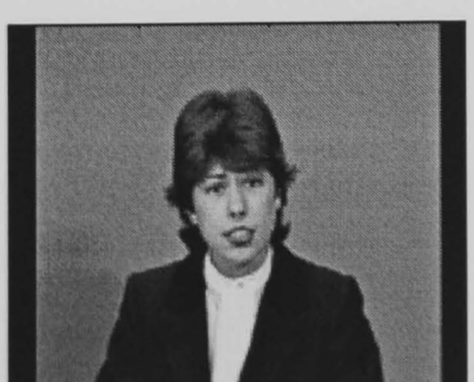
Frame#10(PSNR=39.37dB)

Frame#20

Frame#20 (PSNR=39.97dB)

Frame#30

Frame#30 (PSNR=40.74 dB)



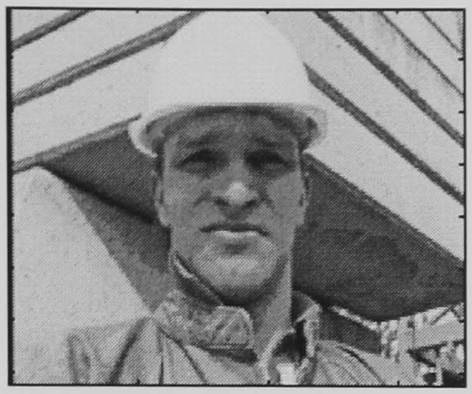
Frame#40

Frame#40(PSNR=45.89dB)

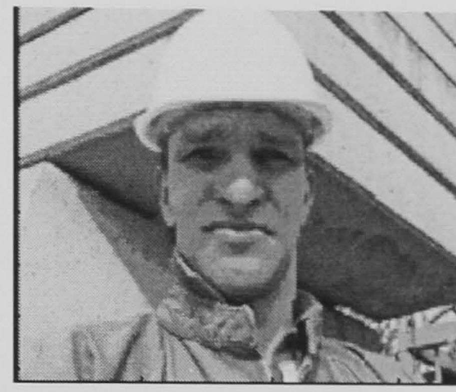
Figure (6.4) "Claire" Subjective quality of the predicted frame using FS

Test sequence: "Foreman"

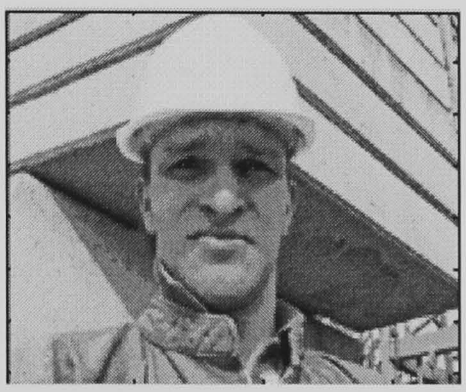
Original Frame *Predicted Frame*



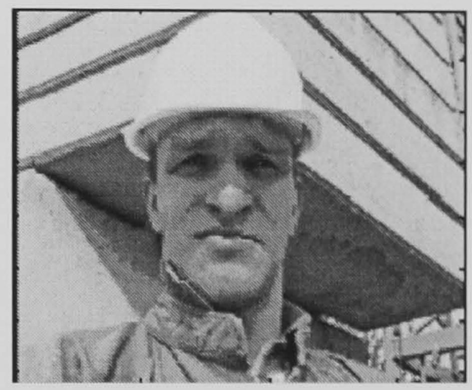
Frame#10



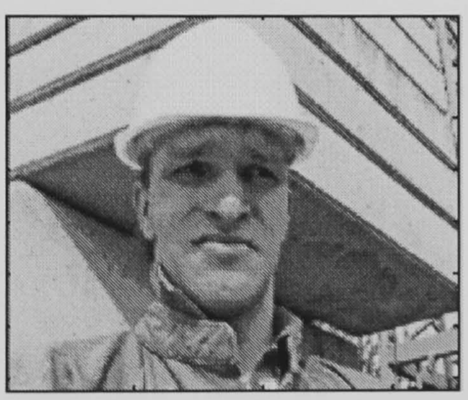
Frame#10(PSNR=32.10dB)



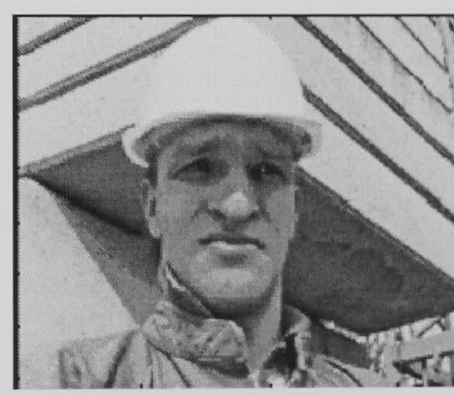
Frame#20



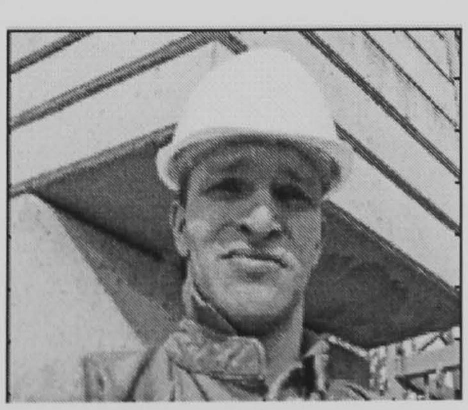
Frame#20 (PSNR=30.84dB)



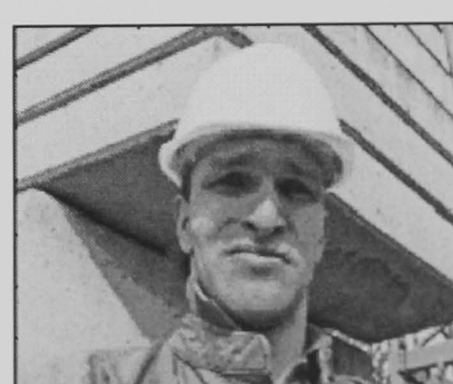
Frame#30



Frame#30 (PSNR=31.39 dB)



Frame#40



Frame#40(PSNR=33.83dB)

Figure (6.5) "Foreman" Subjective quality of the predicted frame using FS

Test sequence: "Carphone"

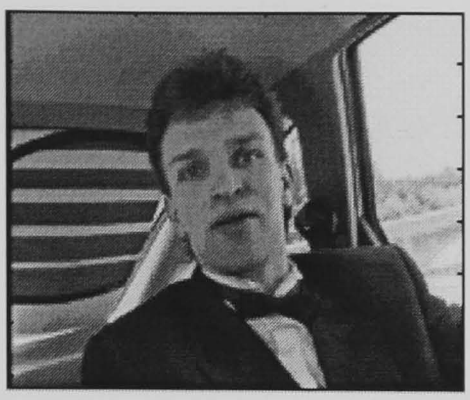
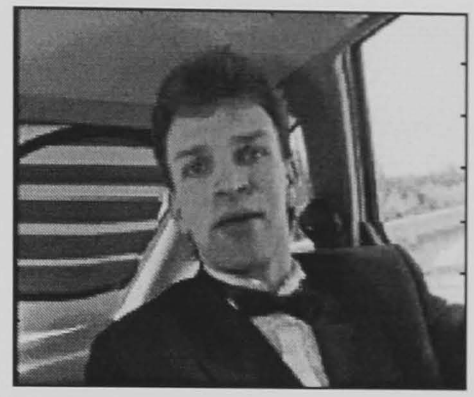
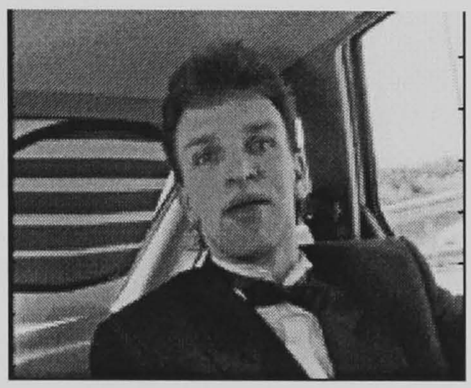
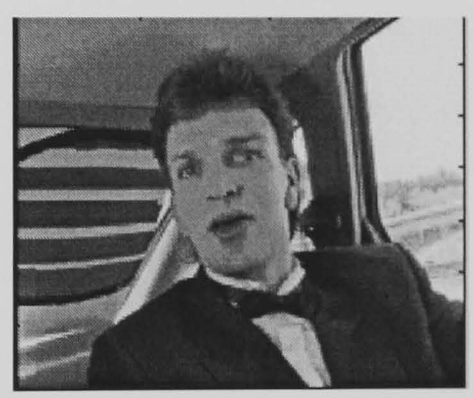
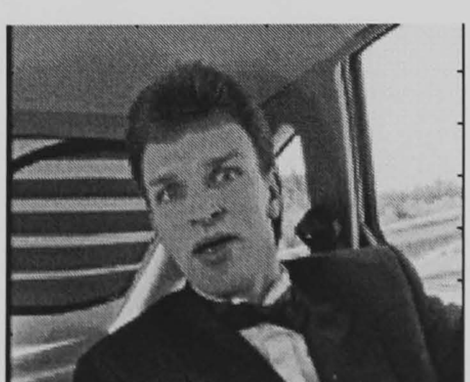
Original Frame	*Predicted Frame*
	
Frame#10	Frame#10(PSNR=32.89dB)
	
Frame#20	Frame#20 (PSNR=31.93dB)
	
Frame#30	Frame#30 (PSNR=30.69 dB)
	
Frame#40	Frame#40(PSNR=38.41dB)

Figure (6.6) "Carphone" Subjective quality of the predicted frame using FS

Having implemented the FS algorithm, the subjective quality of the prediction is assessed as well as object quality. The performances of the FS algorithm in terms of subjective quality are shown in figure (6.4)-(6.6). Figure (6.4)-(6.6) are the results of the predicted frame using the FS algorithm. The frames on the left hand side are the original frame of the 10^{th}, 20th, 30th, and 40^{th} frame in a row. Each original frame is compared with the previous frames which are the 9^{th}, 19th, 29th and 39^{th}. The motion vectors are derived by a comparison between those two consecutive frames using the MAD matching criteria. The simulation is performed in the same way as that of figure (6.3). The subjective quality of the predicted frame is shown in the right hand side of the figures. The value of PSNR is also shown in the figures. Figure (6.4) shows that the FS algorithm can achieve both the subjective quality and objective quality for most of the case in the Claire video sequence. The PSNR is higher than 39 dB. The FS algorithms can also achieve the subjective quality of the Carphone and Foreman video sequence as shown in figure (6.5) and (6.6).

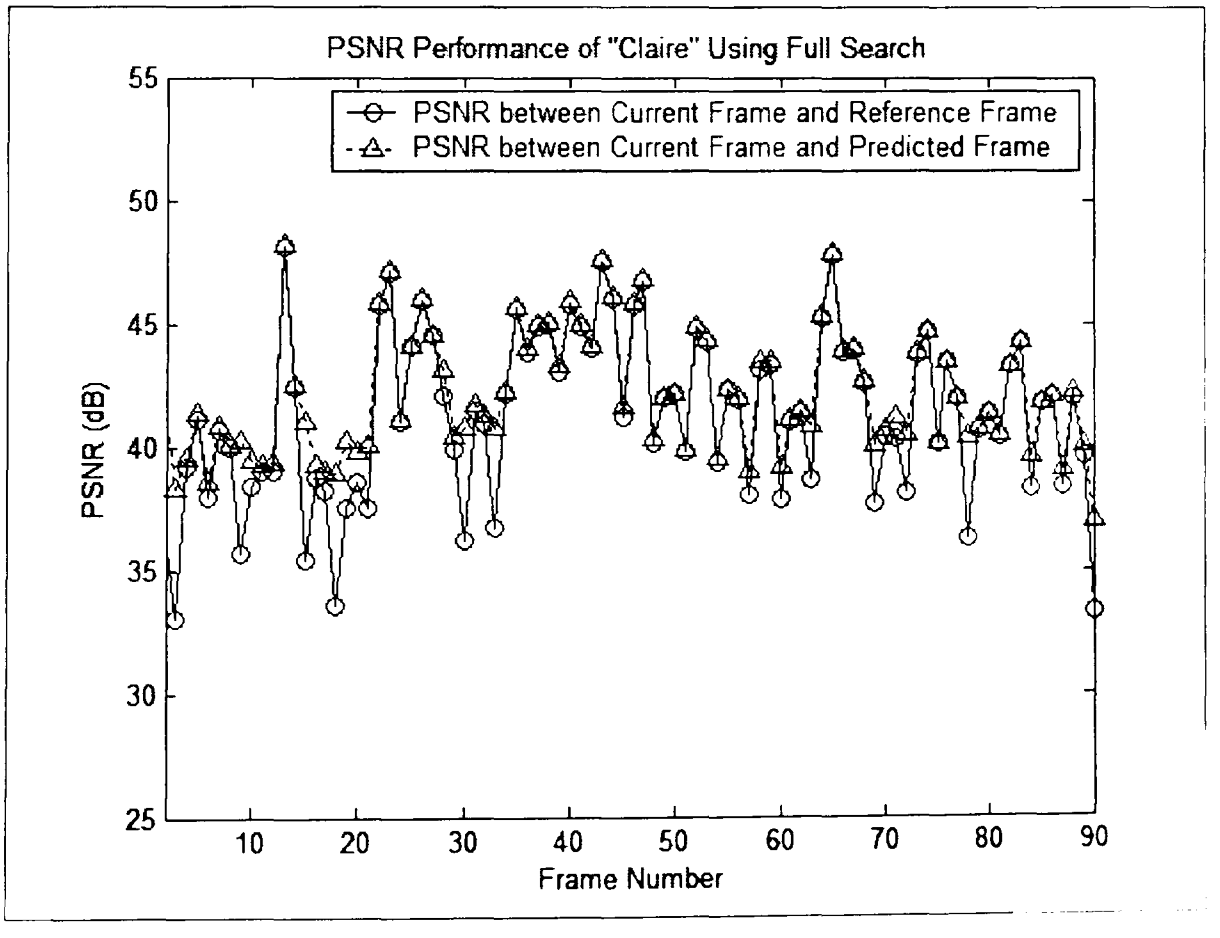


Figure (6.7) The PSNR Performance of "Claire" over 90 frames using FS

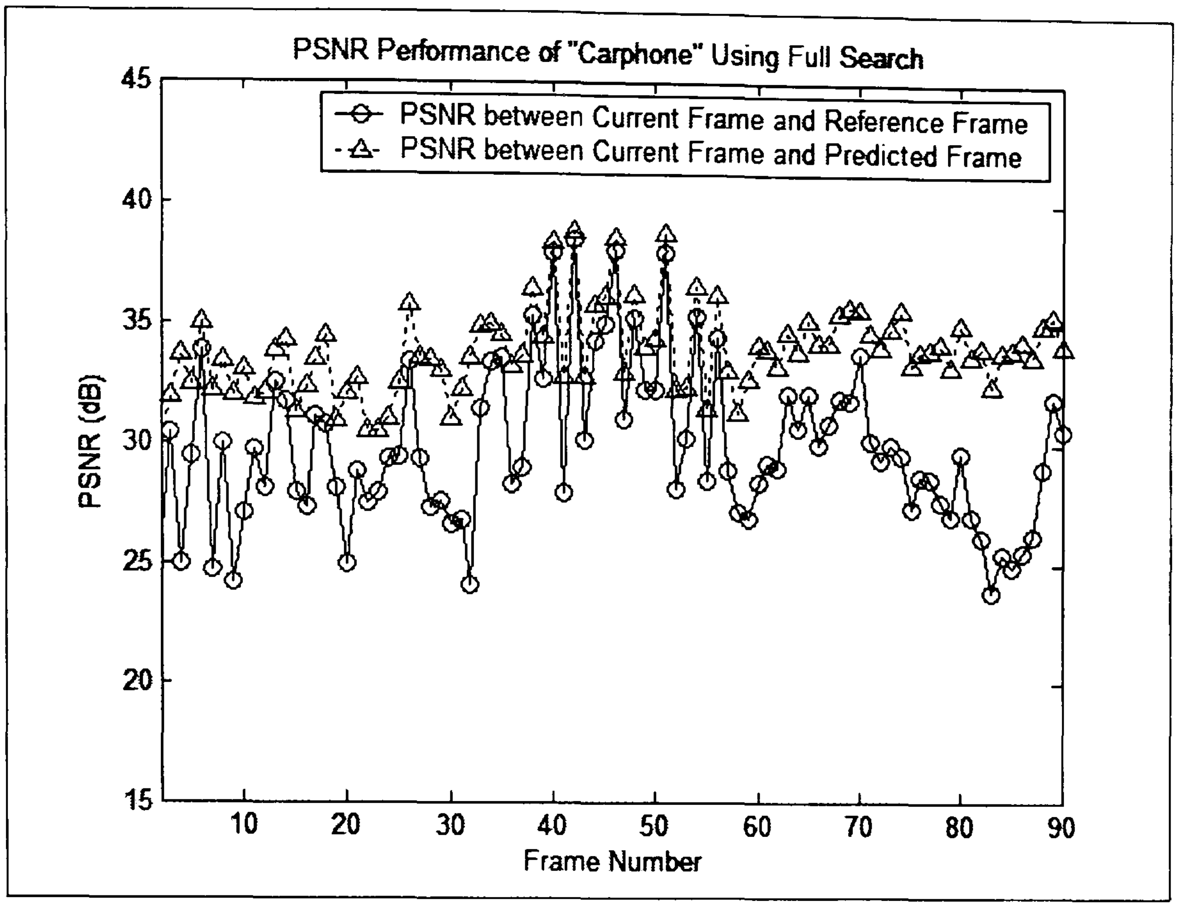


Figure (6.8) The PSNR Performance of "Carphone" over 90 frames using FS

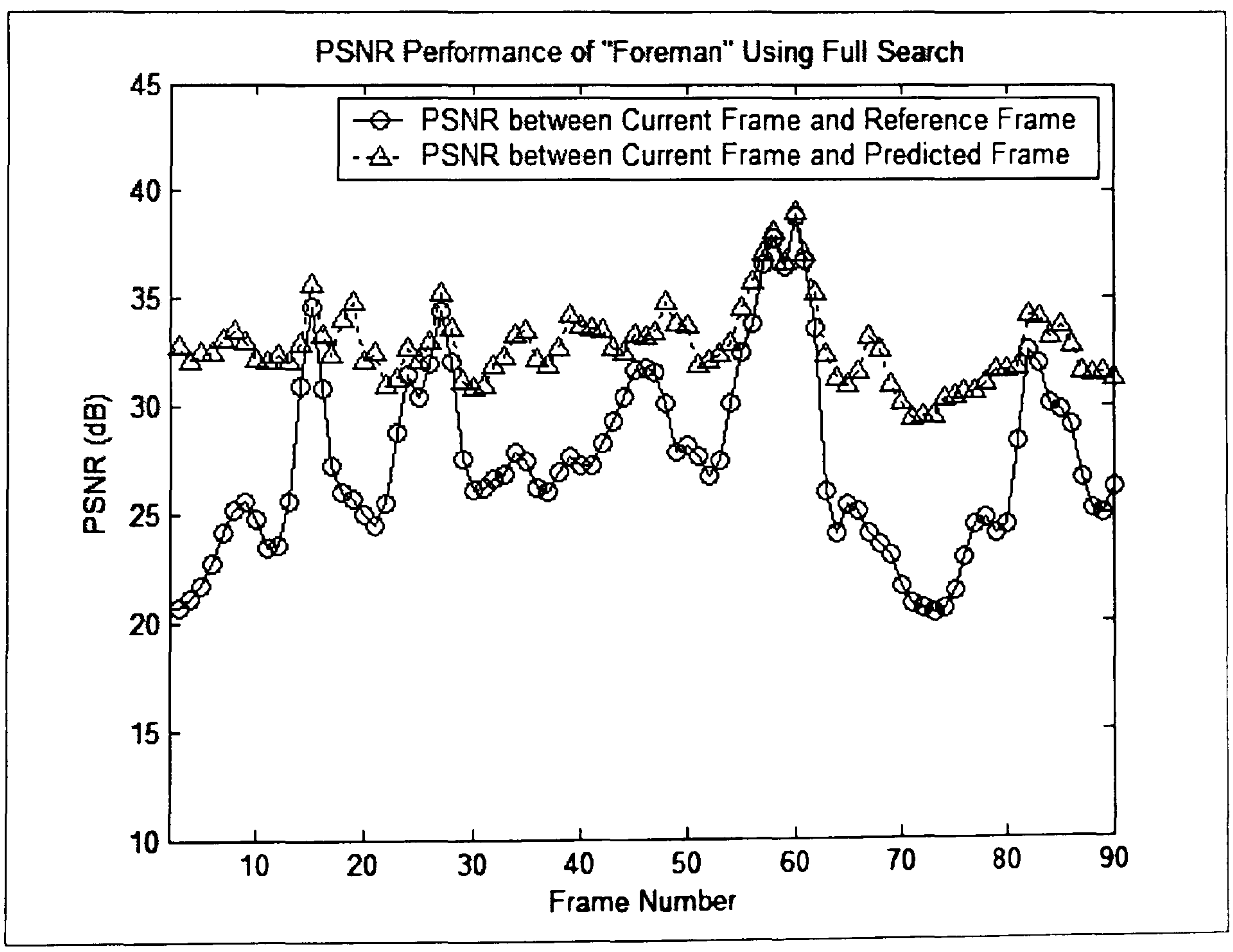


Figure (6.9) The PSNR Performance of "Foreman" over 90 frames using FS

The PSNR performance is simulated as it is an objective quality assessment. There are two sets of PSNR simulated. One is the PSNR1 that is the difference between the current frame and the reference frame. The value of this PSNR aims to measure the difference between the two original consecutive frames and to use it as the reference for the quality assessment of the predicted frame. The second one is the PSNR2 which is the difference between the current and predicted frame. The PSNR2 is the PSNR which show the quality of predicted frame. The best motion estimation should be achieved with as high PSNR as possible. The simulations of FS algorithms are tested using Claire and the results are shown in figure (6.7). The first 90 frames of Claire video sequence are tested. Every two consecutive frame are compared against these 90 frames. As shown in figure (6.7), the FS algorithm can achieve very good prediction as the PSNR2 values are very high. The range of PSNR2 is between 32 to 46 dB. As shown in figure (6.8), the FS algorithm is simulated using the Carphone video sequence. From the figure (6.8), the PSNR1 is low because the movement of the object in the video sequence thus the difference between two consecutive is higher than Claire. Having used the Carphone video sequence as the tested sequence, the FS algorithm is still able to achieve the quality of the prediction as the overall value of PSNR2 is high as it is greater than 30 dB in every predicted frame. The FS algorithm is also tested by using Carphone video sequence. The Carphone video sequence is considered to be a greater movement than the Claire and Foreman video sequence as the value of PSNR1 is much lower than that of FS algorithm using Claire and Foreman video sequence. The FS algorithm can achieve the high value of PSNR2 as it is greater than the 30 dB. From the simulation of FS algorithm using the 3 different video sequence as shown in figure (6.4)-(6.9), the results show that the performance of FS algorithm is very good at the prediction both in terms of subjective and objective quality.

6.5.2 3SS Algorithm

The 3SS algorithm is the technique that is widely used by the well-known standards. There are several search algorithms introduced so far. However the 3SS still remains a favourite choice for the motion estimation. The 3SS algorithm can be implemented and completed in 3 steps. At this search, nine points are searched at each step. The 3SS is simple and robust. The implementation of the algorithm is straight forward. The procedure of the algorithm can be summarised as below:

Step1: The search is initialised at the centre of search windows with position (0,0). The search window's size is 15×15. The eight points around the centre of search windows are tested. For the centre (0,0) and step size is 4, the position *(-4,-4), (-4,0), (-4,4), (0,-4), (0,0), (0,4), (4,-4), (4,0) and (4,4)* are examined. The MAD is used as the matching criteria. The MAD used to evaluate the similarity between the target block with the candidate blocks at those position in the previous frame. The position with the minimum MAD becomes the centre of the next step.

Step2: The steps size reduced by half and the centre is updated. The eight points around the updated centre are examined. For the initial step size is equal to 4, the step size will be 2 at this step. The position with the minimum MAD becomes the centre of the next step.

Step3: The steps size reduced by half, so the step size becomes 1. Again the eight points around the updated centre are tested with the step size 1. The position with minimum MAD will be the best matching block.

