Loughborough University

Thesis Access Form

Copy No……...…….**Location**…….……...…

Author Ashraf Al-Najdawi

Title "A Multi-Objective Performance Optimisation Framework for Video Coding"

Status of access OPEN / RESTRICTED / CONFIDENTIAL

Moratorium Period:……years, ending……../……200…….

Conditions of access approved by (CAPITALS):……

Supervisor (Signature)……...……...

Department of Electronic and Electrical Engineering

Author's Declaration: *I agree the following conditions:*

Open access work shall be made available (in the University and externally) and reproduced as necessary at the discretion of the University Librarian or Head of Department. It may also be digitised by the British Library and made freely available on the Internet to registered users of the EThOS service subject to the EThOS supply agreements.

*The statement itself shall apply to **ALL** copies including electronic copies:*

This copy has been supplied on the understanding that it is copyright material and that no quotation from the thesis may be published without proper acknowledgement.

Restricted/confidential work: All access and any photocopying shall be strictly subject to written permission from the University Head of Department and any external sponsor, if any.

Author's signature…….Date……...…

users declaration: for signature during any Moratorium period (Not Open work): ***I undertake to uphold the above conditions:***			
Date	Name (CAPITALS)	Signature	Address

A Multi-Objective Performance Optimisation Framework for Video Coding

by

Ashraf Al-Najdawi

A Doctoral Thesis

Submitted in partial fulfilment

of the requirements for the award of

Doctor of Philosophy

of

Loughborough University

October 2010

Acknowledgements

I would like to express my heartfelt thanks to a number of people who have supported me and made this research possible.

I would like to thank my academic supervisor: Prof. Roy Kalawsky, for his continual guidance, and his invaluable support and encouragement throughout the duration of this project. I consider it a privilege to have had the opportunity to work with him and share his valuable knowledge and expertise.

Special gratitude goes to my father, Amin; my mother, Najwa; my brother, Nijad and his wife Sara; my sisters, Hazar, Manar, Saba; and my lovely niece Yara. Their unconditional love and support kept me going and spurred me on to greater heights. I would not and could not have done this without them.

I also extend my thanks to my friends for reminding me that there is more to life than work. I also thank them for providing me with all "the much needed" distractions. Special thanks to Dr. Khusvinder Gill (a.k.a. Kush) for always bringing on the good times and for always being there through the challenging ones.

Last but not least, this thesis would not be complete without the constant support and encouragement of my lovely soul-mate, Lina; to whom I dedicate this work.

Abstract

Digital video technologies have become an essential part of the way visual information is created, consumed and communicated. However, due to the unprecedented growth of digital video technologies, competition for bandwidth resources has become fierce. This has highlighted a critical need for optimising the performance of video encoders. However, there is a dual optimisation problem, wherein, the objective is to reduce the buffer and memory requirements while maintaining the quality of the encoded video. Additionally, through the analysis of existing video compression techniques, it was found that the operation of video encoders requires the optimisation of numerous decision parameters to achieve the best trade-offs between factors that affect visual quality; given the resource limitations arising from operational constraints such as memory and complexity.

The research in this thesis has focused on optimising the performance of the H.264/AVC video encoder, a process that involved finding solutions for multiple conflicting objectives. As part of this research, an automated tool for optimising video compression to achieve an optimal trade-off between bit rate and visual quality, given maximum allowed memory and computational complexity constraints, within a diverse range of scene environments, has been developed. Moreover, the evaluation of this optimisation framework has highlighted the effectiveness of the developed solution.

List of Publications

- Al-Najdawi, A. & Kalawsky, R.S. 2010, "Visual quality assessment of video and image sequences—A human-based approach", Journal of Signal Processing Systems, vol. 59, no. 2, pp. 223-231.

- Al-Najdawi, A. & Kalawsky, R.S. 2008a, "A multi-objective optimization framework for video compression and transmission", 6th International Symposium on Communication Systems, Networks and Digital Signal Processing, 2008.IEEE, Graz, Austria, pp. 336.

- Al-Najdawi, A. & Kalawsky, R.S. 2007, "Quantitative quality assessment of video sequences A human-based approach", 6th International Conference on Information, Communications & Signal Processing, 2007, IEEE, Singapore

List of Abbreviations

AVC	-	Advanced Video Coding
BG	-	Background
B-Frame	-	Bi-directional predicted Frame
CABAC	-	Context-based Adaptive Binary Arithmetic Coding
CAE	-	Context-based Arithmetic Encoding
CAVLC	-	Context-based Adaptive Variable-Length Coding
C-D	-	Complexity-Distortion
CIF	-	Common Intermediate Format, a video format
C-M-R-D	-	Complexity-Memory-Rate-Distortion
CPU	-	Central Processing Unit
C-R-D	-	Complexity-Rate-Distortion
DCT	-	Discrete Cosine Transform
DVD	-	Digital Versatile Disc
DWT	-	