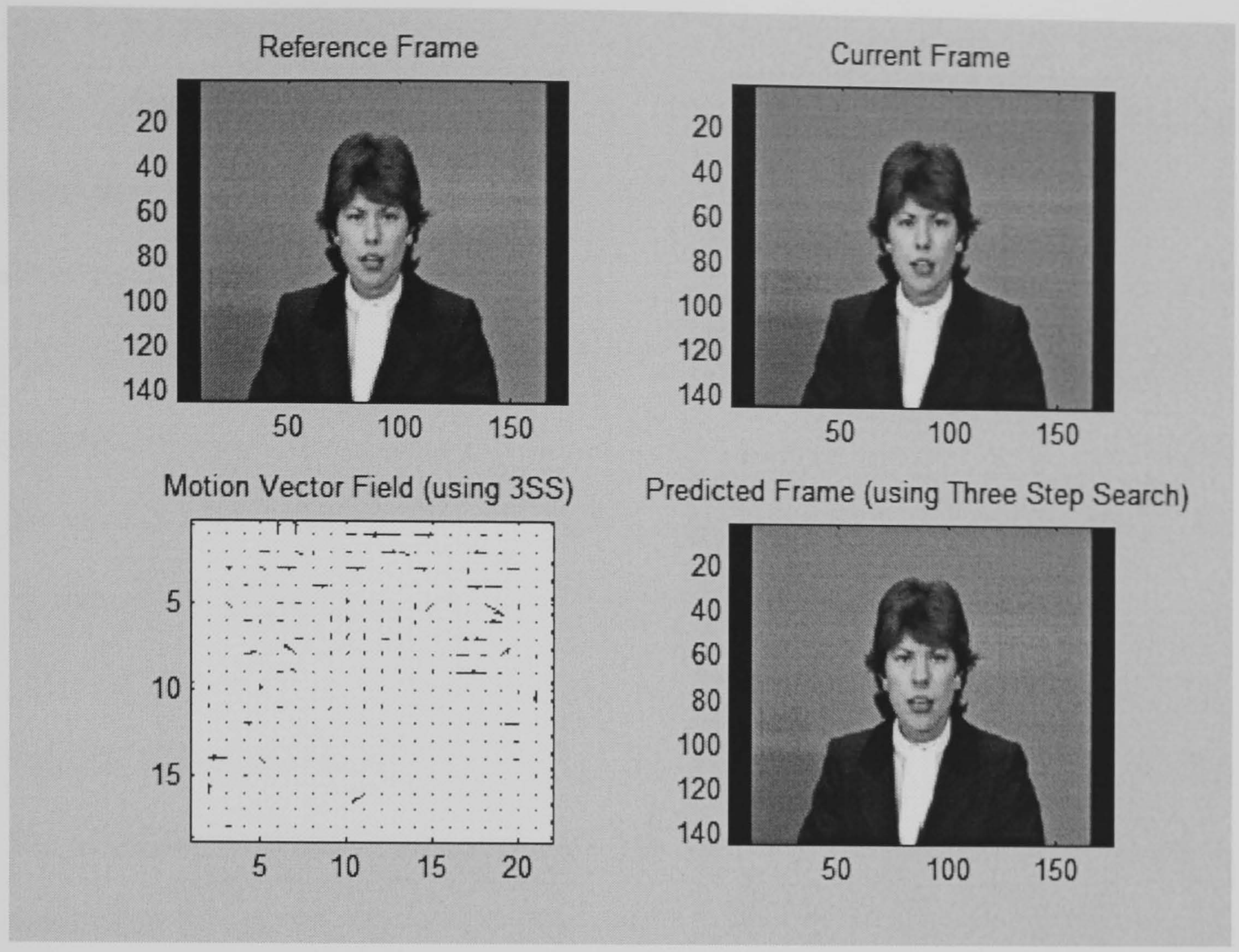


Figure (6.10) The Sample of Motion Estimation using 3SSAlgorithm

The Sample of Motion Estimation using 3SS Algorithm is shown in the figure (6.10). As shown in figure (6.10), the two upper images are the First and the Second original frames of the video sequence. The lower left diagram shows the motion vectors between the two original images. The lower image is the prediction of the second frame by using ME algorithm. The sample shows the implementation of the 3SS algorithm. The second, which is the original frame of Claire video sequence is the frame that the algorithm attempts to predict using the first frame as the reference frame. The motion vectors can be founded by a comparison of the two consecutive frames. The motion vectors add the motion information to the first frame and then the second frame can be predicted. As shown in figure (6.10), the diagram of motion vectors shows that there is a great number of dots. The dots show that displacement of motion vectors are equal to zero. In the other word, the zero motion vectors mean that there are no movement of the block at that particular area. Similarly, the ME algorithm is simulated using the 3SS algorithm. The performances of the subject quality are shown in figure (6.11)-(6.13). From the figures (6.11) the predicted frame of Claire video sequence are shown on the left hand side. From the results we can see that there are small errors on the tiny part of the face. However this is unnoticeable if the video frames runs at speeds of 30 frames per second. The Claire video sequence is considered to be a slow movement. The background of the frame remains still; only the head and shoulder are moving.

The 3SS is a fast search that tries to improve the performance of the speed of operation. The quality of predicted frame will be slightly degraded because less search points are determined as seen in figure (6.11). Nevertheless, the performance of the objective quality is good as it is shown in the bracket in the figure (6.11). The PSNR of the predicted frame is high as it ranges between 39 to 35.88 dB. The 3SS is also simulated by using the Carphone and Foreman, the results are shown in figure (6.12) and (6.13). From figure (6.12), the prediction is successfully high as it is hard to see any error in the predicted frame, even though the PSNR is not as high as that of FS algorithm. Figure (6.13), shows the results of 3SS simulation using the Carphone video sequence. Similar results are shown, the prediction can achieve a very good result as the predicted frames are very alike to the original frame. The value of PSNR is not very good in some cases such as the PSNR of the 30^{th} predicted frame. The PSNR of the predicted frame is lower than 30 dB. However the subjective quality still remains good as can be seen in figure (6.13)

Test sequence: "Claire"

Original Frame *Predicted Frame*

Frame#10

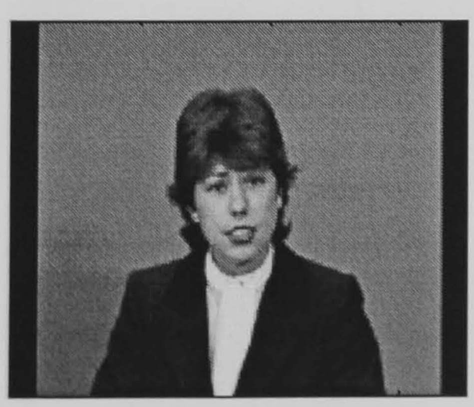
Frame#10(PSNR=39.37dB)

Frame#20

Frame#20 (PSNR=39.80dB)

Frame#30

Frame#30 (PSNR=40.74 dB)



Frame#40

Frame#40(PSNR=45.88dB)

Figure (6.11) "Claire" Subjective quality of the predicted frame using 3SS

Test sequence: "Foreman"

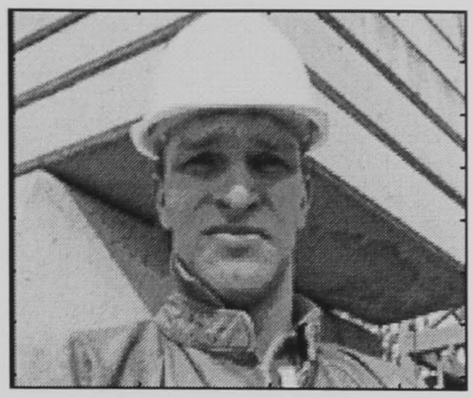
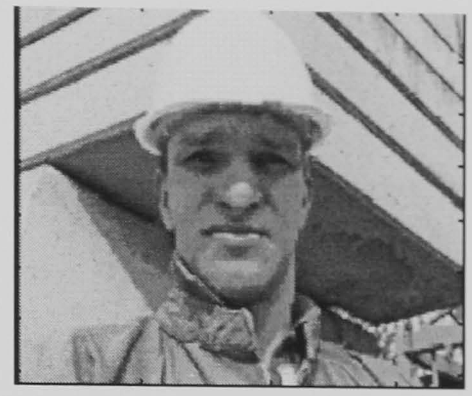
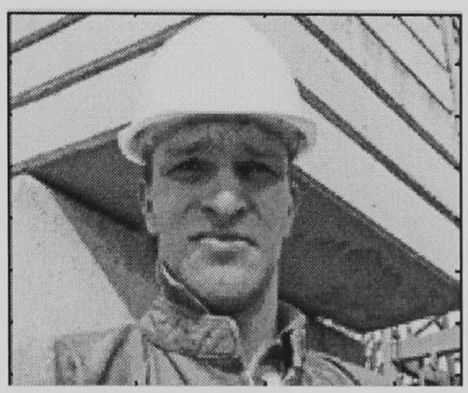
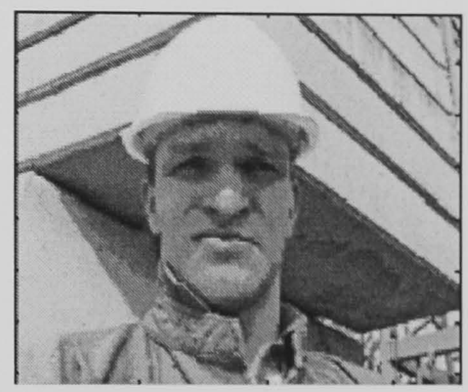
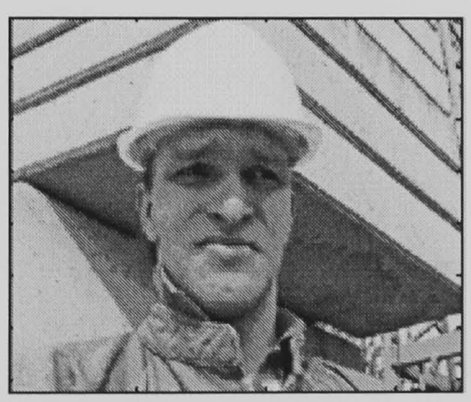
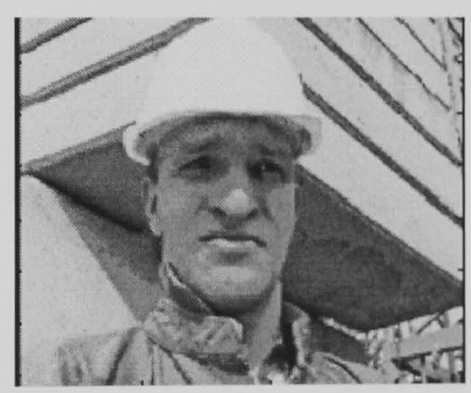
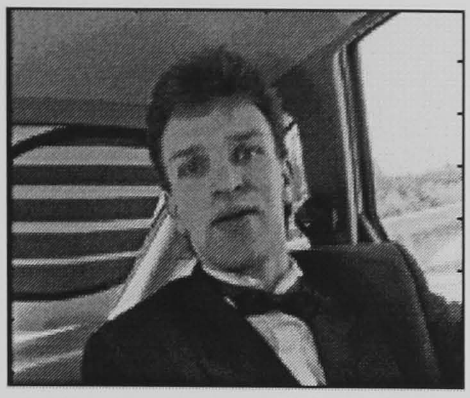
Original Frame	*Predicted Frame*
	
Frame#10	Frame#10(PSNR=30.93dB)
	
Frame#20	Frame#20 (PSNR=30.07dB)
	
Frame#30	Frame#30 (PSNR=30.48 dB)
	
Frame#40	Frame#40 (PSNR=32.49dB)

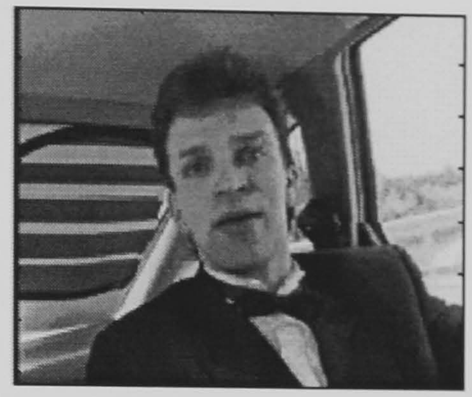
Figure (6.12) "Foreman" Subjective quality of the predicted frame using 3SS

Test sequence: "Carphone"

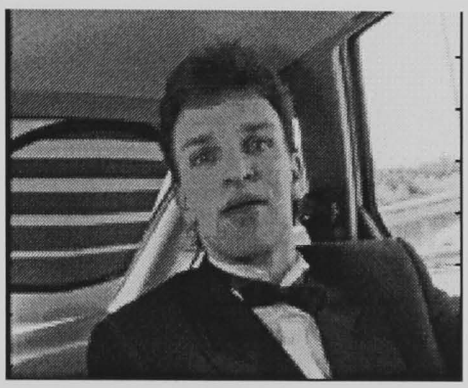
Original Frame *Predicted Frame*



Frame#10



Frame#10(PSNR=31.95dB)

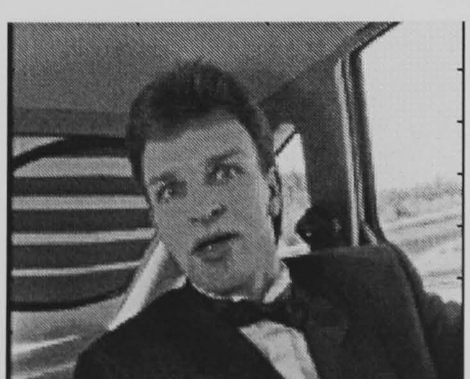


Frame#20

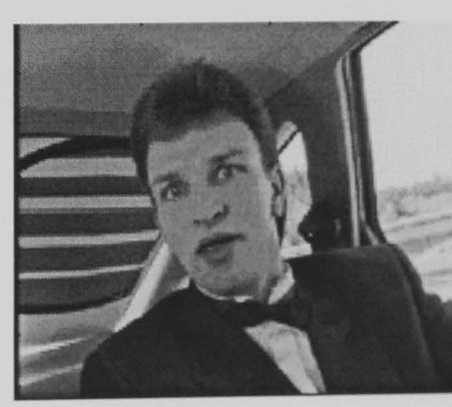
Frame#20 (PSNR=30.45dB)

Frame#30

Frame#30 (PSNR=29.89 dB)



Frame#40



Frame#40(PSNR=38.41dB)

Figure (6.13) "Carphone" Subjective quality of the predicted frame using 3SS

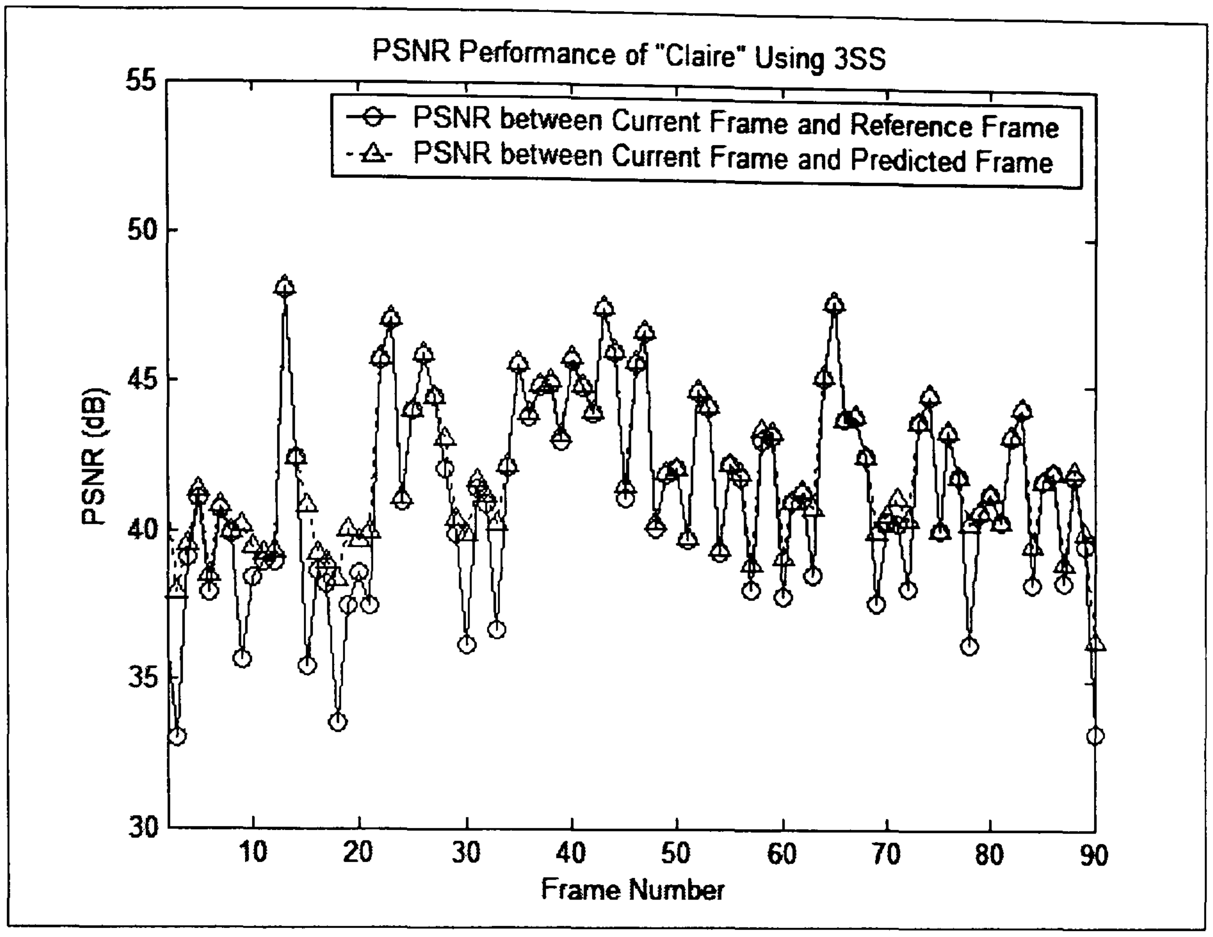


Figure (6.14) The PSNR Performance of "Claire" over 90 frames using 3SS

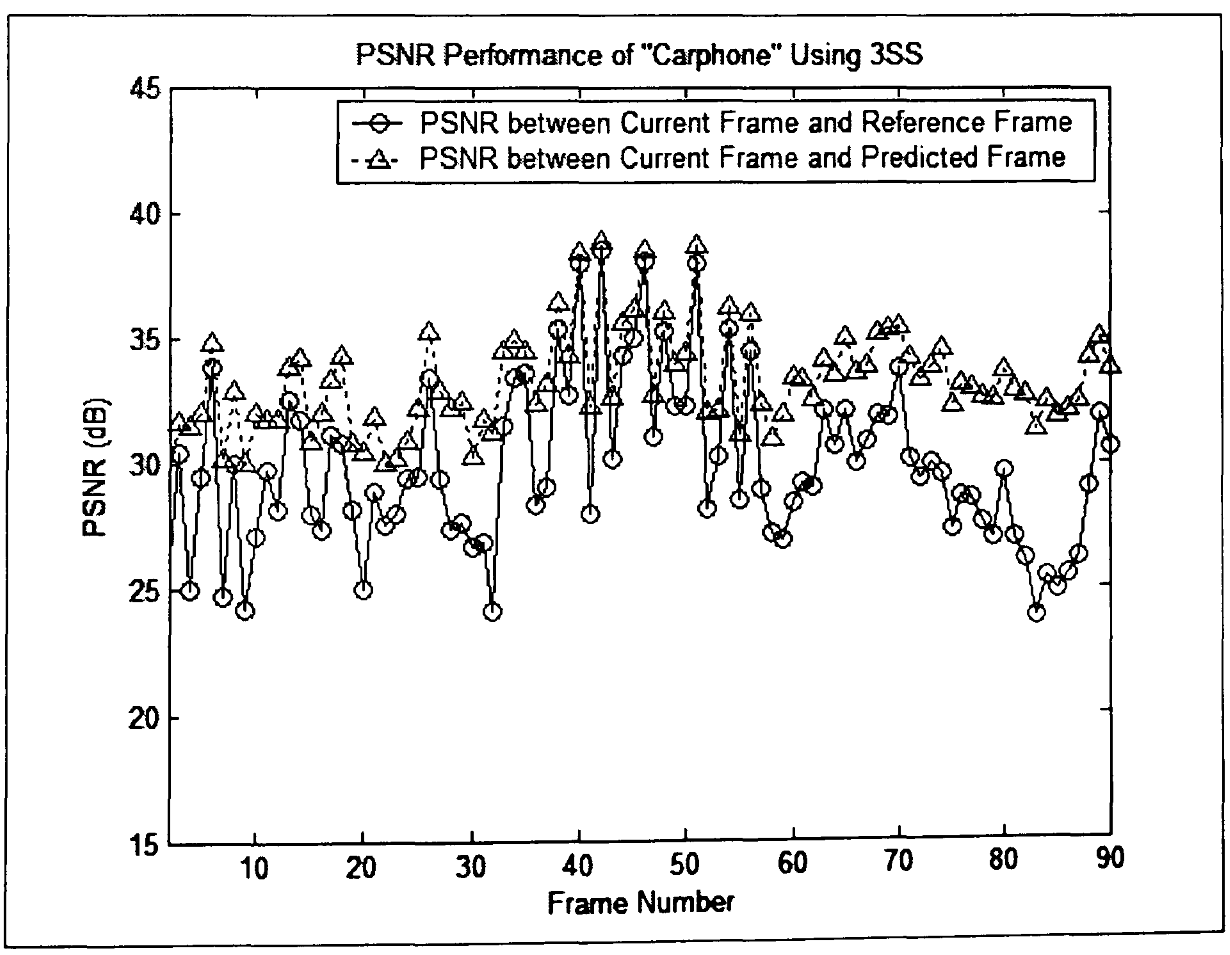


Figure (6.15) The PSNR Performance of "Carphone" over 90 frames using 3SS

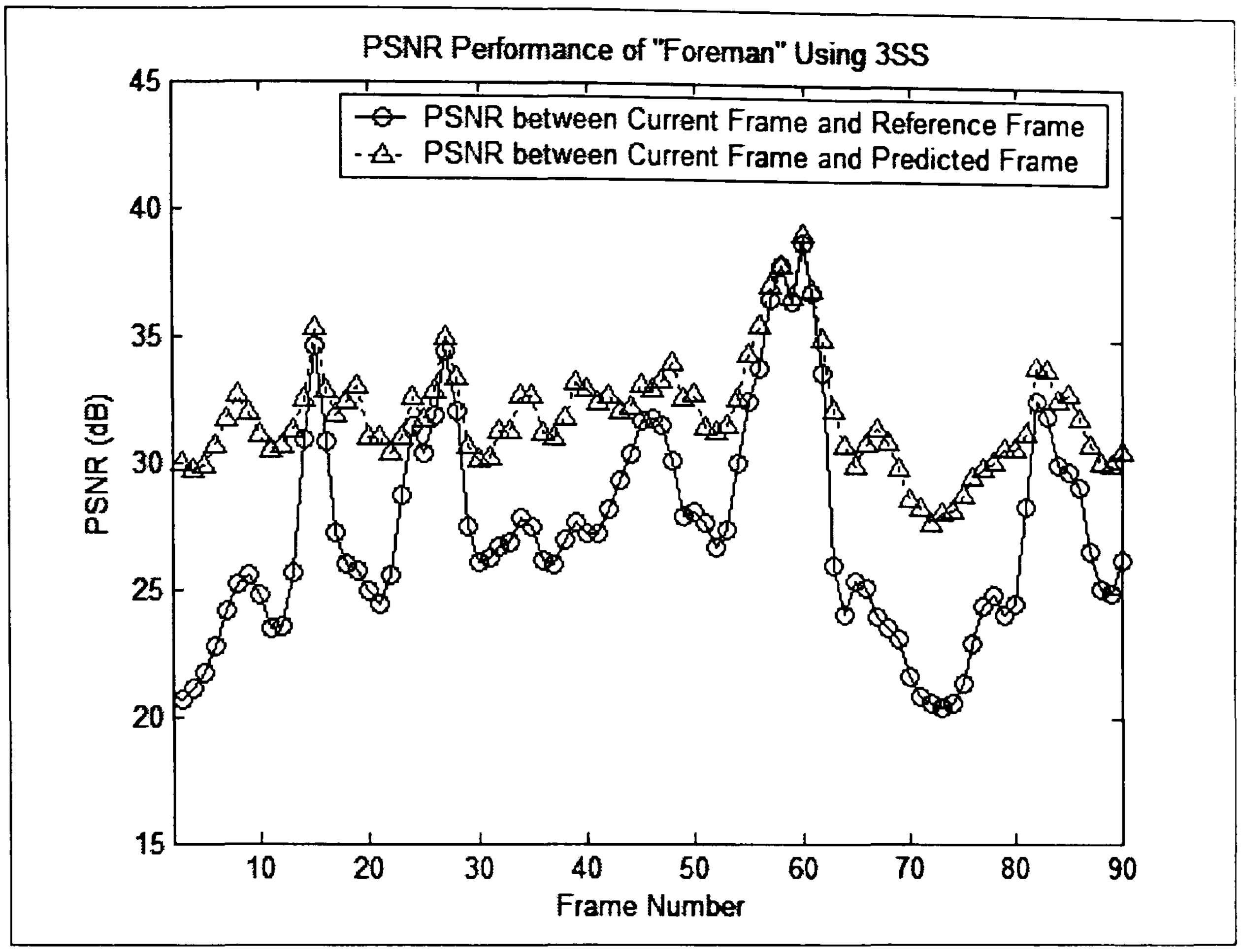


Figure (6.16) The PSNR Performance of "Foreman" over 90 frames using 3SS

The performance of objective quality assessment are simulated and the results are shown in figure (6.14) and (6.16). The PSNR2 between the current frame and the predicted frame of 3SS can achieve very high values in the Claire video sequence as can be seen from the figure (6.14). The PSNR2 values of every predicted frame is higher than 35 dB because of the slow moment of the Claire video sequence. In some cases the PSNR is the higher than 45 dB. As seen from the figure (6.15) the 3SS algorithm is tested by using carphone video sequence. The results are good as most of the predicted frame achieves the value of PSNR2 higher than 30 dB. There are a few predicted frames whose PSNR is lower than 30dB. From the figure (6.16) it can be seen that simulations are performed for 90 frames of the foreman video sequence. The performance of objective quality is good as most of the predicted frame achieve a good PSNR value.

There are a few predicted frames whose PSNR values are lower than 30dB. These frames are the 60th to 30th frames. However the quality of predicted frame is still acceptable in terms of subjective quality.

6.6 Summary

The FS algorithm and the 3SS algorithm are implemented. The simulations are tested using "Claire", "Carphone" and "Foreman" video sequence. The FS is implemented because it is the algorithm used as a benchmark. The FS gives the best performance both in terms of Subjective and Objective quality. The results in this chapter show that the PSNR between the current frame and predicted frame of all video sequence are achieved with very high values of PSNR2.The disadvantage of FS algorithm is that it is time consuming. This is the reason why the 3SS algorithm is proposed. The 3SS is very popular in International Standards as it is frequently used in the MPEG standards. Owing to its good prediction and and its speed performance, the 3SS is always used as the benchmark as well as FS algorithm. The implementation of 3SS algorithm is simulated and the results are shown in this chapter. The disadvantage of 3SS algorithm is the degradation of the prediction because the 3SS tests less search points than the FS algorithm. However, the results shows that the performance of 3SS is still very good both in terms of objective and subjective quality as the values of PSNR2 in most cases is higher than 30 dB.

Chapter 7

NOVEL APPROACHES AND ALGORITHM TESTING

7.1 Introduction

The 3SS algorithm is the conventional technique which is still used in the international standard nowadays. The 3SS algorithm is a technique that aims to improve the performance of the motion estimation. The 3SS algorithm can reduce the computational complexity which is the main disadvantage of FS algorithm. The computational complexity is the main target for most of research. The problem of mobile handset is the power consumption. The FS algorithm requires very high computational complexity which cause very high power consumption. Having reduced the computational complexity, the power consumption is also reduced. The 3SS algorithm is simple and robust. The implementation of 3SS algorithm is straightforward and every step of the process is in the same pattern. The procedure of 3SS is consists of the main 3 steps. Every step examines the eight points around the centre. So the number of the search points reduced. The average search points for each block are 25. There are a number of search algorithms have been proposed but the 3SS algorithm still remains the most favourite choice among them. The 3SS algorithm can be shown to give better performance than the FS in terms of the speed of processing. Even though the quality of the reconstructed image of the 3SS is slightly worse than the FS but the PSNR value does not show a big difference from the FS. The consideration of the subjective quality can achieve the good quality.

This chapter introduces novel techniques, Orthogonal Logarithmic Search (OLS) and Diagonal Logarithmic Search (DLS) which can improve the performance of the speed of the processing. Both OLS and DLS algorithms are proved to be faster than the 3SS and the quality of the reconstructed images is as good as the 3SS. The novel OLS algorithm is tested by using the well-known benchmark video sequence: Claire, Foreman and Carphone. The performance of OLS is shown in both the subjective quality and objective quality. The time consuming of OLS algorithm is also shown. The OLS algorithm is examined and the results are compared with the FS algorithm and the 3SS algorithm. The approach of the OLS algorithm is to reduce the number of search points and to propose a new pattern of searching. The average search points of OLS algorithm is 13 which is less than that of the 3SS algorithm. The processing time of the OLS is compared with the FS algorithm and the 3SS algorithm. The second novel algorithm, DLS algorithm, is also proposed. The main two metric, time and PSNR, are used as the quality assessment. The PSNR is a metric to verify the ability of the prediction. Only the value of PSNR is good enough to justify the capacity of the algorithms. However, the subjective quality is also examined. The subjective quality is used as an supportive measurement. The subjective quality assessment is shown the quality of algorithms in term of perception. The subjective quality is usually consuming time. The subjective quality here is only to justify if the quality of prediction is acceptable or not. The speed of operation or the time required to run the algorithm is another metric which can indicate the computational complexity of the algorithms. The time of processing relatively relies on the complexity of the algorithm. The algorithm with high complexity requires a higher time than the algorithm with less complexity.

The approach of the DLS algorithm is to reduce the number of search points and the new search pattern is proposed. The performances of the novel DLS algorithm are shown. The results of simulation are compared with the conventional FS algorithm and the 3SS algorithm. Both algorithms choose MAD as the matching criteria since it is the best matching criterion among the well-known matching criteria. The advantage of MAD is the smaller computational complexity. The calculation of MAD is straightforward. Also the MAD criterion is used in the well-known standard such as MPEG. At the end of this chapter, the comparisons between those two novel algorithms are also shown. The subjective quality with the value of PSNR is shown. The comparisons between these two methods are discussed. The average PSNR over the 90 frames of video sequence is shown. The average PSNR can indicate which algorithm is better than the others. From the value of average PSNR, the better algorithm can be proved. The complexity of both algorithms is also examined. The computational complexity is measured by the time requirement. The comparison of the time consuming between the OLS algorithm and the DLS algorithm is shown.

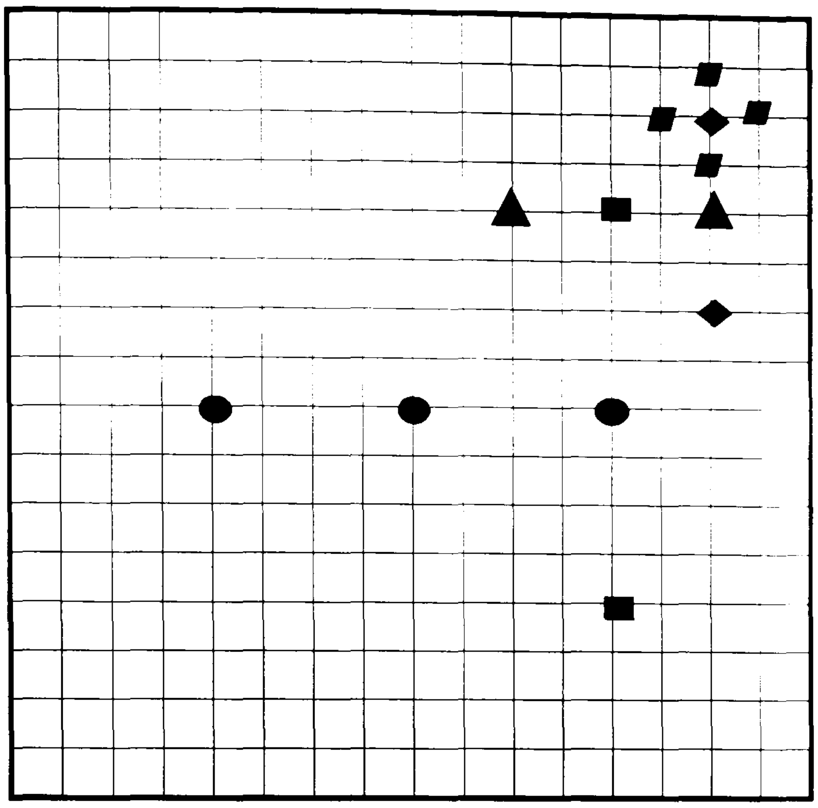
7.2 Orthogonal Logarithmic Search (OLS) Algorithm

The Orthogonal Logarithmic Search (OLS) Algorithms is a novel technique which aims to improve the performance of search algorithm. The approach of OLS algorithm is to design the new search pattern. The new search pattern can reduce the number of the search points by the search points of the OLS algorithm are only 13 points. The search pattern of the OLS algorithm is shown in figure (7.1). The procedure of the OLS algorithms is hybrid between the horizontal and the vertical stages. The procedure of the OLS algorithm start by the current frame is divided into equal sized non-overlapping rectangular blocks. The frame dimensions are multiples of the block size and square blocks are mostly used. The block sizes which usually adopted by the MPEG Standard are 8×8 and 16×16. After the frame is partition into square blocks, then the comparisons between the target block and candidate blocks are determined to search for the best block matching. Every blocks of the current frame is compared to the candidate blocks within the previous frame according to the search pattern. The OLS algorithm has both a vertical and horizontal stage. The OLS searches the best matching block in the horizontal stage first, then the best block matching of the horizontal stage will be the centre for the vertical stage. The search is initialised in the centre of search window or search region (usually 15×15) with two searching points apart from centre with some +/- distance (st) horizontally. The distance of two searching points follows the logarithmic function shown in equation (7.1).

$$st = \log_{10}(s+1)/\log_{10}(2) \tag{7.1}$$

Where s × s is the size of search window

Then the best matching points in the first steps will be the centre of the vertical stages. Another two searching points with distance (st) vertically will be evaluated. Every two steps, the distance (st) is reduced by half. The procedure continues until the distance is equal to one, and then the final step is reached. During the final step, all four searching point will be evaluated to find the best block matching. The searching pattern is shown in figure (7.1). One main advantage with the OLS algorithm is that the searching point is less than the 3SS, so the speed of process is better than the three-step search.



● Points for Stage1 ■ Points For Stage2 ▲ Points for Stage 3 ▰ Points for Stage4

Figure (7.1) Search Pattern of the Orthogonal Logarithmic Search Algorithm

From figure (7.1) is an example of the search pattern. The search point starts from the 3 points of the horizontal direction. In this case the far left search point receives the minimum MAD. So this point becomes the centre of the vertical stage. The two more points in vertical direction are examined. Among these three points in the vertical direction, the point with minimum MAD becomes the centre of the next step. In this example, the point on the top receives the minimum MAD. Then the procedure is carried to the next step where the step size is reduced by half. The procedure of this step is as same as before. The point with minimum MAD of this horizontal is found and becomes the centre of the next step in vertical direction. From the three point of the vertical direction, the point with minimum MAD is found. This point becomes the centre of the next step. The next step is become the final step because the step size is reduced to be one. So this step, the four search point are examined. The search point with minimum MAD is then the best matching block.

7.2.1 Implementation of Orthogonal Logarithmic Search Algorithm

The OLS algorithm is simulated by using MATLAB software. The first step of simulation is to extract the frames from video sequences. The video frames are stored in the hard disk. After the frames are extracted, the motion estimation process begins. The frames are fed into the motion estimation process, then two consecutive frames (1^{st} frame and 2^{nd} frame, 2^{nd} and 3^{rd} frame, 3^{rd} frame and 4^{th} frame,....) are compared by using OLS algorithm. The two consecutive frames are previous frame and current frame. Suppose that the current frame is n^{th} frame so the previous frame will be $(n-1)^{th}$ frame where n is integer number. The current frame is then divided into 8×8 blocks. Since the QCIF format (176×144) are used in the simulation, so the current frame is segmented into 396 blocks. Each block is considered as the target block. Every target block search the best block matching in the previous frame by using OLS algorithm. The OLS algorithm then begins. The steps of OLS algorithm is summarised as bellows:

Step1: The search is initialised at the centre of search windows with position (0,0). The search window's size is 15×15. The two more positions in the horizontal direction at the position (st,0) and (-st,0) are searched. The st can be found from the equation (7.1). The MAD is used as the matching criteria. The MAD used to evaluate the similarity between the target block with the candidate blocks at those position in the previous frame. The position with the minimum MAD becomes the centre of the vertical direction. The minimum position can be at (st,0), (-st,0) and (0,0). Let's the position with the minimum MAD is (st,0). The (st,0) is the centre of the next step.

Step2: The vertical stage begins by using position (st,0) as a the centre of the vertical stage. The two more position in the vertical direction at the position (st,4) and (st,-4) are search. The MAD values are evaluated among those three positions. The centre is moved to the position with the minimum MAD.

Step3: The distance (st) is reduced by half.

Step4: Repeat step step1 to step 3 until the distance (st) is one.

Step5: The final step, all four searching point will be evaluated to find the best block matching.

After the position with the minimum MAD is found, the motion vector can be also found. The motion vectors indicate where the best matching block locate. The procedure is the same for every target blocks. After the motion vector of every blocks are found, the current frame can be reconstructed by using the information of the previous frame plus the motion vector. This procedure is called motion compensation.

7.2.2 Performance of the Orthogonal Logarithmic Search Algorithm

The OLS algorithm is implemented and simulated. The simulation aims to justify the performance of OLS algorithm. The OLS algorithm is tested by using both objective quality and subjective quality assessment. As shown in figures (7.2)-(7.4) where samples of the results are shown. The subjective quality is required because the objective PSNR measure is not always an appropriate indicator. The subjective quality of the reconstructed frames is shown. The 9^{th}, 19th, 29th and 39^{th} frame of video sequences are chosen as the reference frame. These frames are compared with 10^{th}, 20^{th}, 30^{th}, 40^{th} frame in a row. Having used the OLS search pattern, the best matching block are found as well as the motion vectors. The results of "Claire" video sequence show that the OLS can successfully predict the current frame as the subjective quality is very good. The subjective quality of the prediction of 20th, 30th and 40th frame are very good as the difference between the predicted frames and original frames is very small. The prediction of the 10th frame causes an error at the lip of the "Claire". However the error is hardly noticeable. The performance of subjective quality is also very good when the OLS is tested on the Foreman video sequence. The performance of the subjective quality of "Foreman" video sequence using OLS is shown in figure (7.3). The predictions achieve very high performance in the subjective quality as the error is barely discernible. Even though the "Foreman" video sequence considered to be more difficult to predict than "Claire", the OLS algorithm successfully predicted the current frame. The PSNR values of all the samples are high and higher than 30 dB. The "Carphone" video sequence was also tested in a similar way. A sample of the results of the predicted frames is shown in figure (7.4). The performance of the prediction is good as shown in figure (7.4). The errors also are unnoticeable. The prediction is considered to be quite accurate as the PSNR performance is high. The PSNR values range from 32.51 dB to 38.41 dB. The prediction of "Carphone" video sequence is more accurate than that of "Foreman" video sequence. Nevertheless, the prediction of "Claire" video sequence is still more accurate than that of "Carphone" video sequence as the "Claire" video sequence is easier to predict. The prediction of "Claire" video sequence is higher than 35dB and it is as high as 45 dB in some cases. From figure (7.2)-(7.4), the OLS algorithm can be proven to be successful as a motion estimation algorithm. The results show that the quality of the prediction is very good in all three video sequences in terms of subjective quality. In most cases the errors are unnoticeable as they are extremely small.

Test sequence: "Claire"

Original Frame *Predicted Frame*

Frame#10

Frame#10(PSNR=38.42dB)

Frame#20

Frame#20 (PSNR=38.56dB)

Frame#30

Frame#30 (PSNR=36.19dB)

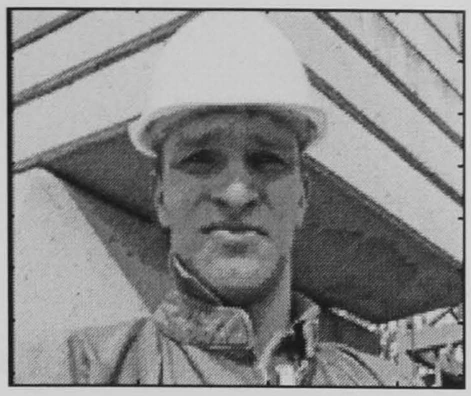
Frame#40

Frame#40(PSNR=45.77dB)

Figure (7.2) "Claire" Subjective quality of the predicted frame using OLS

Sequence: "Foreman"

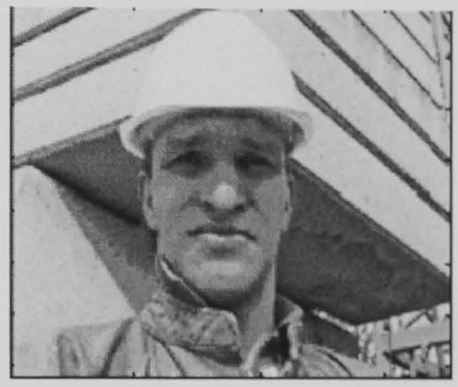
Original Frame *Predicted Frame*



Frame#10

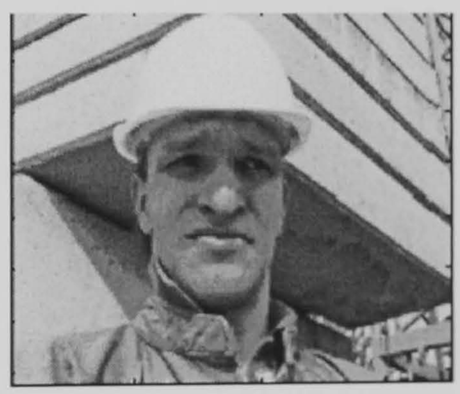
Frame#10(PSNR=31.59dB)

Frame#20

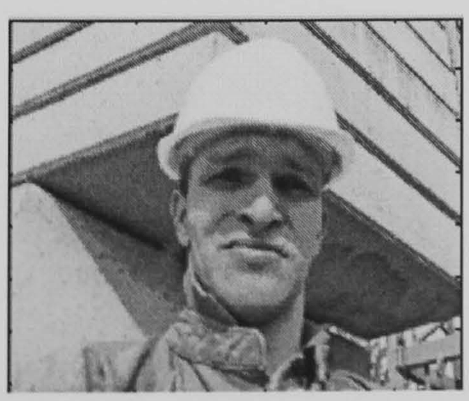


Frame#20 (PSNR=30.42dB)

Frame#30



Frame#30 (PSNR=31.03 dB)



Frame#40

Frame#40(PSNR=33.35dB)

Figure (7.3) "Foreman" Subjective quality of the predicted frame using OLS

Test sequence: "Carphone"

Figure (7.4) "Carphone" Subjective quality of the predicted frame using OLS

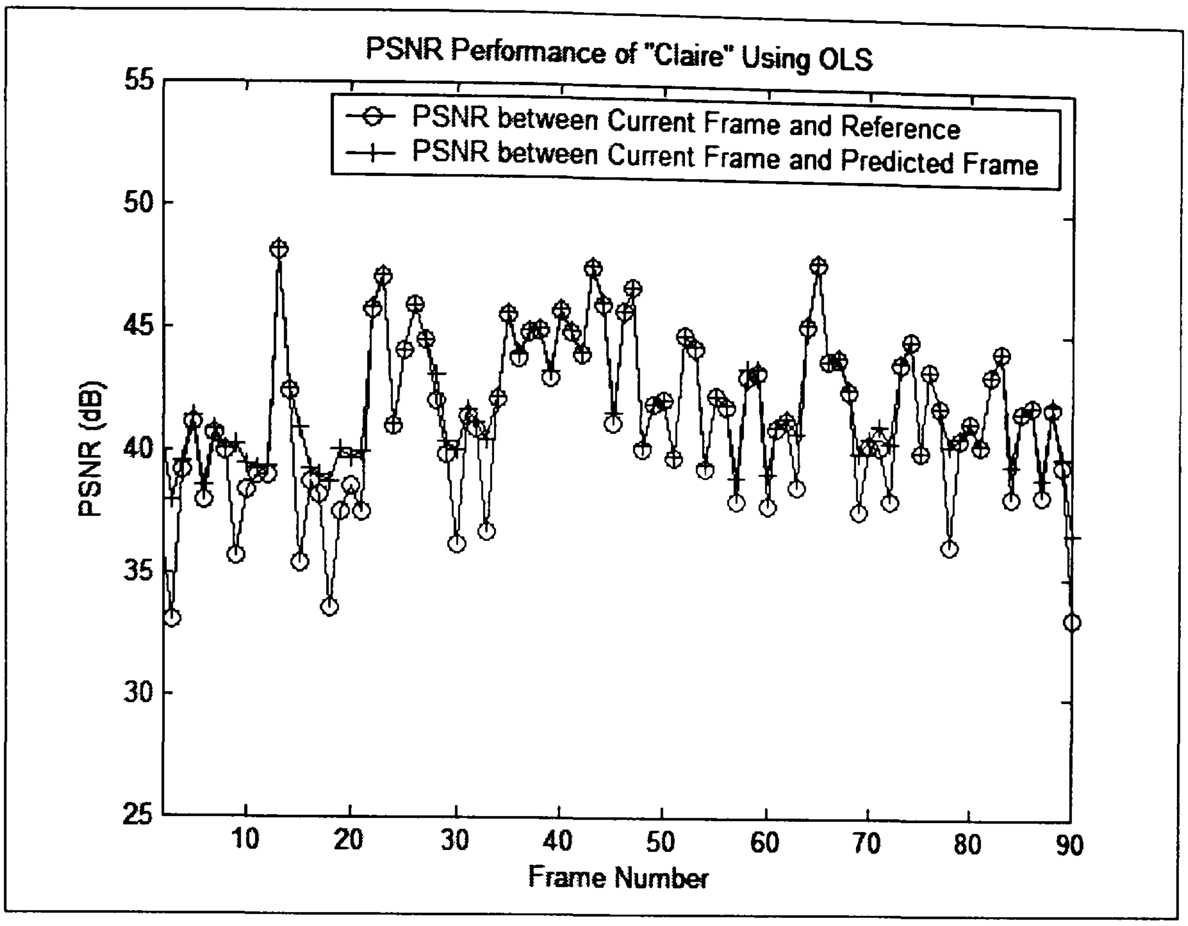


Figure (7.5) The PSNR Performance of "Claire" over 90 frames using OLS

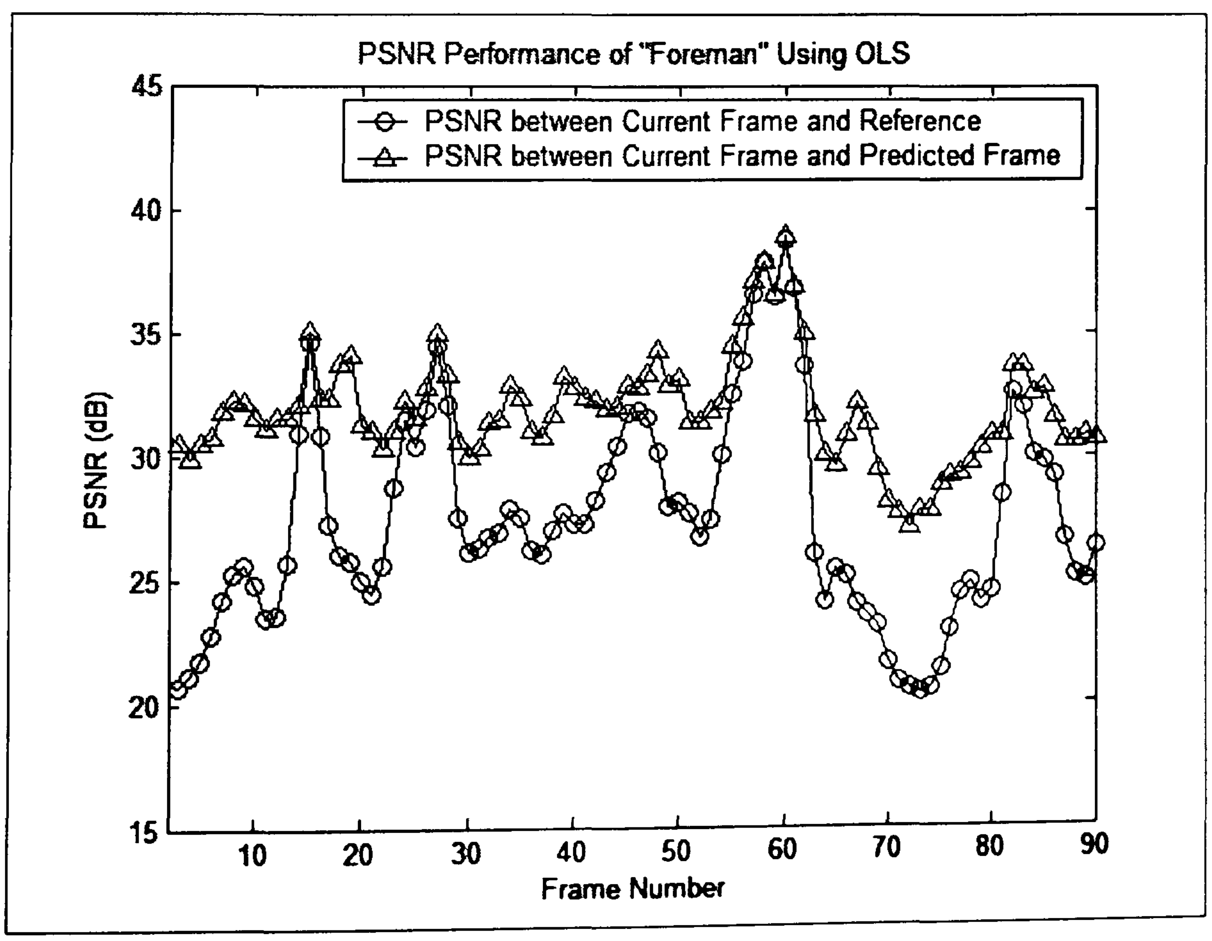


Figure (7.6) The PSNR Performance of "Foreman" over 90 frames using OLS

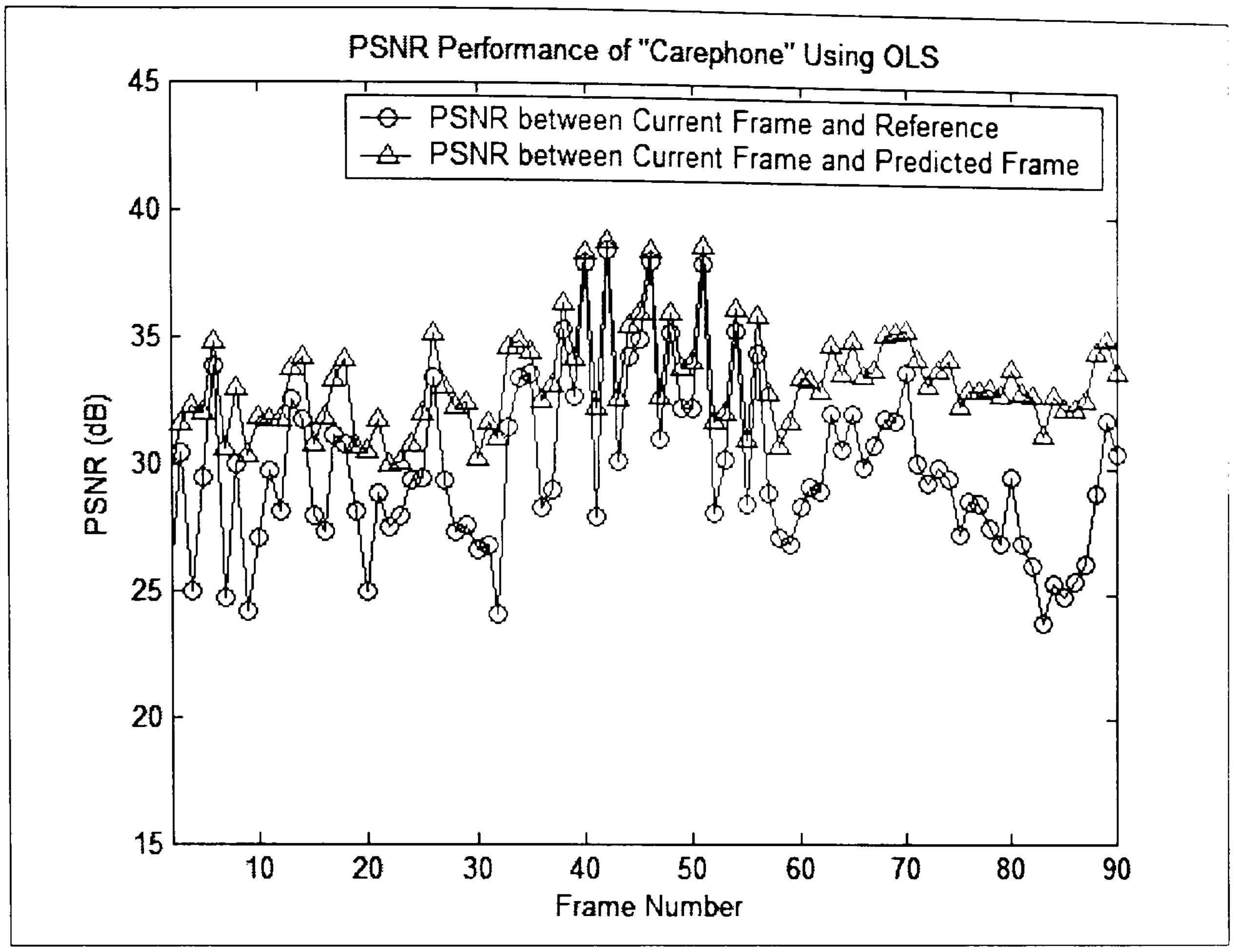


Figure (7.7) The PSNR Performance of "Carphone" over 90 frames using OLS

The objective quality was also tested along with the 90 frames of each video sequences. The results are shown in figure (7.5) – (7.7). The simulation justify the performance of the OLS algorithms. The PSNR between current frame and predicted frame (PSNR2) of "Claire" video sequence is shown in figure (7.5). The results show that the predictions of 90 frames of the Claire video sequence achieve very high PSNR2. All the predictions are higher than 35 dB. These predictions are derived from the previous frame with the motion vectors. The PSNR of the "Foreman" and "Carphone" are also shown in the figures (7.6) and (7.7) respectively. The predicted frame quality of "Foreman" is slightly worse than that of "Claire". The PSNR2 of the "Foreman" video sequence is less than that of "Claire" video sequence. However the values of PSNR2 are still good as they are greater than 30 dB. In some cases, the quality of prediction of the "Carphone" video sequence drops due to the greater movement of foreman. As seen from figure (7.6) the PSNR2 drop during the 67^{th} frame to 73^{rd} frame. The values of PSNR2 of these frames lie between 26 dB and 30 dB. Even though these values are lower than 30 dB, but the quality of prediction is still acceptable. The predictions of these frames are not only difficult for OLS algorithm but also for the 3SS and the FS algorithm. The great movement causes difficulties for OLS, 3SS and FS. Therefore the OLS should be able to predict these frames as well as the 3SS and FS algorithm. In case of the "Carphone" video sequence, the OLS can achieves very good result as the PSNR2 of the predicted frame are all higher than 30 dB.

7.2.3 Comparative Performance of OLS

The OLS is the alternative algorithm which can achieve very good prediction as it can achieve very high PSNR. This section shows the performance of OLS algorithm compared with the 3SS and the FS algorithm. As seen in figure (7.8) the quality of the predicted frames of OLS algorithm are shown along with that of the 3SS and the FS algorithm. The simulation takes place over the 90 frames of video sequence. From figure (7.8), the results show that the performance of OLS is as good as 3SS and FS as the PSNR are almost the same. There are only a few predicted frames for which the PSNR values are slightly smaller than that of 3SS and FS. Thus the performance of OLS on "Claire" video sequence is as good as that of 3SS and FS. The average PSNR over 90 frame of Claire video sequence are 41.66 dB, 41.62 dB, and 41.63 dB for Full Search, 3SS and OLS respectively.

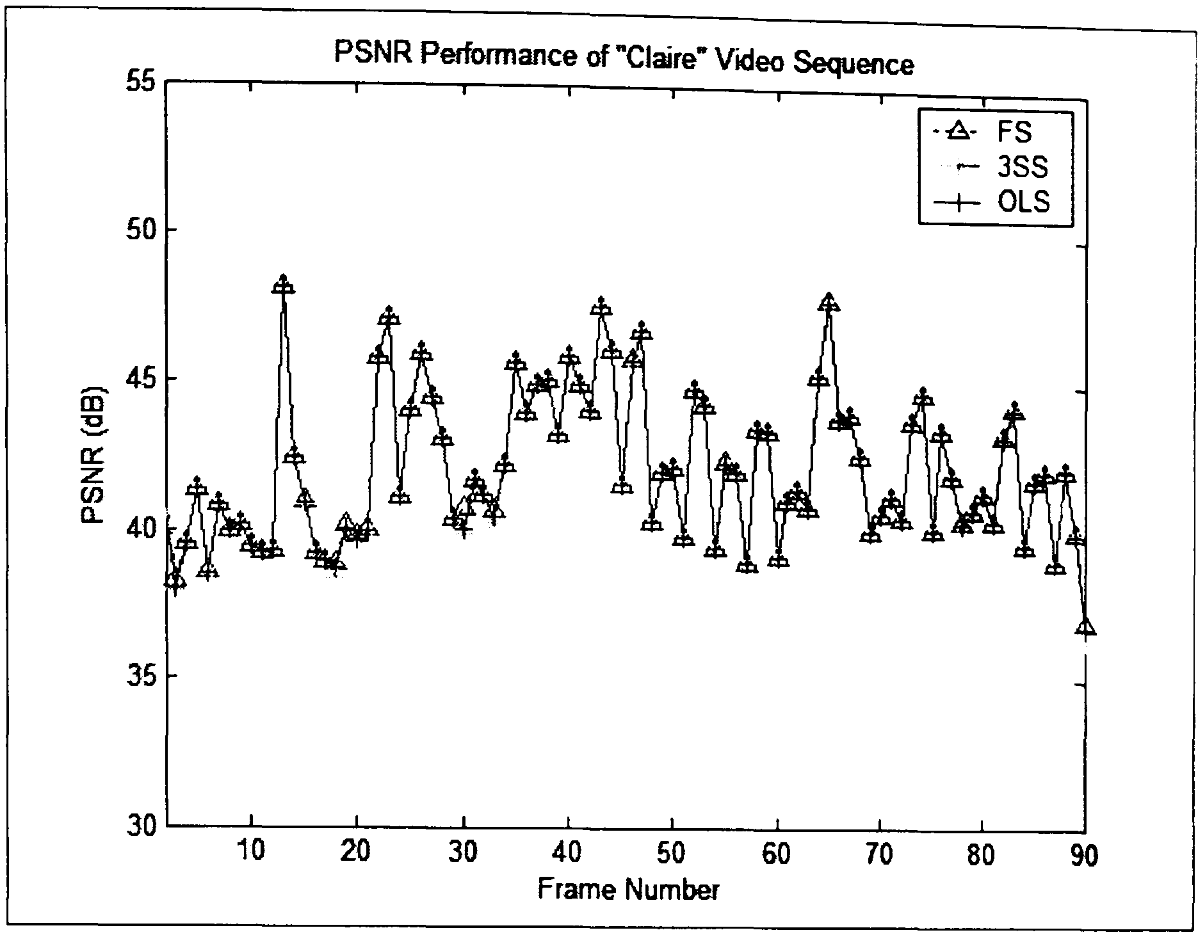


Figure (7.8) Comparative PSNR Performance of "Claire" Using FS, 3SS and OLS

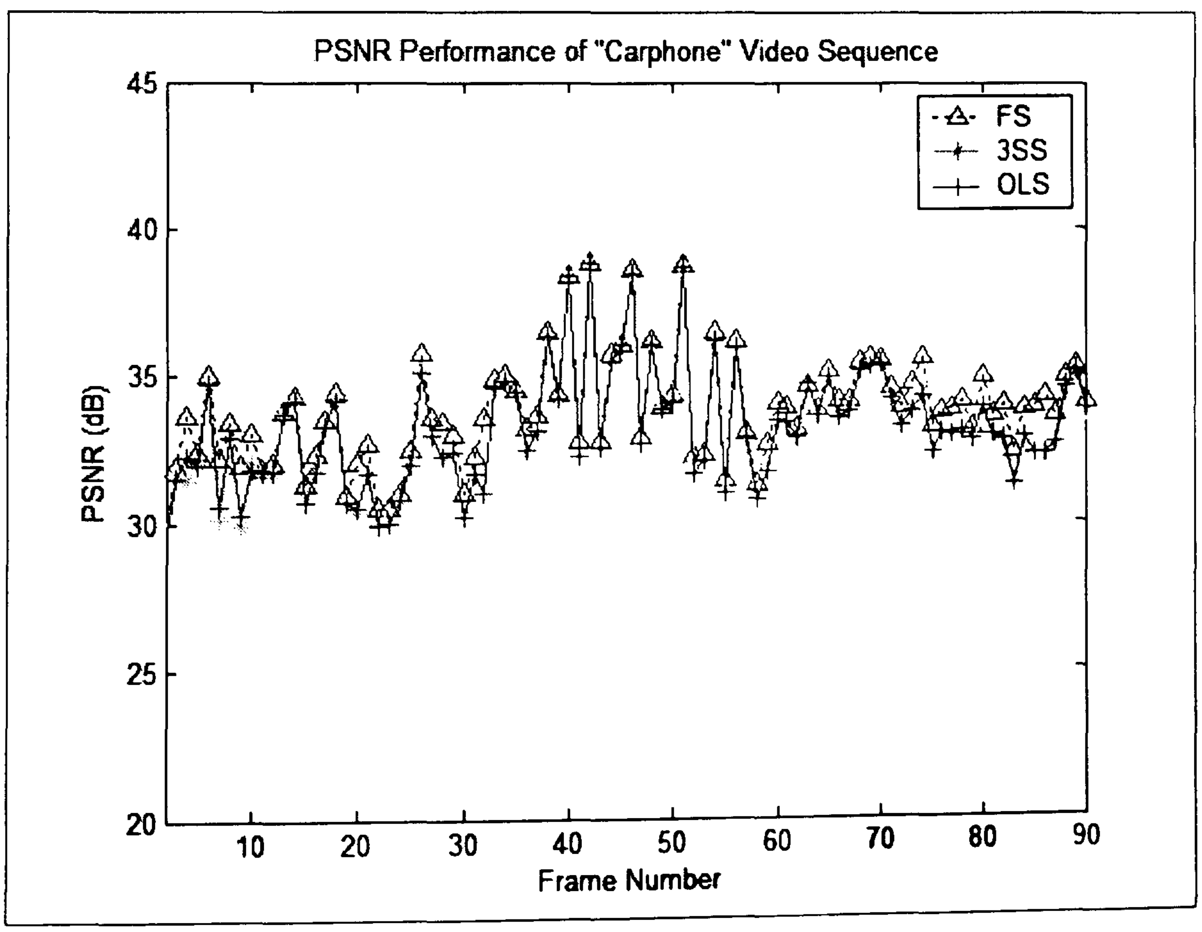


Figure (7.9) Comparative PSNR Performance of "Carphone" Using FS, 3SS and OLS

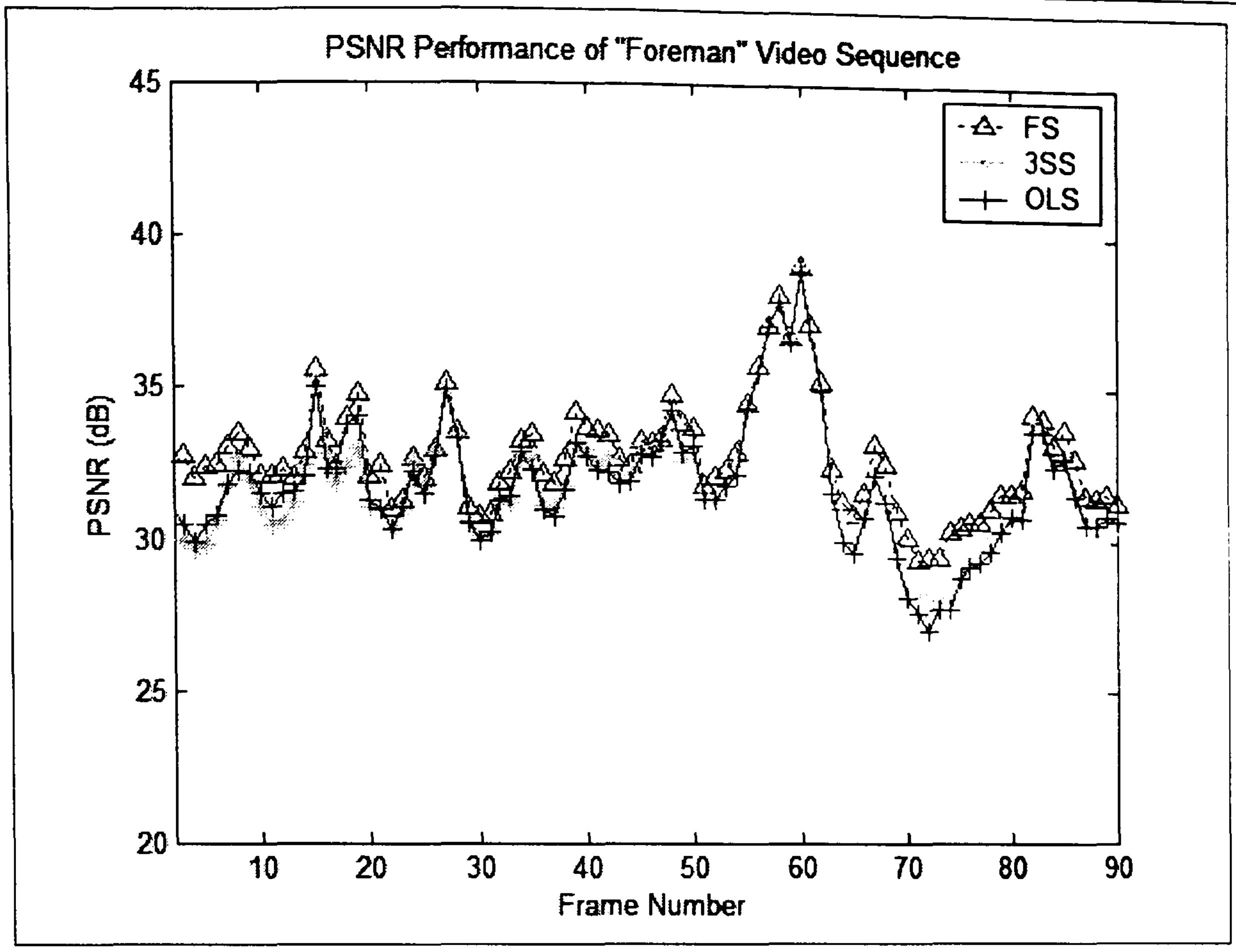


Figure (7.10) Comparative PSNR Performance of "Foreman" Using FS, 3SS and OLS

The PSNR performance of "Carphone" Video sequence using OLS is shown in figure (7.9). As seen from figure (7.9), the PSNR of the 3SS and the OLS are both slightly worse than FS algorithm as both algorithms are evaluated using less search points than the FS. The average PSNR over 90 frame of Carphone video sequence are 33.43 dB, 32.82 dB, and 32.88 dB for Full Search, 3SS and OLS respectively. The results on the average PSNR shows that the average PSNR of OLS is better than that of 3SS. The "Foreman" video sequence is simulated using OLS and the results of the PSNR are shown in figure (7.10). The PSNR performance of the 3SS and the OLS are not as good as that of FS. However the results are good enough for the prediction since the PSNR is still high. Even though the OLS performance is not as good as the FS, but it is comparable with the 3SS. The PSNR of the OLS is almost the same as that of 3SS and it is better than that of 3SS in some case. The average PSNR over 90 frames of Foreman video sequence are 32.39 dB, 31.53 dB, and 31.51 dB for Full Search, 3SS and OLS respectively. From the average PSNR point of view, the OLS is as good as that of 3SS since the values of PSNR of OLS is only 0.01 dB less than that of 3SS.

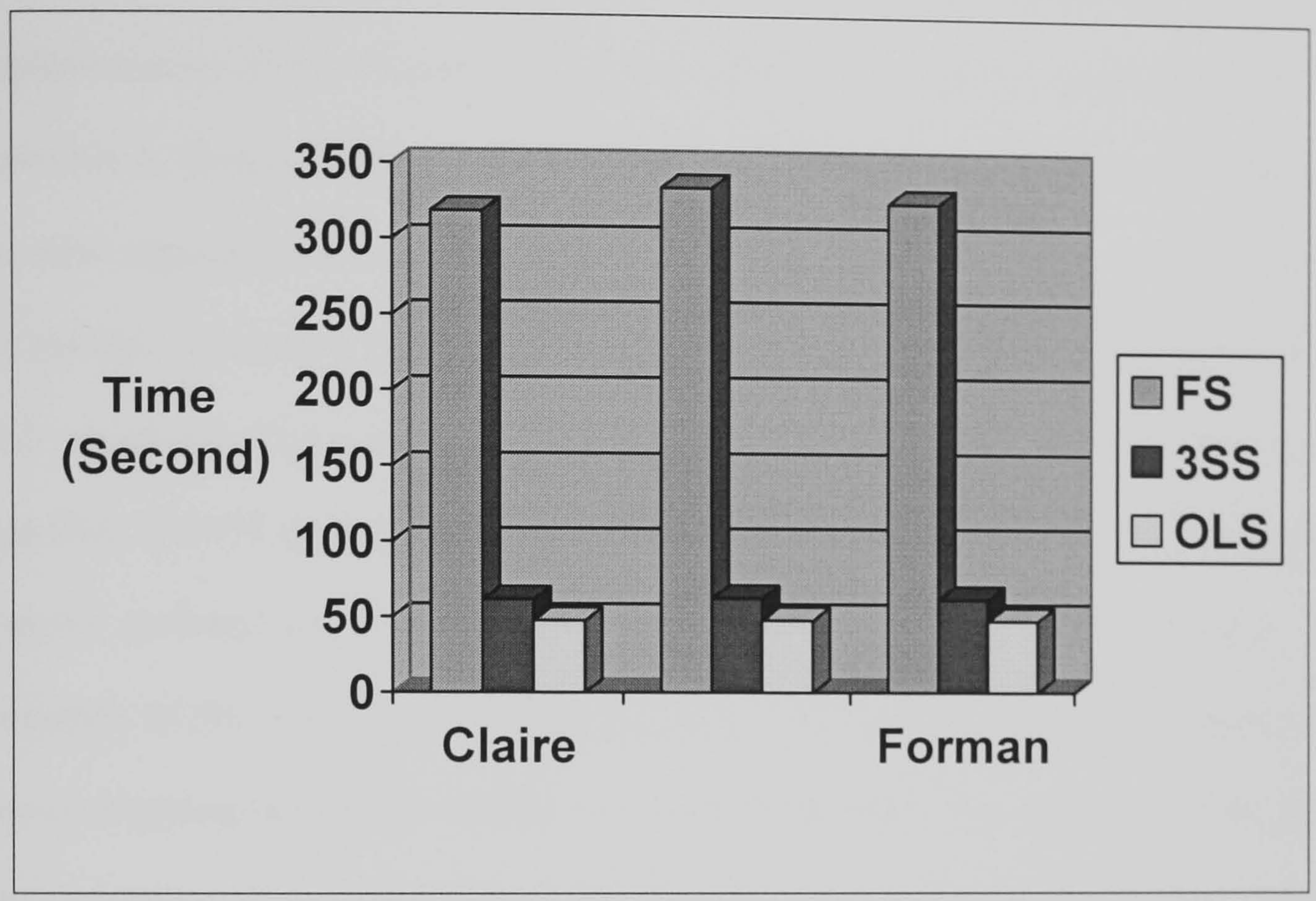


Figure (7.11) Comparative Speed of Operation using FS, 3SS and OLS

The advantage of OLS is the speed of operation. The comparison of processing time is shown in figure (7.11). The processing time is the total time of the algorithm spends on prediction of 90 frames video sequence. As seen in figure (7.11), the processing time of the FS is extremely high when it tested on all three video sequences. The FS algorithm spends more than 300 seconds for predicting 90 frames for each of the video sequence. The 3SS can improve the performance of the processing time of operation as it can be seen in figure (7.11). The 3SS is much faster than that of the FS.

The OLS gives a better performance than 3SS on the processing speed. The OLS can improve the processing time and the quality of prediction still remains as good as that of 3SS. Therefore the OLS is a better choice than the 3SS.

7.3 Diagonal Logarithmic Search (DLS) Algorithm

The novel DLS algorithm is another algorithm which has successfully improved the performance of search algorithm. The novel DLS algorithm is simple and robust. The implementation of the algorithm is straight forward. The search pattern of the DLS algorithm is shown in figure (7.12). The main different between the DLS algorithm and the other algorithms is the search pattern. The advantage of the DLS algorithms is that the number of search points is less than that of the 3SS algorithm and also that of the OLS algorithm. The objective of DLS algorithm is as same as the other search algorithm. The DLS algorithm aims to find the best matching block. The DLS algorithm improve performance both the quality of prediction and the processing time. The procedure of the OLS algorithm start by the current frame is divided into equal sized non-overlapping rectangular blocks. The frame dimensions are multiples of the block size and square blocks are mostly used. The block sizes which usually adopted by the MPEG Standard are 8×8 and 16×16. After the frame is partition into square blocks, then the comparisons between the target block and candidate blocks are determined to search for the best block matching. Every blocks of the current frame is compared to the candidate blocks within the previous frame according to the search pattern.

The DLS Algorithm has both a vertical and horizontal stage. Firstly, the method searches for the similarity in both horizontal and vertical axis initially some distance +/- *disp* from the centre block. From the five blocks, the block that produces the minimum distortion function becomes the centre of the next step. The distortion function or the matching criteria using in this technique is MAD. The search pattern of DLS is shown in figure (7.10). The DLS is very simple to implement and found to be a faster algorithm than the 3SS algorithm and the OLS algorithm. For every step of this algorithm the displacement will reduce by half. Every step considers four surrounding points. The average search point of this the technique is approximately 12 points. The procedure continues until the distance s is equal to one, and then the block which has the minimum in this stage is the best matching block. One main advantage with the DLS algorithm is that it does not have three steps like the 3SS algorithm. Each time the steps of the procedure are related to the size of the search area using the formula below

$$disp = \log_{10}(s+1)/\log_{10}(2) \tag{7.2}$$

Where s × s is the size of the area

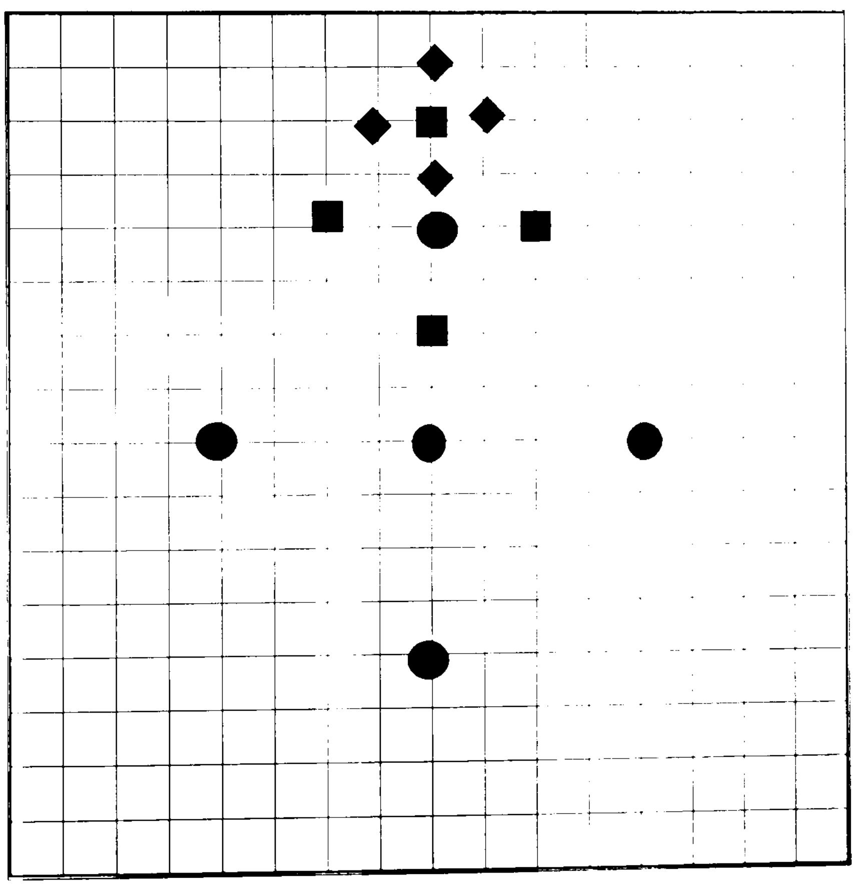


Figure (7.12) the Search Pattern of the Diagonal Search Algorithm

From figure (7.12) is an example of the search pattern. The search point starts from the centre of search window and the four points around the center. The MAD values of these five points are evaluated. The point with the minimum MAD becomes the centre of the second step. In this case the search point above the centre receives the minimum MAD. So this point becomes the centre of the next step. The four more points around the new centre are tested. Among these five points of this stage, the point with minimum MAD becomes the centre of the next step. The next step is become the final step because the step size is reduced to be one. So this step, the four search point are examined. The search point with minimum MAD is then the best matching block. The detail of the implementation of DLS algorithms is explained in the next section.

7.3.1 Implementation of Diagonal Logarithmic Search Algorithm

The DLS is simulated by using MATLAB software. Firstly, the video sequence is divided into frames and stored in the hard disk. These frames are fed into the motion estimation process. The two consecutive frames (1^{st} frame and 2^{nd} frame, 2^{nd} and 3^{rd} frame, 3^{rd} frame and 4^{th} frame,....) are compared. The search pattern follows the DLS search algorithm. The two consecutive frames are previous frame and current frame. Suppose that the current frame is n^{th} frame so the previous frame will be $(n-1)^{th}$ frame where n is integer number. The current frame is then divided into 8×8 blocks. Since the QCIF format (176×144) are used in the simulation, so the current frame is segmented into 396 blocks. Each block is considered as the target block. Every candidate block search the best block matching in the previous frame by using DLS algorithm. The OLS algorithm then begins. The steps of OLS algorithm is summarised as bellows:

Step1: The search is initialised at the centre of search windows with position (0,0). The search window's size is 15×15. The four positions in both horizontal direction and vertical direction at the position (*disp*,0) and (-*disp*,0), (0,*disp*) and (0,-*disp*) are searched. The *disp* can be found from the equation (7.2). The MAD is used as the matching criteria. The MAD used to evaluate the similarity between the candidate block with the block at those position in the previous frame. The position with the minimum MAD becomes the centre of the next step. The minimum position can be at (*disp*,0) (-*disp*,0), (0, *disp*), (0,-*disp*) and (0,0). The position with the minimum MAD becomes the centre of the next step.

Step2: The distance (*disp*) is reduced by half.

Step3: Repeat step step1 and step 2 until the distance (*disp*) is one.

Step4: The final step, the block which has the minimum in this step is the best matching block.

After the position with the minimum MAD is found, the motion vector can be also found. The motion vectors indicate where the best matching block locate. The procedure is the same for every candidate blocks. After the motion vector of every blocks are found, the current frame can be reconstructed by using the information of the previous frame plus the motion vector.

7.3.2 Performance of the Diagonal Logarithmic Search (DLS) Algorithm

The DLS performance is simulated and the results of the subjective quality performance are shown in figure (7.13)-(7.15). The simulations are performed for every 10^{th} frame. The constructed frames are shown on the left hand side. As seen in figure (7.13), the subjective quality of DLS is very good as the errors of the predicted frame are unnoticeable. The prediction of DLS has some errors on the face of "Claire" video sequence. However the overall of prediction is good as most of the parts in the frame are predicted accurately. The value of PSNR of 10^{th}, 20th, 30th, 40th predicted frames are very high and higher than 39 dB. The DLS algorithm simulation tested the "Foreman" video sequence and shows that the subjective quality is very good. The subjective quality of predicted frames is comparable with the original frames. Figure (7.15) shows the results of the simulation on Carphone video sequence using the DLS algorithms. It is acknowledged that the movement of carphone is more than that of Claire video sequence as the head of carphone is moving around. The prediction will be more difficult than the other video sequences. The DLS shows that the simulation results achieve are very good for the Carphone video sequence. The prediction is very good as can be seen in figure (7.15) The PSNR of the predicted frames are high and higher than 30 dB. Therefore the DLS algorithm is a good algorithm that can achieve very good prediction.

Test sequence: "Claire"

Original Frame *Predicted Frame*

Frame#10

Frame#10(PSNR=39.37dB)

Frame#20

Frame#20 (PSNR=39.80dB)

Frame#30

Frame#30 (PSNR=40.41dB)

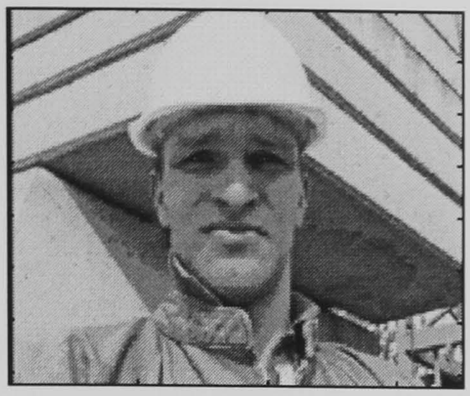
Frame#40

Frame#40(PSNR=45.85dB)

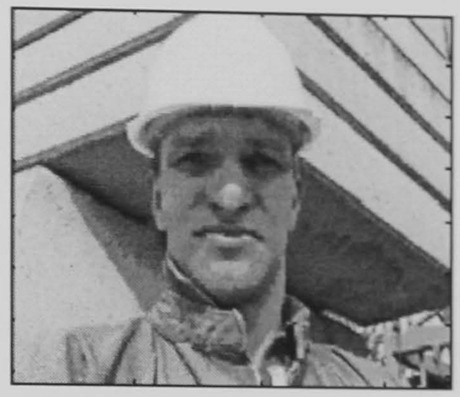
Figure (7.13) "Claire" Subjective quality of the predicted frame using DLS

Test sequence: "Foreman"

Original Frame *Predicted Frame*

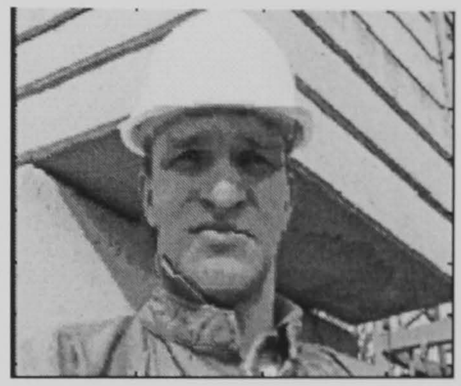


Frame#10

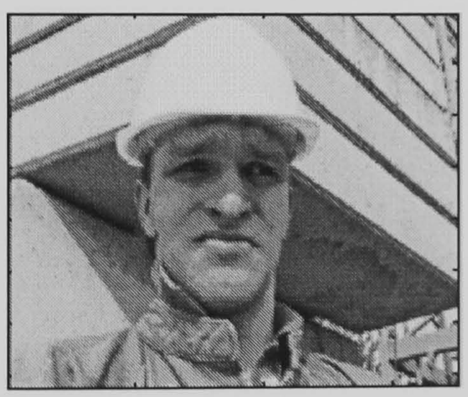


Frame#10(PSNR=31.59dB)

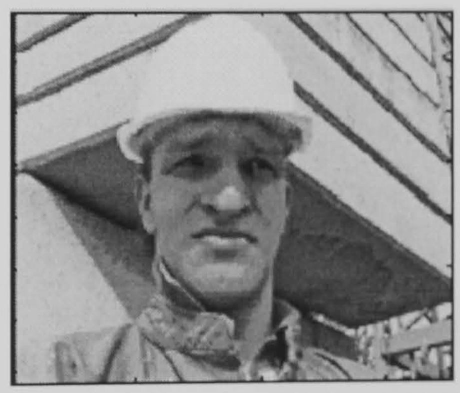
Frame#20



Frame#20 (PSNR=30.42dB)



Frame#30



Frame#30 (PSNR=31.04dB)

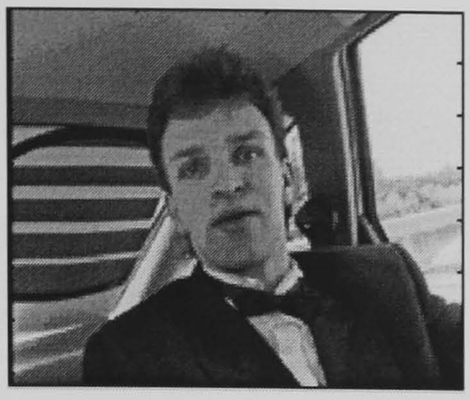
Frame#40

Frame#40(PSNR=33.35dB)

Figure (7.14) "Foreman" Subjective quality of the predicted frame using DLS

Test sequence: "Carphone"

Original Frame *Predicted Frame*

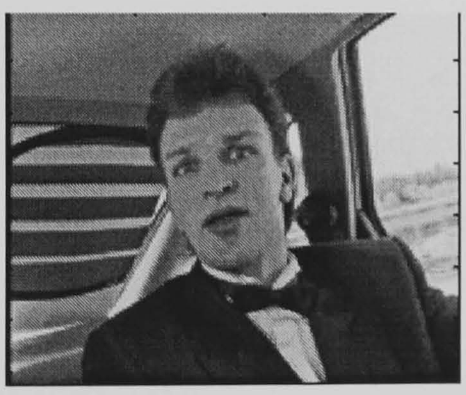


Frame#10

Frame#10(PSNR=32.51dB)

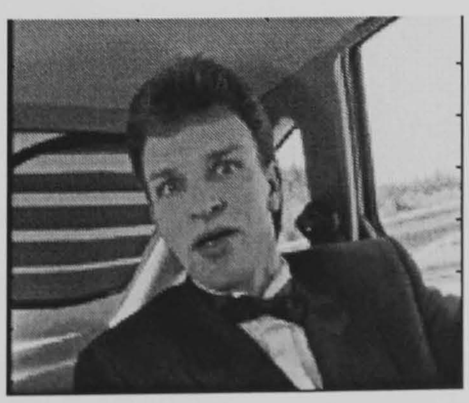
Frame#20

Frame#20 (PSNR=31.29dB)



Frame#30

Frame#30 (PSNR=30.30dB)



Frame#40

Frame#40(PSNR=38.41dB)

Figure (7.15) "Carphone" Subjective quality of the predicted frame using DLS

The performance of the objective quality is shown in figure (7.16)-(7.18). The PSNR between the current frame and predicted frame (PSNR2) of the 90 predicted frames of "Claire" video sequence are shown in figure (7.16). The performance achieves very good prediction. The PSNR2 of all 90 frames are higher than 35 dB. The value is extremely high so the accuracy of the prediction is very high. The performance of algorithm is also tested using "Foreman" video sequence as well as "Carphone" video sequence. The simulation on "Foreman" video sequence achieves very high PSNR for most of the predicted frame. The PSNR2 is higher than 30 dB. In some case the movement between the frames is high, so it causes prediction degradation. In figure (7.18) the prediction of the "Carphone" video sequence are also shown. The PSNR between the current frame and predicted frame is again high and is higher than 30dB. The predictions of the 90 frames of "Carphone" video sequence are accurate. The PSNR2 is ranges between 30dB to 38dB

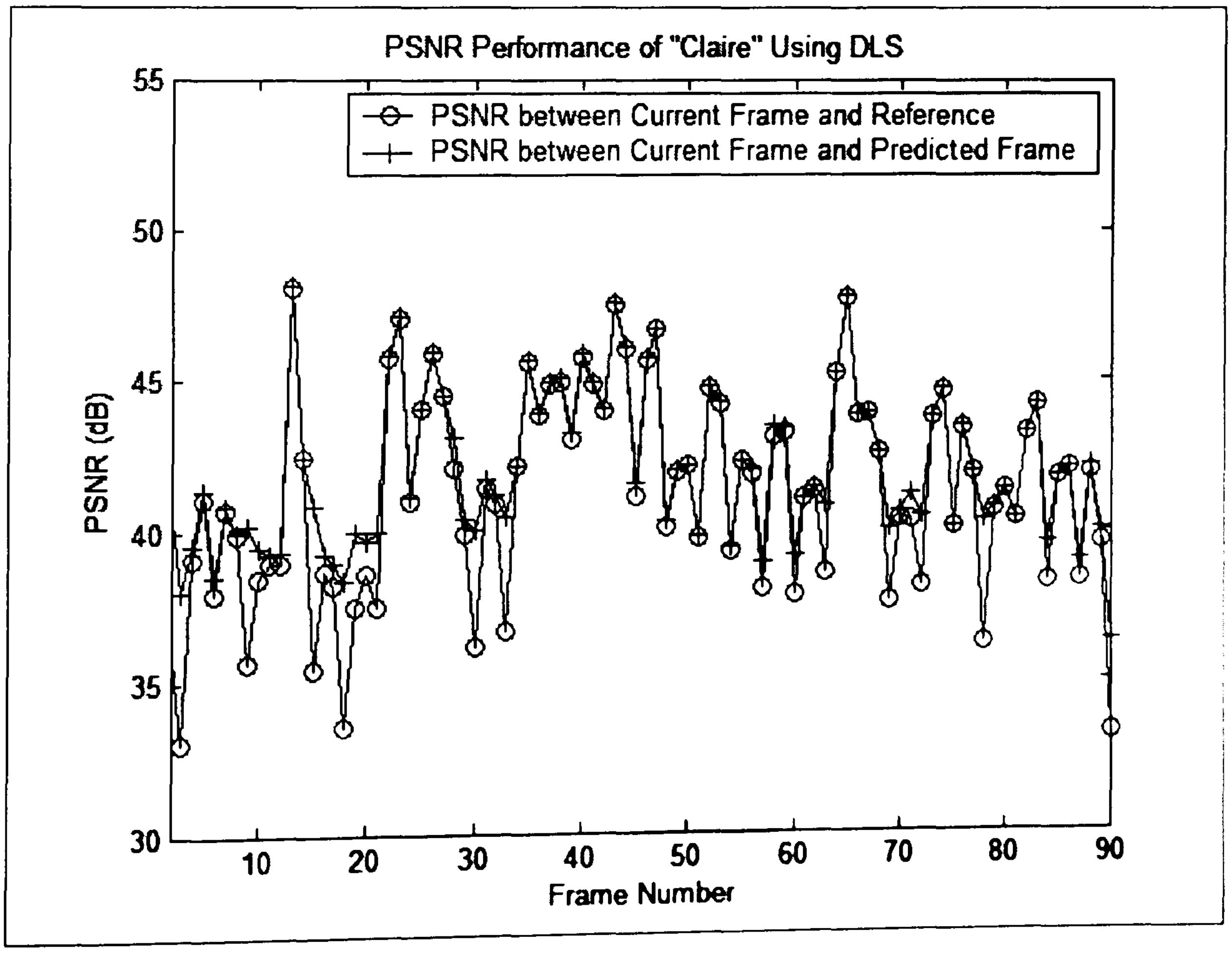


Figure (7.16) The PSNR Performance of "Claire" over 90 frames using DLS

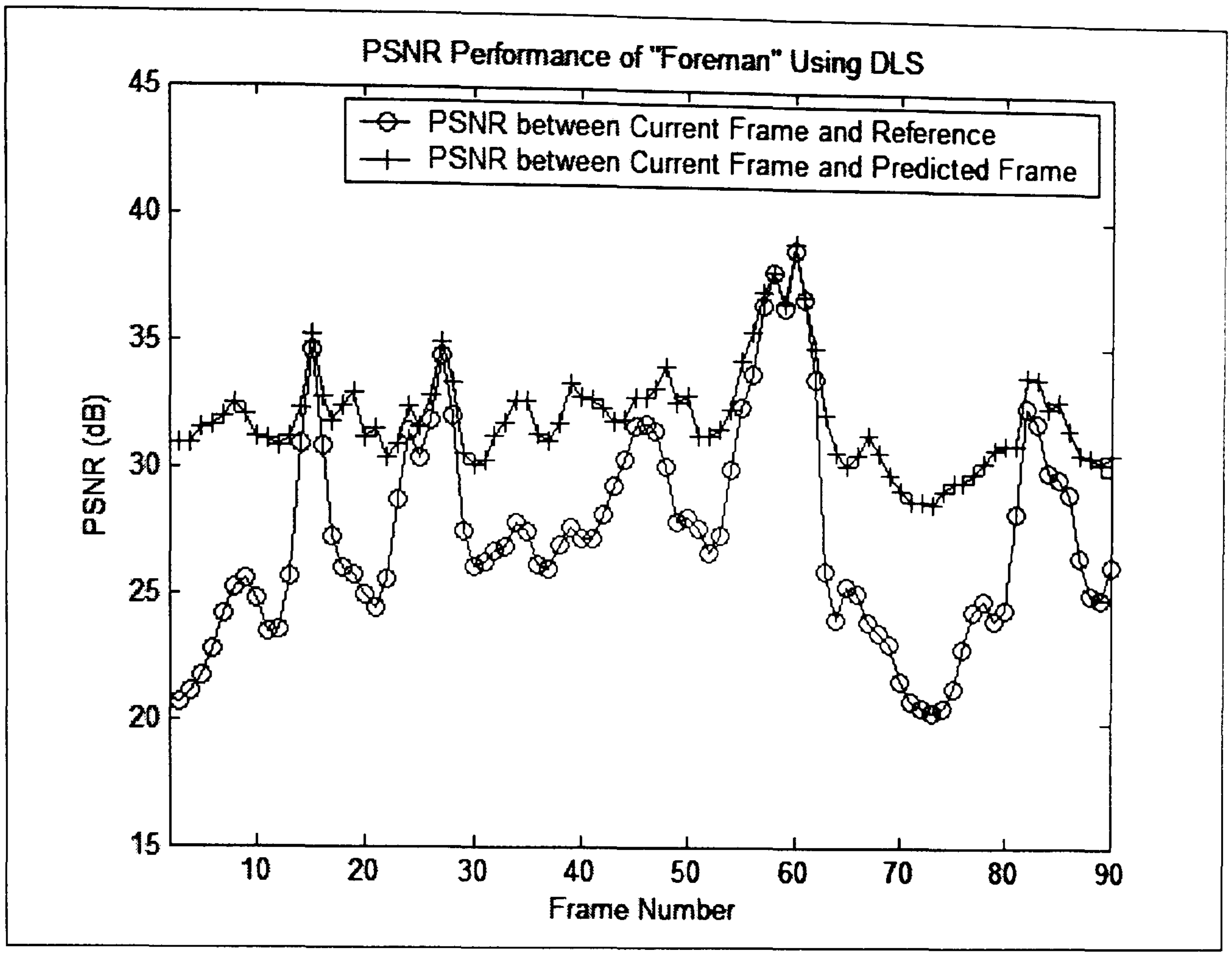


Figure (7.17) The PSNR Performance of "Forman" over 90 frames using DLS

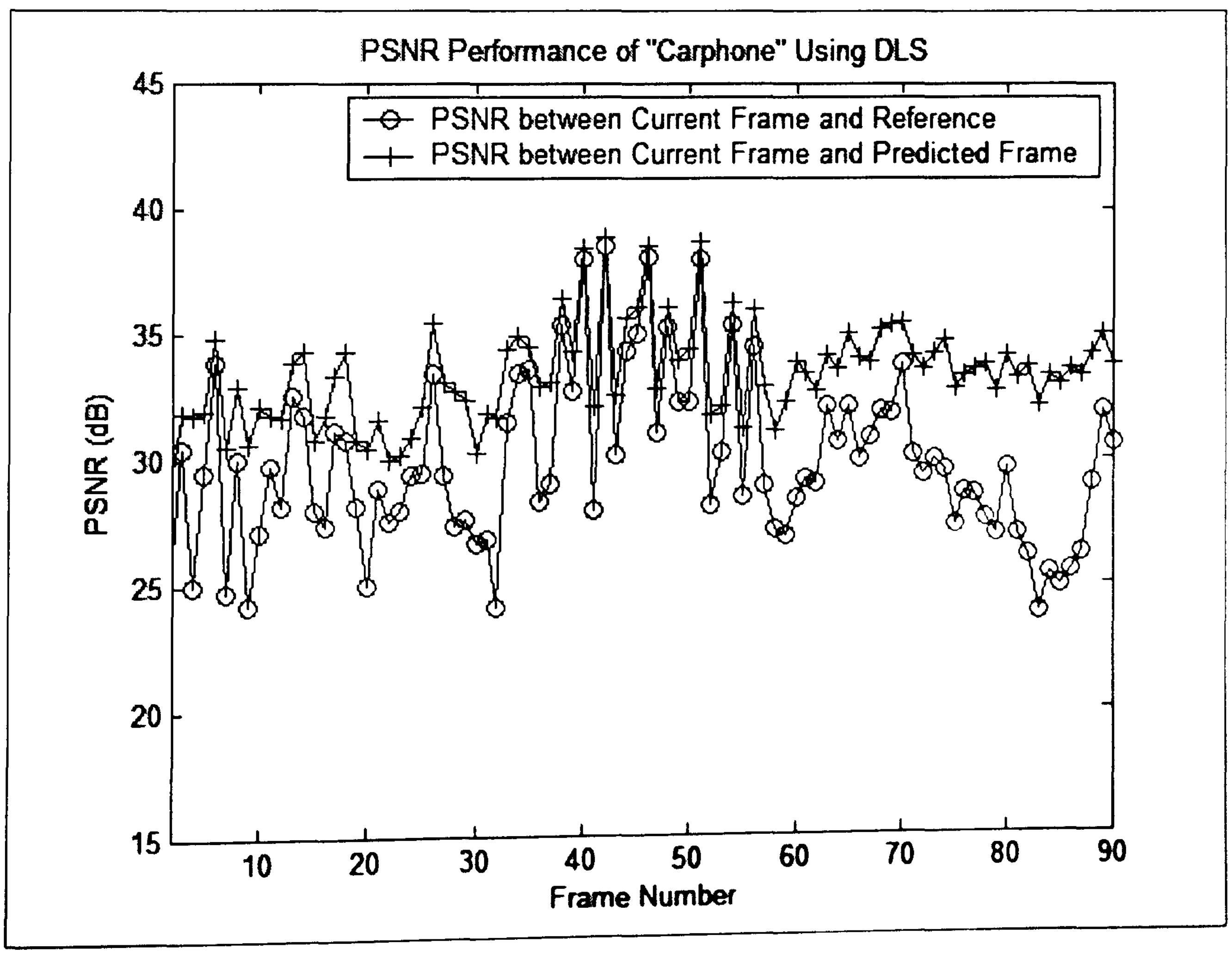


Figure (7.18) The PSNR Performance of "Carphone" over 90 frames using DLS

7.3.3 Comparative Performance of DLS

The DLS algorithm is compared with the 3SS and FS algorithms. The prediction performance is assessed by means of objective PSNR quality. The PSNR performance of "Claire" video sequence is shown in figure (7.19) The PSNR values of prediction using DLS algorithm is as good as that of the 3SS and FS algorithm. As seen from the figure (7.19) the PSNR of all 90 frames using DLS is almost the same values as using 3SS and FS algorithm. The PSNR values of DLS algorithm lie on almost the same line with that of the 3SS and the FS. The PSNR values are also very high. From figure (7.19) the results shown for all three algorithms, FS 3SS and DLS can successfully predict the "Claire" video sequence. The predictions are very accurate because the movement between two consecutive frames is very small. The video sequences that have a greater movement need to be tested as well. The average PSNR over 90 frames of Claire video sequence are 41.66 dB, 41.62 dB and 41.61dB for Full Search, 3SS and DLS respectively.

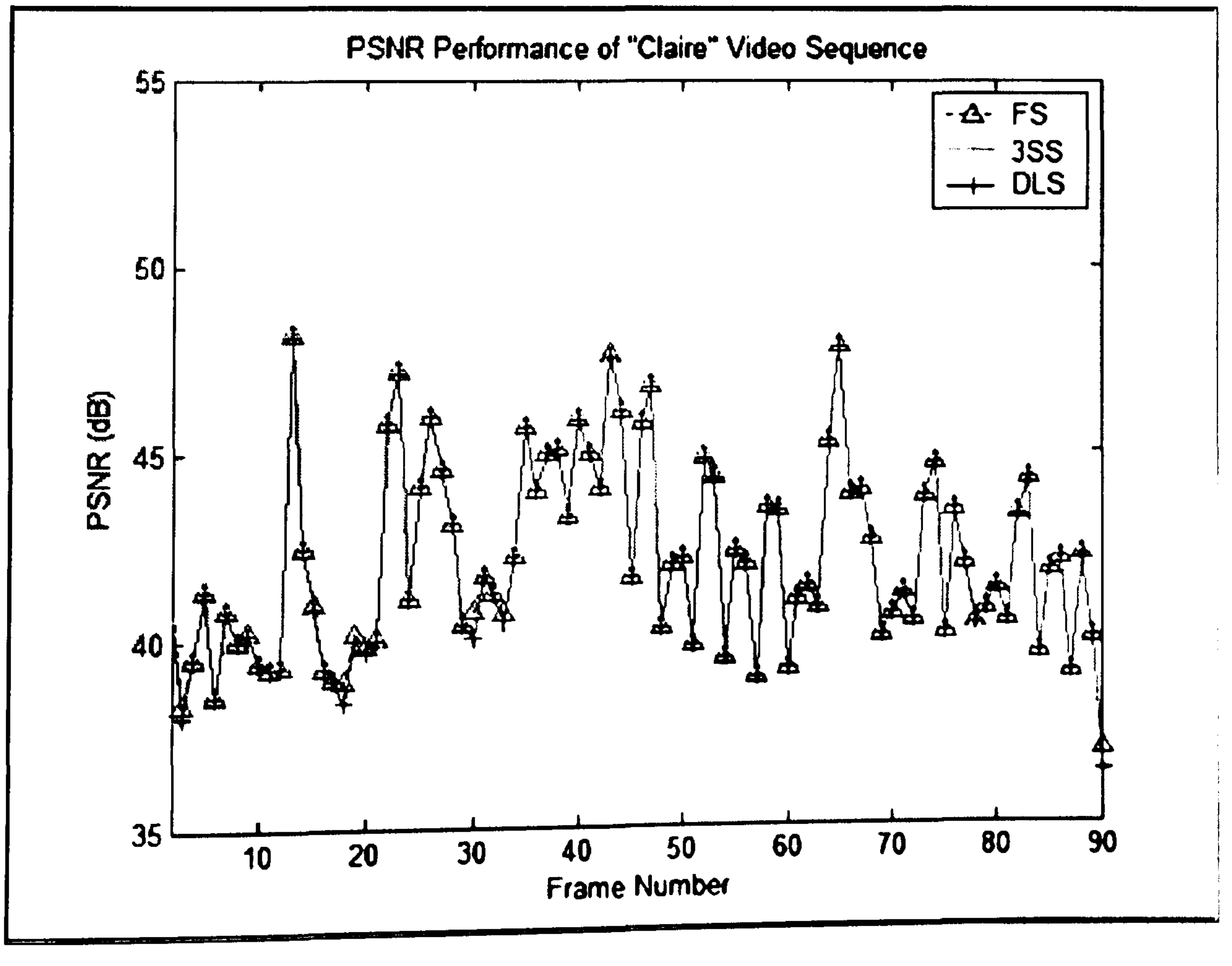


Figure (7.19) Comparative PSNR Performance of "Claire" Using FS, 3SS and DLS

The performance of DLS over "Foreman" and "Carphone" compared with 3SS and FS is shown in figure (7.20) and (7.21). The performance of DLS algorithm is slightly worse than FS algorithm in "Carphone" video sequence but it is better than 3SS. During the 75th frame and the 90th frames, the performance of 3SS drops and is much worse than that of the DLS. The average PSNR over 90 frames of Carphone video sequences are 33.43 dB, 32.82 dB, and 32.98 dB for Full Search, 3SS and DLS respectively. The results of "Foreman" video sequence are shown in figure (7.21). The results show that the DLS algorithms still gives better performance than 3SS. The average PSNR over 90 frames of Foreman video sequence are 32.39 dB, 31.53dB, and 31.68dB for Full Search, 3SS and OLS respectively.

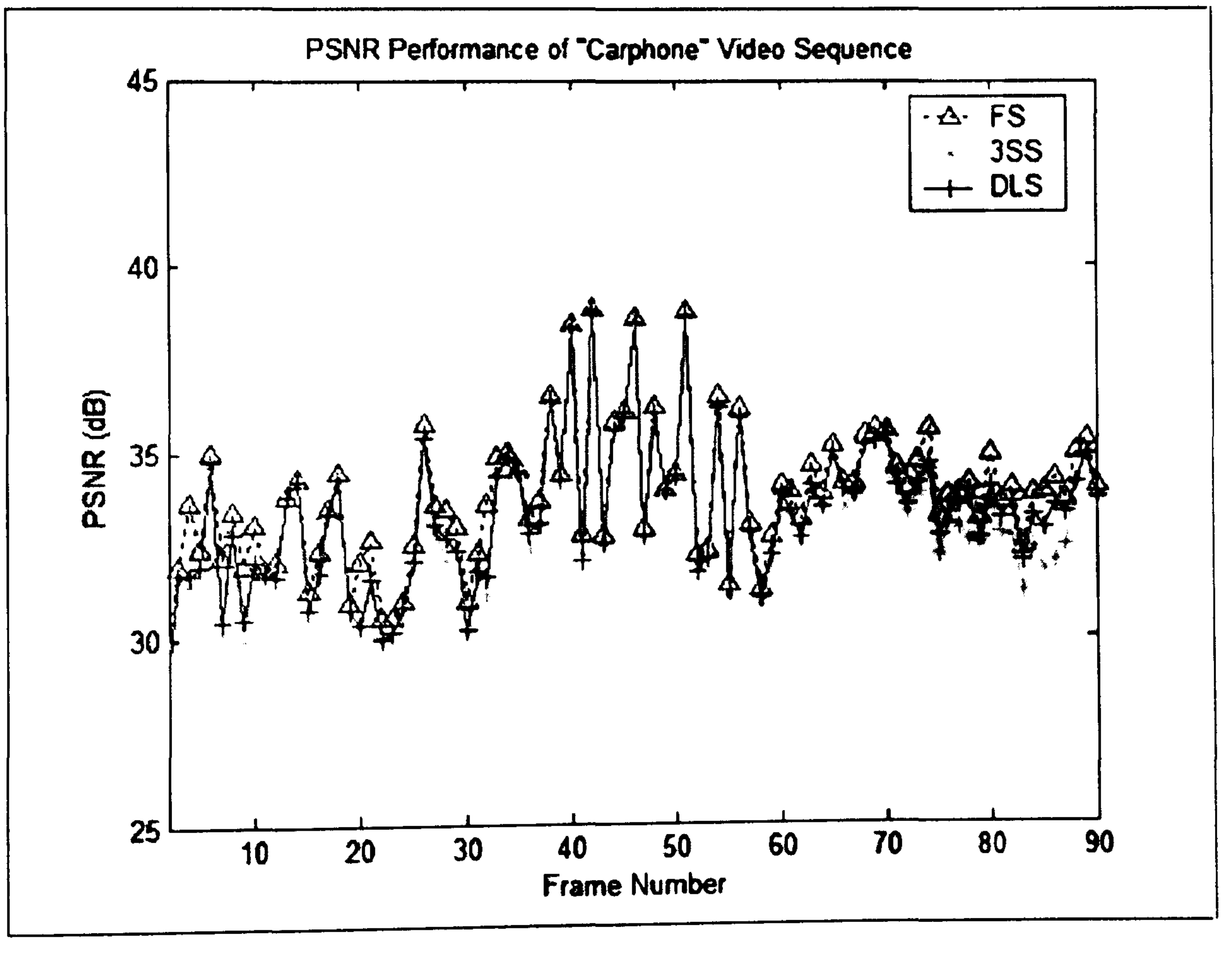


Figure (7.20) Comparative PSNR Performance of "Carphone" Using FS, 3SS and DLS

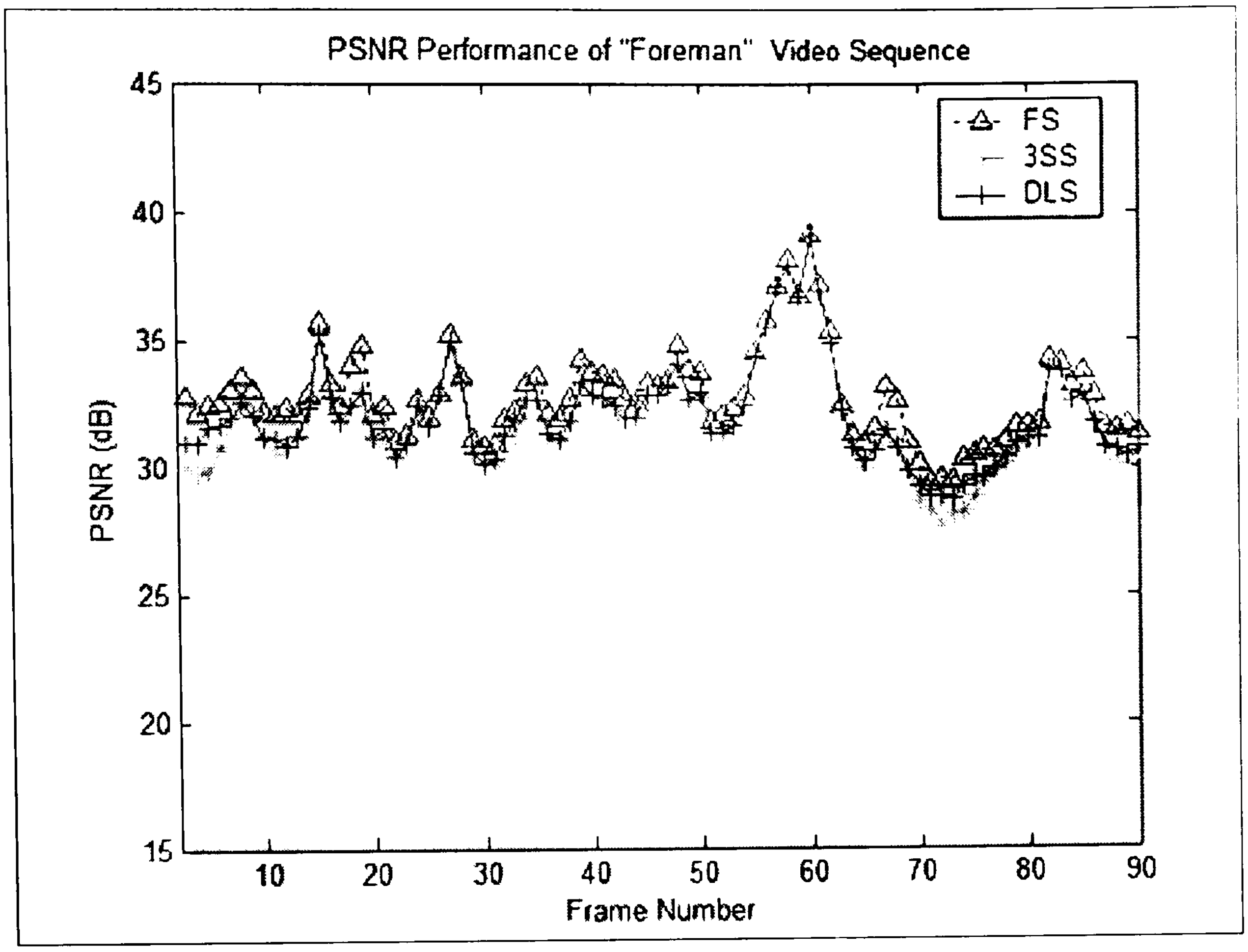


Figure (7.21) Comparative PSNR Performance of "Foreman" Using FS, 3SS and DLS

The advantage of the DLS algorithm lies in the speed of operation. The 3SS algorithm is designed for the improvement of FS in terms of processing time. The main target of the DLS algorithm is also the same. The DLS aims to improve the performance of prediction both in terms of speed and quality. The results in figure (7.22) give the total time which the algorithm spend on the prediction over 90 frames of each video sequences. The result shows the total time of DLS is much better than FS and 3SS. Also the quality of prediction of DLS is better than 3SS.

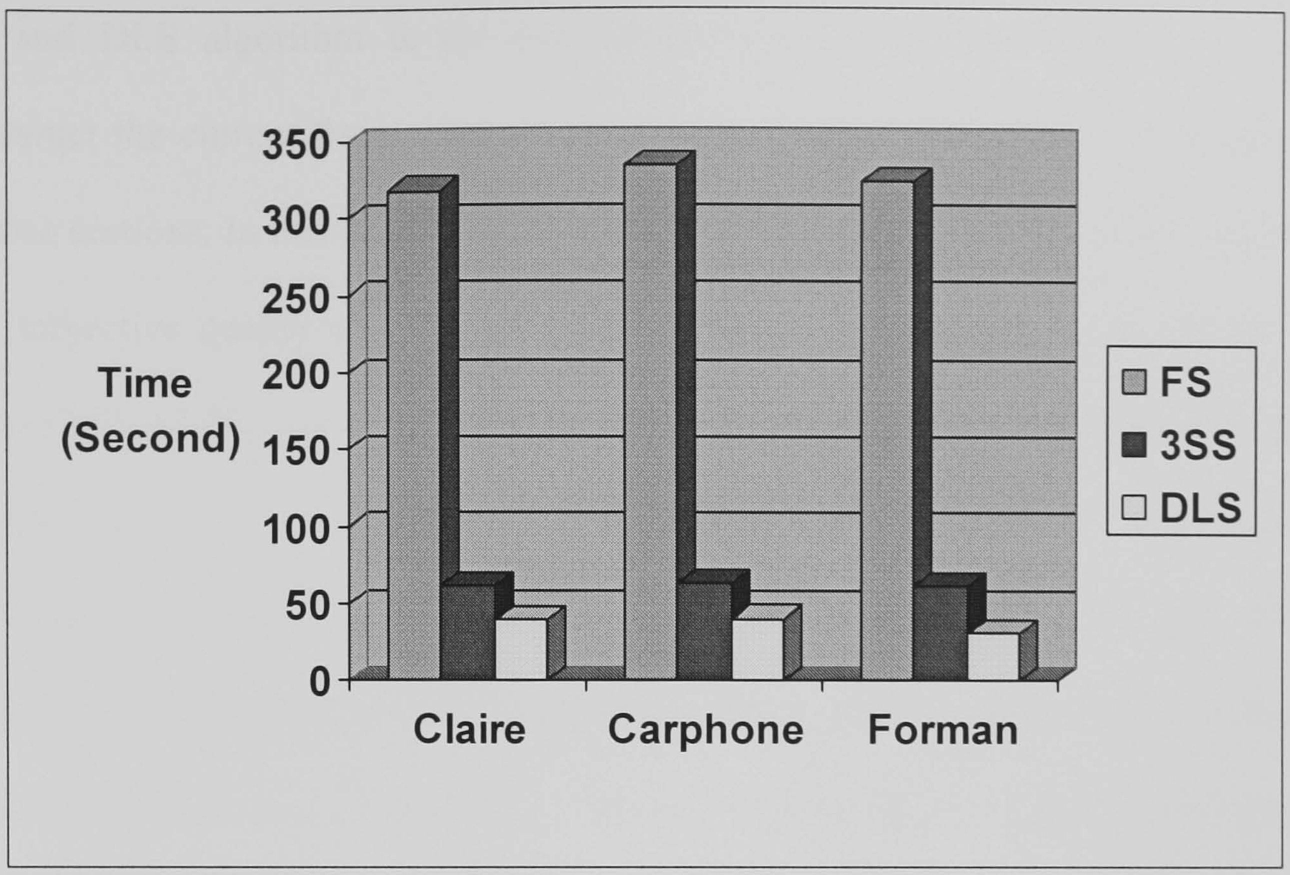


Figure (7.22) Comparative Speed of Operation using FS, 3SS and DLS

7.4 Comparison between OLS and DLS Algorithms

The OLS and DLS algorithms are the novel methods. Both OLS and DLS can improve the performance of search algorithm by reducing the complexity of the algorithms. When the complexity of the algorithms is reduced, the power consumption is also decreased. The OLS and DLS algorithms can be successfully implemented. The results show that the algorithms successfully improve the speed of operations. The speed of the operations is much higher than that of the FS algorithm and the 3SS algorithm. The performance of both OLS and DLS are also acceptable as they achieve very high PSNR. Their PSNR values are as good as that of the FS and the 3SS. In some case, the PSNR is even higher than the 3SS algorithm. Both OLS and DLS base on logarithmic function. The main difference between the OLS and DLS is the search pattern. The procedures of these both algorithms are the same. The algorithms start by the video sequence extracted into frames. These frames are fed into the motion estimation process. The two consecutive frames are compared by using OLS or DLS algorithm. The result from the OLS and DLS algorithm is the best matching blocks. These blocks are used to reconstruct the current frame. The performances of these algorithms are shown in the previous sections. In this section, the comparison between the OLS and DLS are shown. Both subjective quality and objective quality are used for comparison. As shown in figures (7.23)-(7.25) where samples of the results are shown. The subjective quality of the reconstructed frames is shown. The 9th, 19th, 29th and 39th frame of video sequences are chosen as the reference frame. These frames are compared with 10th, 20th, 30th, 40th frame in a row. Having used the OLS and the DLS search pattern, the best matching block are found as well as the motion vectors. The results of "Claire" video sequence show that the OLS and the DLS can successfully predict the current frame as the subjective quality is very good. The subjective quality of the prediction of 20th, 30th and 40th frame are very good as the difference between the predicted frames and original frames is very small. From the figure (7.23), the results show that the quality of the prediction using the DLS algorithm is slightly higher than that of the OLS algorithm. The subjective quality of the OLS algorithm is as good as the OLS algorithm.

The performances of the subjective quality of "Foreman" video sequence using the OLS and the DLS are shown in figure (7.24). The predictions achieve very high performance in the subjective quality as the error is barely discernible for both algorithms. Even though the "Foreman" video sequence considered to be more difficult to predict than "Claire", the OLS algorithm and the DLS algorithm successfully predicted the current frame. From the figure (7.24) the results show that the performance of the OLS algorithm is good as that of the DLS algorithm as the PSNR values is almost the same as that of DLS. The subjective quality of the OLS is also comparable with that of the DLS.

Test sequence: "Claire"

Reconstructed Frames using OLS *Reconstructed Frames using DLS*

Frame#10(PSNR=38.42dB)

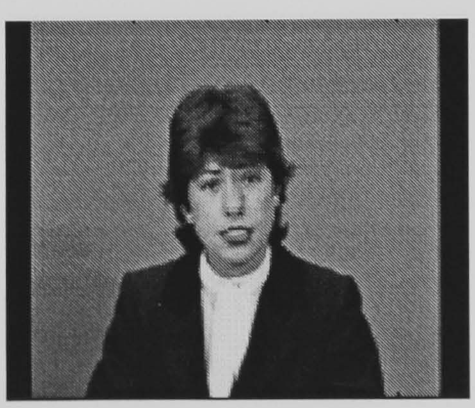
Frame#10(PSNR=39.37dB)

Frame#20(PSNR=38.56dB)

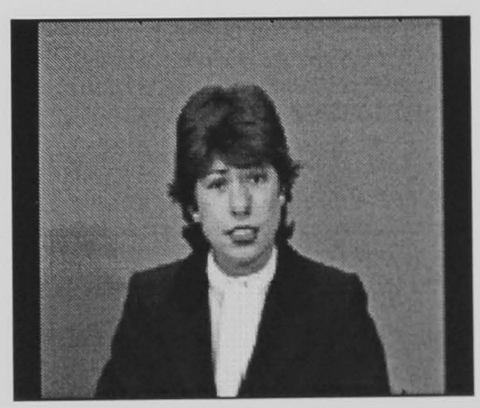
Frame#20 (PSNR=39.80dB)

Frame#30 (PSNR=36.19dB)

Frame#30 (PSNR=40.41dB)



Frame#40(PSNR=45.77dB)



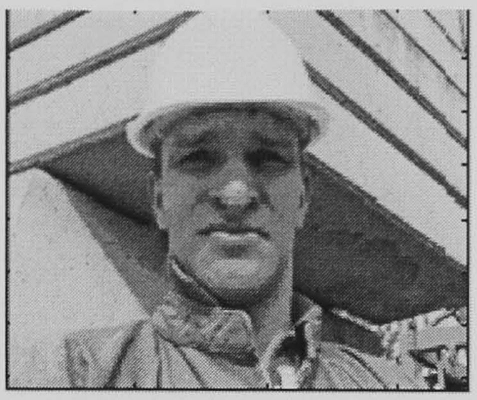
Frame#40(PSNR=45.85dB)

Figure (7.23) "Claire" Subjective quality comparison between OLS and DLS

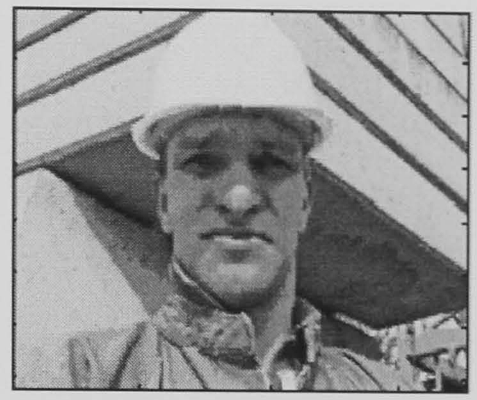
Test sequence: "Foreman"

Reconstructed Frame using OLS

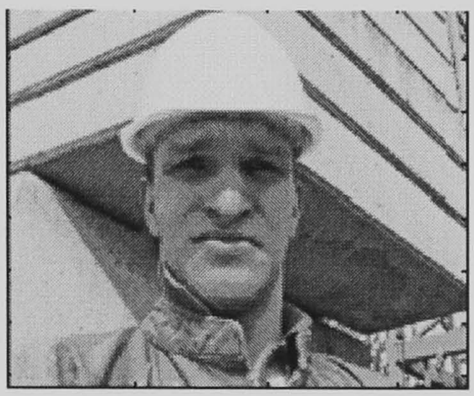
Reconstructed Frame using DLS



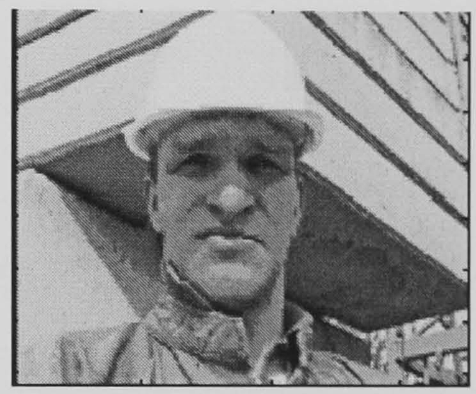
Frame#10(PSNR=31.59dB)



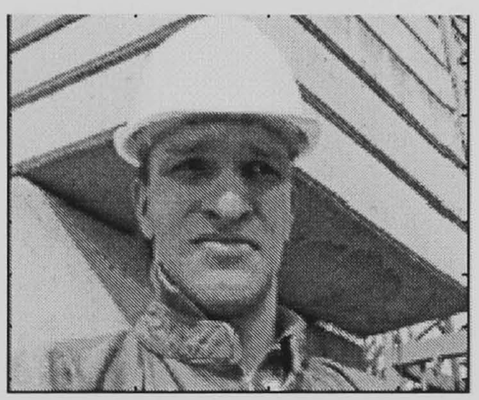
Frame#10(PSNR=31.59dB)



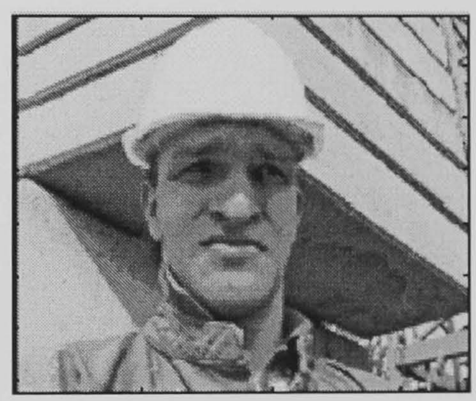
Frame#20 (PSNR=30.42dB)



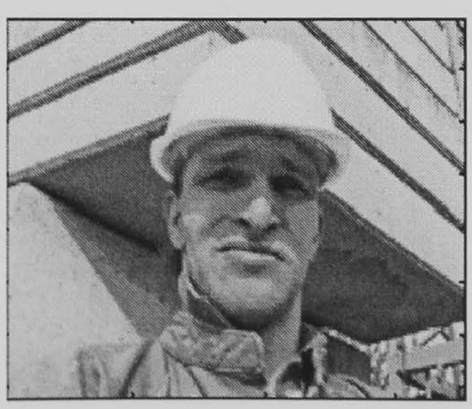
Frame#20 (PSNR=30.42dB)



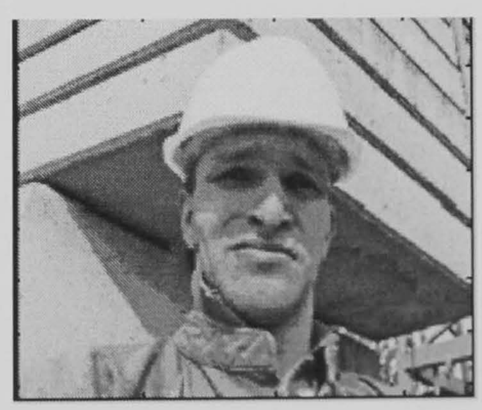
Frame#30 (PSNR=31.03 dB)



Frame#30 (PSNR=31.04dB)



Frame#40(PSNR=33.35dB)

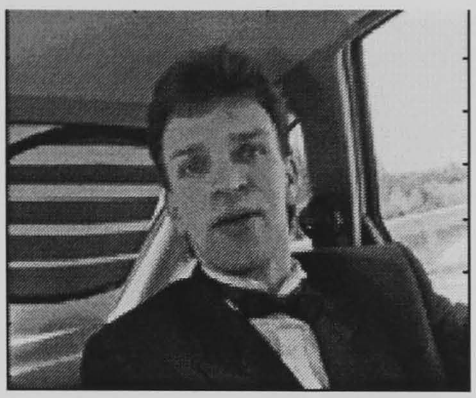
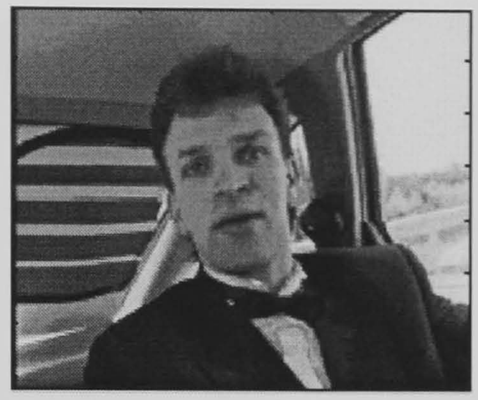


Frame#40(PSNR=33.35dB)

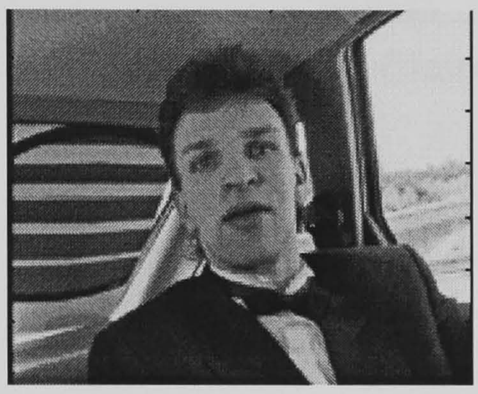
Figure (7.24) "Foreman" Subjective quality comparison between OLS and DLS

Test sequence: "Carphone"

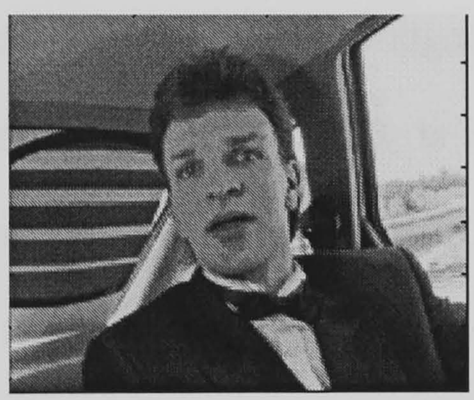
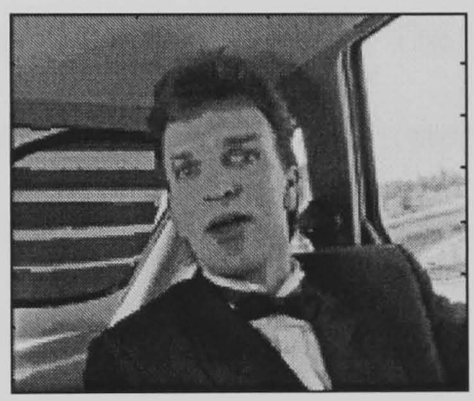
Reconstructed Frame using OLS *Reconstructed Frame using DLS*

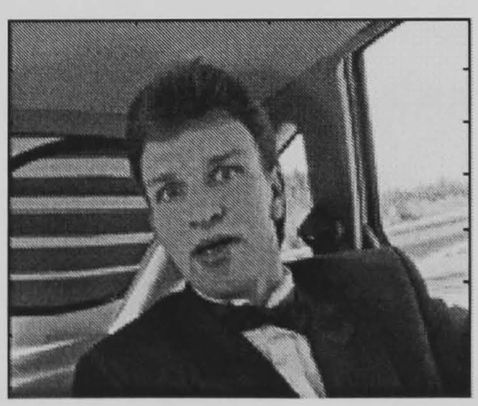
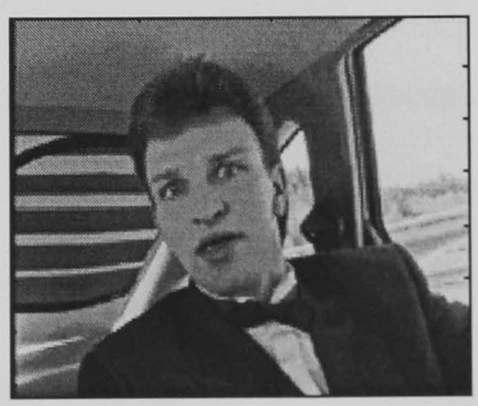
Frame#10(PSNR=32.51dB) Frame#10(PSNR=32.51dB)




Frame#20 (PSNR=31.29dB) Frame#20 (PSNR=31.29dB)

Frame#30 (PSNR=30.30dB) Frame#30 (PSNR=30.30dB)

Frame#40(PSNR=38.41dB) Frame#40(PSNR=38.41dB)

Figure (7.25) "Carphone" Subjective quality comparison between OLS and DLS

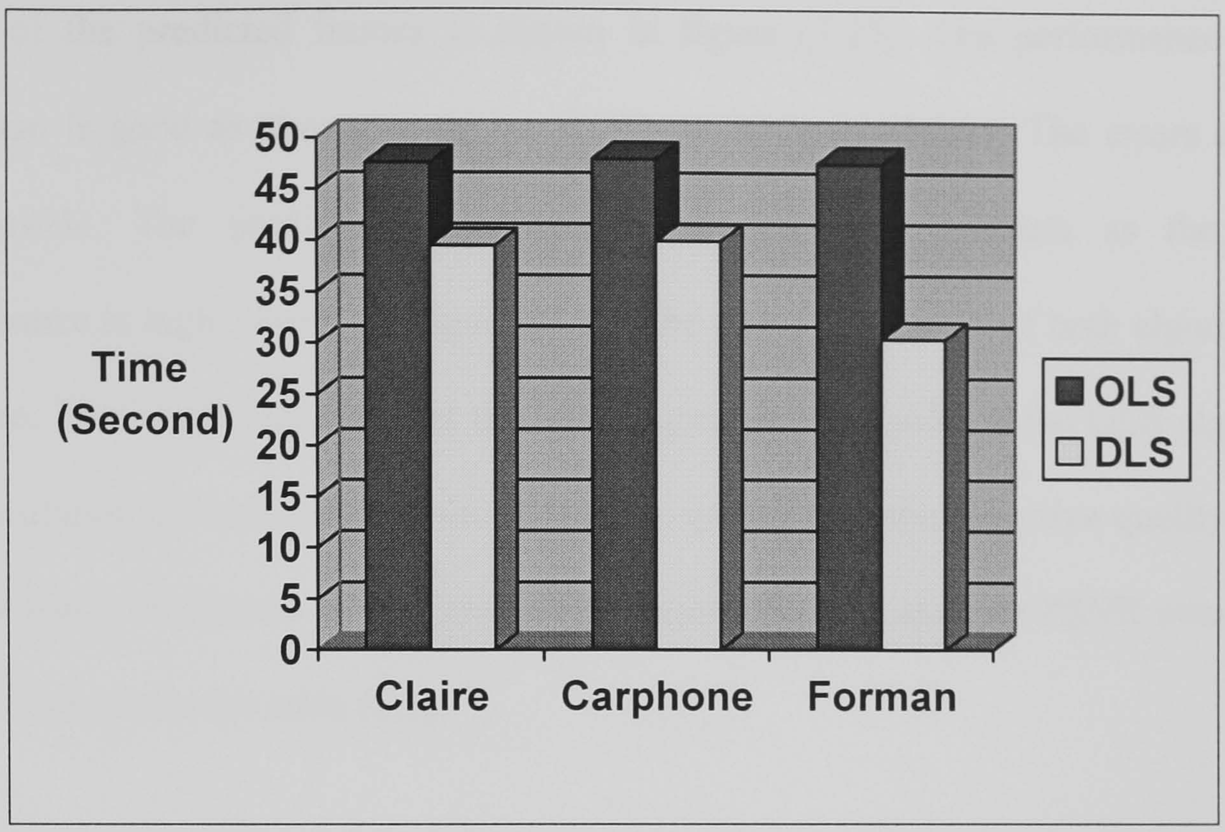


Figure (7.26) Comparative Speed of Operation using OLS and DLS

The advantage of the DLS algorithm lies in the speed of operation. The aims of implementation of the DLS is to design the algorithm which and improve the performance of both the quality of prediction and the speed of operation. The results in figure (7.26) give the total time which the algorithm spend on the prediction over 90 frames of each video sequences. The result shows the total time of DLS is much better than OLS. The processing time of the OLS algorithm is 47.60 seconds, 47.98 seconds and 47.57 seconds for Claire, Carphone and Foreman respectively. The processing time of the DLS algorithm is 39.50 seconds, 39.95 seconds and 30.43 seconds for Claire, Carphone and Foreman. The results show that the speed of the DLS algorithm is faster than the OLS algorithm. The processing time of the DLS algorithm is less than that of the OLS at all cases.

The "Carphone" video sequence was also tested in a similar way. A sample of the results of the predicted frames is shown in figure (7.25). The performance of the prediction is good as shown in figure (7.25) for both algorithms. The errors also are unnoticeable. The prediction is considered to be quite accurate as the PSNR performance is high. From the figure (7.25), the objective quality of both algorithms is the same. The results indicate that the DLS algorithm is as good as the OLS algorithm. The simulation did not only test the subjective quality, but the objective quality is also tests by using 90 frames of the three video sequences. The average PSNR over the 90 frames is shown in the table (7.1).

	Average PSNR (dB)		
	"Claire"	"Foreman"	"Carphone"
OLS	41.63	31.51	32.88
DLS	41.61	31.68	32.98

Table (7.1) Comparison of the average PSNR using OLS and DLS

From the table (7.1), the average PSNR of the OLS algorithm is better than the DLS algorithm on the "Claire" video sequence. However the average PSNR values using "Clair" video sequence of both algorithms is not a big difference. The difference of PSNR on "Claire" video sequence is only 0.02 dB. The simulation of the "Foreman" and "Carhone" are also tested over the 90 frames of video sequence. The average PSNR of "Foreman" and "carphone" using the OLS and the DLS are shown in table (7.1). The DLS algorithm is proved to be better than the OLS on the "Foreman" and "Carphone" video sequence as the average PSNR of the DLS algorithm is higher than that of OLS algorithm.

7.5 Summary

These two novel methods, OLS and DLS can improve the performance of the processing times and still remain the good performance of the quality of the reconstructed images. The algorithms are simulated by using MATLAB. The well-known video sequences were tested. The two main metric, time and Peak Signal to Noise ratio are used to evaluate the algorithms. The simulation is carried on the 90 frames of the video sequence. The average PSNR of each algorithm is summarised in Table (7.2). In addition, the subjective quality is assessed. The intention of subjective quality assessment is to support the results of PSNR.

	Average PSNR (dB)			
	Full Search	3SS	OLS	DLS
Claire	41.66	41.62	41.63	41.61
Carphone	32.39	31.53	31.51	31.68
Foreman	33.43	32.82	32.88	32.98

Table (7.2) Summary of the average PSNR using FS, 3SS, OLS and DLS

From Table (7.2), the results is shown that the quality of prediction using FS algorithm is the best as the average PSNR is highest. However the computational complexity of the FS algorithm is high. So the FS algorithm is not normally used in the real-time application. The 3SS algorithm is designed to solve this problem and the 3SS algorithms still remain the most favourite algorithm in the well-known standards. Our algorithms are even better than 3SS algorithm. From the table (7.2) the average PSNR of the OLS algorithm is 41.63 dB, 31.51dB and 32.88 dB for Claire, Carphone and

Foreman video sequence. The average PSNR of the OLS algorithm is 41.62 dB, 31.53dB and 32.82 dB for Claire, Carphone and Foreman video sequence. The average PSNR of both the 3SS and the OLS is not big difference. The quality of prediction of the OLS is good as that of the 3SS. But the main advantage of the OLS algorithm is the speed of operation. The time consuming of the OLS algorithm is shown in table (7.3). From the table (7.2) the average PSNR of the DLS algorithm is 41.61 dB, 31.68dB and 32.98 dB for Claire, Carphone and Foreman video sequence. The average PSNR of both the DLS is higher than the 3SS and the OLS algorithms on the "Carhone" and "Foreman" video sequence. In the case of the "Claire" video sequence, eventhough the PSNR is slightly less than the 3SS, the difference is very narrow.From the performance of the prediction, we can conclude that the performances of the novel algorithms are comparable or even better in some case than the conventional FS algorithm and the 3SS algorithm. But the main advantage of both algorithms lies in the speed of the operation. The processing speed of the algorithms are summarised in the table (7.1)

	Time Consuming (Seconds)			
	Full Search	3SS	OLS	DLS
Claire	318.60	61.38	47.60	39.50
Carphone	334.53	62.43	47.98	39.95
Foreman	325.09	61.89	47.57	30.43

Table (7.3) Comparison of the Processing Time

Using the benchmark QCIF video sequences the results show that the average speed of operation for the OLS it is 47.72 seconds, for the DLS it is 36.63 seconds, for the 3SS it is 61.9 seconds and the FS it is 326.07. Consequently, the strength of the OLS algorithm lies in its speed of operation as it is 85.37 % faster than the FSA and over

22% faster than the 3SS. Also the processing time of DLS is 88.77 % faster than the FS and over 40% faster than the 3SS. The speed of operation for DLS is even better than that of OLS. The processing time of DLS is 1.7 % faster than the OLS.

Having considered the results of the simulation of both time and Peak Signal to Noise Ratio, we can conclude that the OLS and the DLS are the better alternatives to the FSA since the quality of the reconstructed frames of the OLS and DLS are comparable with that of the FS method and the 3SS and the speed of operation for the OLS and DLS are faster than that of the FS and the 3SS.

Chapter 8

CONCLUSIONS AND FUTURE WORK

8.1 Conclusions

Motion compensation is an effective technique to reduce temporal redundancy between two consecutive frames of a video sequence. The basic idea behind motion compensation is to estimate the displacement of objects. The methods used are called motion estimation (ME). The ME methods can be classified into 3 types which are optical flow equation based methods, Pixel-recursive methods and block matching methods. The goal of motion estimation is to obtain a prediction from the reference frame for the current frame such that the prediction error is minimised. Block matching is an effective technique for motion estimation owing to its regular data structure and simpler implementation. The block matching method can achieve a good balance between computational complexity and coding efficiency. Novel block matching algorithms are proposed for fast motion estimation. All these algorithms work for the same goal which is to speed up the motion estimation while keeping the motion estimation accuracy as high as possible. The block matching is widely used by the well-known the international standard such as MPEG and H.263. In this thesis, a typical motion compensation scheme has been presented. This scheme is suitable for the 176×144 pixel QCIF videophone sequences. The general fundamental theory of video coding is explained in Chapter 2. The well known video coding both for spatial redundancy and temporal redundancy reduction are explained.

In chapter 4, the basic ideas of motion estimation are presented. The problems of motion estimation and the solution for these problems are explained. These solutions lead to the motion estimation methods called Optical Flow. In this chapter, the main approaches for motion estimation are summarised such as Optical flow, Pixel-recursive and Block Matching. In block matching techniques the search algorithm are used to obtain the best block matching. There are a number of search algorithms that have been proposed. The conventional FS algorithm is considered to be the best performance in the prediction. However, the FS algorithm requires very high computational complexity. So the fast search algorithms are proposed to reduce the computation complexity. The well-known three step search is also described. Another two fast search algorithm, N3SS and OTA, are investigated. The disadvantages of the new three step search are highlighted. The comparison of the performance of the three step search and new three step search are shown. The OTA technique is also investigated. However this method is considered to be worse than the 3SS. The result of prediction using OTA technique is shown. From these four algorithms, the 3SS is shown to have the best performance both for the quality and processing times. So the 3SS is investigated thoroughly in chapter 6. In chapter 6, the FS algorithm was also investigated because it is the conventional technique which achieves very good PSNR performance.

In chapter7, the novel OLS and DLS algorithms are implemented. The performances of both algorithms are investigated. There are 3 main groups of techniques which try to improve the performance of the search algorithm which are:

1. those that reduce the candidate blocks for searching motion vectors
2. those that reduce the calculated pixels for computing the distortion measure
3. those that reduce the current blocks for employing block matching.

The implementation of algorithms, the cost function or matching criteria is chosen by considering the performance of the well-known matching criteria, MSE, MAD, NCF and NTD. The simulation shows that the cost function MAD is as good as MSE and NCF but the computation complexity is the best among them. So the novel OLS and DLS algorithms in this thesis use the MAD as the matching criteria. The novel algorithms improve the performance by reducing the number of searching point. The numbers of the search points of the new algorithms are less than the three step search. The number of search points of OLS and DLS is 13 and 12 respectively while the number of search points of 3SS is 25.The novel OLS and DLS is proved to be very good choice and better than the 3SS. The performance of prediction is very good both in terms of subjective and objective quality. The performance of prediction of DLS is better than 3SS as the PSNR of reconstructed frame is improve approximately 0.06 dB over 3SS. The advantage OLS and DLS algorithm lie in the speed of operation.

The speed of OLS and DLS is extremely fast as it is 85.37 % faster than the FSA and over 22% faster than the 3SS for OLS. For DLS, the speed of operation is 88.77 % faster than the FSA and over 40% faster than the 3SS. The speed of the DLS is even faster than that of the OLS as the processing time of DLS is 1.7 % faster than the OLS The system such as the mobile phone communications which the bandwidth and time is limited the DLS and OLS is better choice than the 3SS and FS.

8.2 Future Work

Throughout this research, the block matching technique has been the main issue since it is the most popular technique adopted by international standards. The aim of this research has been to design the novel techniques which are better than the methods adopted by international standard such as MPEG and the H263. The objectives of this thesis are realised by the implementations of the OLS algorithm and DLS algorithm.

They are both better than the 3SS search which is used in MPEG Standard. The speed of OLS and DLS is much faster than FS algorithm. The research considered only the luminance component of the video sequences because the motion vectors can be predicted by only using luminance component. However the real world video sequence is colour, thus this research can be simulated in the extension of Chrominance component.

Owing to the fast movement of the technology the current mobile communication systems are much better than the communications systems in the last decade. The current mobiles systems have already added the service that can send and receive multimedia information such as image and video. However the real-time multimedia mobile communication system is still not very efficient since the bandwidth is very limited. The mobile communication system is used in the MPEG-4 standards to send and receive video or image data. Instead of using the 3SS of MPEG4 standards these novel algorithms can be applied and would be a benefit to assist the transmission of the high amount of multimedia data. Future research can concentrate on the application of the novel search algorithms in mobile systems. The OLS and DLS are a good guide for future research as they are very fast and robust algorithms. Moreover these two algorithms can be put forward to be tested with ITU standards. They can be considered as alternative algorithms for ITU standards. The video size of mobile communication is as the same size and format as the QCIF video. Therefore, the time complexity of the motion estimation will be reduced for the overall system. However future mobile systems will be able to send and receive larger sizes of picture formats such as CIF (Source Input Format).The OLS and DLS algorithm are an extremely good base to develop new motion estimation algorithms that can improve both compression and prediction in fast processing times. These metrics of data management and processing speed are very important for future mobile systems.

REFERENCES

[1] http://www.forum.nokia.com/seap/documents

[2] G. H. Barry, P Atul and N. N. Arun (1997) "Digital video: an introduction to MPEG-2" *Chapman & Hall*, New York.

[3] H. Roy (1997), "Data compression in digital Systems." *Chapman & Hall*, New York.

[4] A. K. Jain (1989), "Fundamentals of digital signal processing" *Prentice-Hall*

[5] J. Zhang, M. O. Anmad and M.N.S. Swamy (1998) "New windowing techniques for variable-size block motion compensation." *IEE Proceeding on Vision Image and Signal Processing,* vol. 145, pp. 399-407

[6] T. Y. Kuo and C. C. J. Kuo (1998). "Fast overlapped block motion compensation with checkerboard block partitioning." *IEEE Transactions on Circuits & Systems for Video Technology*, vol. 8, pp. 705-712

[7] L. Hsu and K. R. Rao (1998). "Wavelet and Lapped Orthogonal Transforms with overlapped motion-compensation for multiresolution coding of HDTV." *IEEE Transaction on Circuits and Systems II-Analogue and Digital Signal Processing,* vol. 45, pp. 1002-1014

[8] Y. Q. Shi, S. Lin and Y. Q. Zhang (1998). " Optical flow-based motion compensation algorithm for very low-rate video coding." *International Journal of Imaging Systems and Technology,* vol. 9, pp.230-237

[9] T. H. Kweh, F. Eryurtlu and A.M. Kondoz (1997). "Closed-loop motion compensation for video coding standards." *IEE Proceeding on Vision Image and Signal Processing,* vol. 144, pp. 227-232

[10] R. Rajagopalan, E. Feig and M. T. Orchard (1998). "Motion optimization of ordered blocks for overlapped block motion compensation." *IEEE Transactions on Circuits & Systems for Video Technology*, vol. 8, pp. 119-123

[11] J. J. Zhang, M. O. Ahmad and M. N. S. Swany (1998). "Bidirectional variable size block motion compensation." *IEE Electronics Letters*, vol.34, pp. 52-53

[12] C. C. Lien, C. L. Huang and J. G. Chen (1997). "Complex-subband transform for subband-based motion estimation/compensation and coding." *IEEE Transaction on Image Processing,* vol. 6, pp.694-702

[13] M. J. Chen., L. G. Chen and R. M. Weng (1997). "Error concealment of lost motion vectors with overlapped motion compensation." *IEEE Transactions on Circuits & Systems for Video Technology*, *vol.* 7, pp.560-563.

[14] H. Jozawa, K. Kamikura, A. Sagata, H. Kotera and H. Watanabe (1997). "Two-stage motion compensation using adaptive global MC and local affine MC." *IEEE Transactions on Circuits & Systems for Video Technology*, vol. 7, pp. 75-85.

[15] E. Francois, J. F. Vial and B. Chupeau (1997). "Coding algorithm with region-based motion compensation" *IEEE Transactions on Circuits & Systems for Video Technology,* vol. 7, pp. 97-108.

[16] X. Lee and Y. Q. Zhang (1996). "A fast hierarchical motion-compensation scheme for video coding using block feature matching." *IEEE Transactions on Circuits & Systems for Video Technology*, vol. 6, pp. 627-635

[17] Y. Yokoyama, Y. Miyamoto and M. Ohta (1995). "Very low bit rate video coding using arbitrarily shaped region-based motion compensation." *IEEE Transactions on Circuits & Systems for Video Technology*, vol. 5, pp. 500-507

[18] T. Sikora (1997). "MPEG Digital Video Standards." *IEEE Signal Processing Magazine*, pp.82-100

[19] http://www.mpeg.org/MPEG

[20] L. Guan, S. Y. Kung and J. Larsen (2000), "Multimedia Image and Video Processing" *CRC Press*, Florida

[21] http://www.nta.no/brukere/DVC

[22] ITU-T (1995) "Video Coding for Low Bit-Rate Communications." *Recommendation H.263-Draft Version.*

[23] A. Gresho and R. M. Gray (1992), "Vector quantization and Signal Compression", *Kluwer Academic Publications.*

[24] M. Rabbani, P. W. Jones (1991), "Digital Image Compression Techniques." *Tutorial Tesxts in Optical Engineering*, SPIE

[25] N. S. Jayant and P. Noll (1984), "Digital Coding of Waveforms', *Prentice-Hall Signal Processing Series*, pp. 146-148

[26] D. Huffman (1962), "A Method for The Construction of Minimum Redundancy Codes", *Proceeding of IRE* , pp. 1098-1101

[27] R. Steinmetz (1994), "Data Compression in Multimedia Computing - Standards and Systems,"Part I and II, *Journal of Multimedia Systems*, vol. 1, pp. 166-172 and 187-204.

[28] G. LeGall (1991), "MPEG: A Video Compression Standard for Multimedia Applications," *Communications of the ACM,* vol. 34, No. 4, pp. 45-68.

[29] W. B. Pennenbaker and J.L. Mitchell (1993), "JPEG Still Image Data Compression Standard," *Van Nostrand Reinhold*, New York,

[30] R. Aravind, G.L. Cash, D.C. Duttweller, H.-M. Hang, and A. Puri (1993), "Image and Video Coding Standards," *AT&T Technical Journal*, vol. 72, pp. 67-88

[31] B. Furht, J. Greenberg, and R. Westwater (1997), "Motion Estimation Algorithms for Video Compression," *Kluwer Academic Publishers*, Norwel, MA

[32] D. Y. Pen (1993), "Digital Audio Compression." *Digital Technical Journal*, vol. 5, No. 2, pp. 28-40

[33] V. Bhaskaran and K. Konstantinides (1995), "Image and Video Compression Standards–Algorithms and Architectures," *Kluwer Academic Publishers*, Norwell

[34] J. L. Mitchell, W.B. Pennenbaker, C.E. Fogg, and D.J. LeGall(1996), "MPEG Video Compression Standard," *Chapman & Hall*, New York

[35] H. Chavari, M. Mills (1990). "Block Matching Motion Estimation Algorithms-New Results" *IEEE Transactions on Circuits & Systems for Video Technology*, vol.37, pp 649-651

[36] L. Scargall, S. S.Dlay (1998), "Mobile Videophone Systems for Radio Speech Channels", *Proceeding Of SPIE'98*, pp. 170-179

[37] A. Netravali and J.D.Robbins (1979), "Motion compensated television coding part I", *Bell Syst. Tech. Journal, vol. 58(no. 3)*, pp. 629-668.

[38] A.J. Jerri (1977), "The shannon sampling theorem - its various extensions and applications: A tutorial review" . *Proceedings of IEEE: Vol. 65(11)*, pp. 1565-1596

[39] A. Verri and T. Poggio (1989), "Motion field and optical flow: qualitative properties" *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 490-498

[40] B. Horn and B. Schunck(1981), "Determining optical flow" *Artificial Intelligence*, (17), pp. 185-203

[41] Telecommunication Standardization Sector of ITU. Draft ITU-recommendation h.263. *ITU Recommendations*, July 1995.

[42] D. Le Gall. Mpeg (1991), "A video compression standard for multimedia applications" *Communications of the ACM*, vol. 34, pp.46-58

[43] F. Pereira, K. O'Connell, R. Koenen, and M. Etoh(1997), "Special issue on mpeg-4, part 1: Invited papers" S*ignal Processing: Image Communication Vol. 9(4)*

[44] T. Y. Kuo and C.C.J. Kuo (1998), "Fast Overlapped Block Motion Compensation with checkerboard block partitioning" *IEEE Transactions on Circuits & Systems for Video Technology*, vol. 8, pp.705-712

[45] J. R. Jain and A. K. Jain (1981), "Displacement measurement and its application in interframe image coding", *IEEE Transaction on Communications.*, vol. 29, pp. 1799-1806

[46] M. Ghanbari (1990), "The cross-search algorithm for motion estimation". *IEEE Transaction on Communications.*, vol. COM-38, pp. 950-953

[47] R. Li, B. Zeng, and M. L. Liou(1994), "A new three-step search algorithm for block motion estimation", *IEEE Transactions on Circuits & Systems for Video Technology*, vol. 4, pp. 438-442

[48] L. M. Po and W. C. Ma (1996), "A novel four-step search algorithm for fast block motion estimation", *IEEE Transactions on Circuits & Systems for Video Technology*, vol. 6, pp. 313-317

[50] A. M. Tekalp (1995)," Video compression" *Prentice Hall, PTR*, pp. 419-428

[51] B. D. Lucas and T. Kanade (1981), "An iterative image registration technique with an application to stereo vision", *Proceeding of DARPA image understanding workshop*, pp. 121-130

[52] H. H. Nagel and W. Enkelmann (1986),"An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences" *IEEE Transactions on Pattern Analysis and Machine Intelligence* , vol. 8, pp. 565-593

[53] J. D. Robbins and A. N. Netravali (1983), "Recursive motion compensation: a review," in *Image Sequence Processing and Dynamic Scene Analysis*, pp. 76-103

[54] D. R. Walker and K. R. Rao (1984). "Improved pel-recursive motion compensation", *IEEE Transaction on Communications.*, vol. COM-32, pp. 1128-1134

[55] J. Biemond, L. Looijenga, D. E. Boekee, and R. H. J. M. Plompen (1987), " A pel-recursive Wiener-based displacement estimation algorithm", *IEEE Transaction on. Signal Processing, vol. 13*, pp. 399-412

[56] A. Puri, H. M. Hang, and D. L. Schilling (1987), "An efficient block matching algorithm for motion-compensated coding", in *Proc. IEEE Int. Conf. Acoust.,Speech, and Signal Proc., pp. 1063-1066*

[57] L. K. Liu and E. Feig (1996), "A block-based gradient descent search algorithm for block motion estimation in video coding", *IEEE Transactions on Circuits & Systems for Video Technology, vol. 6*, pp. 419-422

[58] M. Bierling(1988), "Displacement estimation by hierarchical block-matching," in Proc. SPIE Conf. on Vis. Commun. and Image Proc., vol. 1001, pp. 942-951

[59] ITU-T SG16 (1998), "Video codec test model, near-term, version 8 (TMN8) release 0", *in Draft ITU-T recommendation H.263 version 2 /H.263+*

[60] L. M. Po and C. K. Cheung(1996), "A new center-biased orthogonal search algorithm for fast block motion estimation", *in Proc. of IEEE* Region Ten Conf., vol. 2, pp. 874-877

[61] K. H. K. Chow and M. L. Liou(1993), "Generic motion search algorithm for video compression", *IEEE Transactions on Circuits & Systems for Video Technology*, vol. 3, pp. 440- 445

[62] B. Liu and A. Zaccarin(1993), "New fast algorithms for the estimation of block motion vectors", *IEEE Transactions on Circuits & Systems for Video Technology*, vol. 3, pp. 148-157

[63] C. D. Bei and R. M. Gray(1985), "An improvement of the minimum distortion encoding algorithm for vector quantization", *IEEE Transactions on. Communications., vol. COM-33*, pp 1132-133

[64] T. Yoshida and et. al. (1997), "Block motion estimation using block integration based on reliability metric", in *Proc. IEEE Int. Conf. on Image Proc., vol. II*, pp. 152-155

[65] T. Yoshida and et. al.(1995), "Reliability metric of motion vectors and its application to motion estimation" in Proc. VCIP, pp. 799-809

[66] N. Brady(1999), "MPEG-4 standardized method for the compression of arbitrary shaped video objects," *IEEE Transactions on Circuits & Systems for Video Technology*, vol. 9, pp. 1170–1189

[67] P. M. Kuhn and W. Stechele(1998), "Complexity analysis of the emerging MPEG-4 standard as a basis for VLSI implementation," in *Proc. SPIE Visual Communication and Image Processing 1998*, vol. 3309, pp.798–809

[68] D. Gong and Y. He(2000), "Computation complexity analysis and VLSI architectures of shape coding for MPEG-4," in *Proc. SPIE Visual Communicationand Image Processing 2000*, vol. 4067, pp. 1459–1470.

[69] Y.-C. Wang, H.-C. Chang, W.-M. Chao, and L.-G. Chen(2001), "An efficient architecture of binary motion estimation for MPEG-4 shape coding," in *Proc. SPIE Visual Communications and Image Processing 2001*, vol. 4310, pp. 959–967

[70] H. Shao, W. Zhu, and Y. Zhang(2000), "User and content-aware object-basedvideo streaming over the internet," in *Proc. SPIE Visual Communication and Image Processing 2000*, vol. 4067, pp. 653–661

[71] T. Koga, K. Iinuma, A. Hirano, and T. Ishiguro(1981), "Motion-compensated interframe coding for video conferencing," in *National TelecommunicationsConf.*, pp. G5.3.1–G.5.3.5.

[72] R. Li, B. Zeng, and M. L. Liou(1994), "A new three-step search algorithm forblock motion estimation," *IEEE Transactions on Circuits & Systems for Video Technology*, vol. 4, pp. 438–442

[73] R. Srinivasan and K. Rao(1985). "Predictive coding based on efficient motion estimation," *IEEE Transaction on Communications.*, vol. COM-33, pp. 888–896

[74] L. M. Po and W. C. Ma (1996), "A novel four-step search algorithm for fast block motion estimation," *IEEE Transactions on Circuits & Systems for Video Technology*, vol. 6, pp. 313–317

[75] J. Lu and M. L. Liou(1997), "A simple and efficient search algorithm for blockmatching motion estimation," *IEEE Transactions on Circuits & Systems for Video Technology*, vol. 7, pp. 429–433

[76] J. Y. Tham, S. Ranganath, M. Ranganath, and A. A. Kassim(1998), "A novel unrestricted center-biased diamond search algorithm for block motion estimation," *IEEE Transactions on Circuits & Systems for Video Technology*, vol. 8, pp.369–377

[77] Bunschoten R, Krose B. (2003). "Robust scene reconstruction from an omnidirectional vision system" *IEEE Transactions on Robotics & Automation*, vol.19, no.2, pp.351-7

[78] Demin Wang. (2003) "Comparison of motion-compensated algorithms for frame interpolation" Optical Engineering, vol.42, no.2, pp.586-90. Publisher: SPIE, USA.

[79] Woo Young CHOI, Rae-Hong PARK (1989). " Motion vector coding with conditional transmission" Signal Processing, Vol. 18, No. 3, pp. 259-267

[80] S. Zhu and K. K. Ma (2000), "A new diamond search algorithm for fast block-matching motion estimation," *IEEE Transaction on Image Processing*, vol. 9, pp. 287-290

[81] Ming-gang Liu and Chao-huan Hou (2001), "A Fast Block-Matching Motion Estimation Algorithm Based on Spatial-Temporal Motion Vector Correlation."

Proc. 2001 International Symposium on Intelligent Multimedia, Video and Speech Processing

[82] C. H. Cheng and L. M. Po (2002), "A novel four-step search algorithm for fast block motion estimation", *IEEE Transactions on Circuits & Systems for Video Technology*, vol. 12, pp. 1168- 1177

[83] C. H. Cheng and L. M. Po (2003), "Adjustable Partial Distortion Search Algorithm for Fast Block Motion Estimation" *IEEE Transactions on Circuits & Systems for Video Technology*, vol. 13, pp. 100- 110

[84] A. M. Tourapis, O. C. Au, Ming L. Liou, G. Shen, and I. Ahmad (2000), "Optimizing the MPEG-4 encoder—advanced diamond zonal search", *ISCAS 2000 IEEE International Symposium on Circuits and Systems, May 28-31*, pp. 674-677

[85] H. Nisar, and T. S. Choi (2000), "An Advanced Center Biased Three Step Search Algorithm for Motion Estimation", *IEEE transactions on Consumer electronics,* vol.66, pp.95-98

[86] H. Nisar, and T. S. Choi (1999), "An new four step search algorithm with minimum checking points for block motion estimation", *Proceedings of MICC and ISCE'99*, Melaka, Malaysia, pp.211-215

[87] J. N. Kim and T. S. Choi (1998), "A fast three steps search algorithm with minimum checking points using Unimodal error surface assumption", *IEEE Transactions on Consumer electronics, vol.44*

[88] H. S. Kang(2000), "Motion estimation algorithm with low complexity for video compression", *Electronics Letters* Vol.36, No.18

[89] K. R. Namuduri, and A. Ji (2001), "Computation and performance trade-offs in motion estimation algorithm" *IEEE Transactions on Circuits & Systems for Video Technology*, pp263-267.

[90] A. M. Tourapis, Oscar C. Au. Ming L.Liou(2001), " New results on zonal based motion estimation algorithms- advanced predictive diamond zonal search", *IEEE*, pp183-186

[91] Y. Hengli and Z. Zhengbing(2002), "Comparing block matching algorithms," *SPIE-Int. Soc. Opt. Eng. Proceedings of Spie - the International Society for Optical Engineering*, vol. 4925, pp. 55-59

[92] H. Sun, H.-y. Zhao, J.-w. Xiong, and H. Jin(2002), "Method of estimating image motion based on the optical flow model," *Optics & Precision Engineering*, vol. 10, pp. 443-447

[93] N. Yu, H. Chen, Z. Liu, X. Li, and F. Niu (2002), "A new fast block-matching algorithm based on block sum pyramid," *SPIE-Int. Soc. Opt. Eng. Proceedings of Spie - the International Society for Optical Engineering*, vol. 4875, pp. 559-566

[94] Y. Altunbasak, R. M. Mersereau, and A. J. Patti(2003), "A fast parametric motion estimation algorithm with illumination and lens distortion correction," *IEEE Transactions of Image Processing*, vol. 12, pp. 395-408,

[95] F. Boughorbel, P. Crilly, A. Koschan, and M. Abidi(2003), "Estimating 3D camera motion without correspondences using a search for the best structure," *Pattern Recognition Letters*, vol. 24, pp. 327-337

[96] D. Cremers and C. Schnorr(2003), "Statistical shape knowledge in variational motion segmentation," *Image & Vision Computing*, vol. 21, pp. 77-86

[97] Y. Donghoon, J. Sung Kyu, and R. Jong Beom(2003), "Fast motion estimation for shape coding in MPEG-4," *IEEE Transactions on Circuits & Systems for Video Technology*, vol. 13, pp. 358-363

[98] K. Jong Bae and K. Hang Joon(2003), "Efficient region-based motion segmentation for a video monitoring system," *Pattern Recognition Letters*, vol. 24, pp. 113-128

[99] K. Jong-Nam, B. SeongChul, and A. ByungHa(2003), "Full search based fast block matching algorithm with efficient matching order in motion estimation," *IEICE Transactions on Communications*, vol. E86-B, pp. 1191-1195

[100] Y. Keller and A. Averbuch(2003), "Fast gradient methods based on global motion estimation for video compression," *IEEE Transactions on Circuits & Systems for Video Technology*, vol. 13, pp. 300-309

[101] C. J. Kuo, L. Tsang-Gang, H. Ruey-Song, and S. F. Odeh(2003), "Facial model estimation from stereo/mono image sequence," *IEEE Transactions on Multimedia*, vol. 5, pp. 8-23

[102] C. Lap-Pui and Z. Ce(2003), "A fast octagon-based search algorithm for motion estimation," *Signal Processing*, vol. 83, pp. 671-675

[103] F. Moschetti, M. Kunt, and E. Debes(2003), "A statistical adaptive block-matching motion estimation," *IEEE Transactions on Circuits & Systems for Video Technology*, vol. 13, pp. 417-431

[104] N. Ploskas, D. Simitopoulos, D. Tzovaras, G. A. Triantafyidisa, and M. G. Strintzis(2003), "Rigid and non-rigid 3D motion estimation from multiview image sequences," *Signal Processing: Image Communication*, vol. 18, pp. 185-202

[105] J. Xuan, Z. Ce, and C. Lap-Pui(2003), "Smooth constrained motion estimation for video coding," *Signal Processing*, vol. 83, pp. 677- 680

[106] J. Y Kim, S B. Yang (1999). "An efficient search algorithm for block motion estimation", *IEEE Workshop on Signal Processing Systems. SiPS 99. Design and Implementation* pp.100-109.

[107] http://www.ece.cmu.edu/~ee899/project/deepak_mid.htm

[108] http://atlantis.ucc.ie/dvideo/contents.htm

[109] D. Weng (2003), "Very Low-Bit Rate Digital Video Coding", *Literature review*, University of Newcastle upon Tyne

[110] I. Kompatsiaris, D. Tzovaras, M.G.Strintzis (1998), "3-D model-based segmentation of videoconference image sequences" *IEEE Transactions on Circuits & Systems for Video Technology*, vol. 8, pp. 547-561

[111] Y. K. Lai, L. G. Chen (1998), "A data-interlacing architecture with two-dimensional data-reuse for full-search block-matching algorithm" *IEEE Transactions on Circuits & Systems for Video Technology*, vol. 8, pp. 124-127

[112] J. N. Kim, T. S. Choi (2000), "A fast full-search motion-estimation algorithm using representative pixels and adaptive matching scan" *IEEE Transactions on Circuits & Systems for Video Technology*, vol. 10, pp. 1040-1048.

[113] K. Andra, C. Chakrabarti, T. Acharya (2003), "A High-Performance JPEG2000 Architecture" *IEEE Transactions on Circuits & Systems for Video Technology*, vol. 13, pp. 209-218.

[114] S. H. Kim, R.H. Park (2003), "Fast Local Motion-Compensation Algorithms for Video Sequences With Brightness Variations" *IEEE Transactions on Circuits & Systems for Video Technology*, vol. 13, pp. 289-299.

[115] K. Lengwehasatit, Antonio Ortega (2001), "Probabilistic Partial-Distance Fast Matching Algorithms for Motion Estimation" *IEEE Transactions on Circuits & Systems for Video Technology*, vol. 11, pp. 139-152.

[116] Y. Zhao, I. E. G. Richardson (2003), "Macroblock Skip-Mode Prediction for Complexity Control Of Video Encoders" *Proceeding of International Conference on Visual Information Engineering 2003"*, pp. 5-8.

[117] M. E. Al-Mualla, C.N. Canagarajah, D. R. Bull(2003), "Motion Field Interpolation for Improved Motion Compensation and Frame-Rate Conversion" *Proceeding of International Conference on Visual Information Engineering 2003"*, pp.17-20.

[118] J. H. Lee, K. W. Lim, B. C. Song, J. B, Ra (2001), "A fast multi-resolution block matching algorithm and its LSI architecture for low bit-rate video coding" *IEEE*

Transactions on Circuits & Systems for Video Technology, vol. 11, pp. 1289-1301.

[119] S. Dutta, W. Wolf, (1996), "A flexible parallel architecture adapted to block-matching motion-estimation algorithms" *IEEE Transactions on Circuits & Systems for Video Technology*, vol. 6, pp. 74-86.

[120] Z. He, M. L. Lieo (1997) "A high performance fast search algorithm for block matching motion estimation" *IEEE Transactions on Circuits & Systems for Video Technology*, vol. 7, pp. 826-828.

[121] L. G. Chen, Y. C. Liu (1994), "A high quality MC-OBTC Codec for video signal processing" *IEEE Transactions on Circuits & Systems for Video Technology*, vol. 7, pp. 92-98.

[122] P. Hsu, K. J. R. Liu, T. Chen (2001), "A low bit-rate video codec based on two-dimensional mesh motion compensation with adaptive interpolation" *IEEE Transactions on Circuits & Systems for Video Technology*, vol. 11, pp. 111-117.

[123] M. J. Chen, L. G. Chen, T. D. Chiueh, Y. P. Lee (1995), "A new block-matching criterion for motion estimation and its implementation" *IEEE Transactions on Circuits & Systems for Video Technology*, vol. 5, pp. 231-236.

[124] V. G .Moshnyaga (2001), "A new computationally adaptive formulation of block-matching motion estimation" *IEEE Transactions on Circuits & Systems for Video Technology*, vol. 11, pp. 118 – 124.

[125] J. H. Luo, C. N. Wang, T. Chiang (2002), "A novel all-binary motion estimation (ABME) with optimized hardware architectures" *IEEE Transactions on Circuits & Systems for Video Technology*, vol. 12, pp. 700 – 712.

[126] J. F. Shen, T. C. Wang, L. G. Chen (2001), "A novel low-power full-search block-matching motion-estimation design for H.263+" *IEEE Transactions on Circuits & Systems for Video Technology*, vol. 11, pp. 890-897.

[127] J. Y. Tham, S. Ranganath , M. Ranganath , A. A. Kassim (1998), "A novel unrestricted center-biased diamond search algorithm for block motion estimation" *IEEE Transactions on Circuits & Systems for Video Technology*, vol. 8, pp. 369- –377.

[128] M. F. Chowdhury, A. F. Clark, A. C. Downton, E. Morimatsu, D.E. Pearson (1994), "A switched model-based coder for video signals" *IEEE Transactions on Circuits & Systems for Video Technology*, vol. 4, pp216-227.

[129] M. Caccia (2003), "Vision-based linear motion estimation for unmanned underwater vehicles" *IEEE International Conference on Robotics and Automation,* Part vol.1, pp.977-982.

[130] H. M. Jong, L. G. Chen, T. D. Chiueh (1994) "Accuracy improvement and cost reduction of 3-step search block matching algorithm for video coding" *IEEE Transactions on Circuits & Systems for Video Technology*, vol. 4, pp. 88-90.

[131] T. Sikora (1997), "The MPEG-4 video standard verification model" *IEEE Transactions on Circuits & Systems for Video Technology*, vol. 7, pp. 19-31.

[132] J. Salas (2003), "An iterative framework for projection-based image sequence registration" *IEEE International Conference on Robotics and Automation,* Part vol.2, pp.1893-1898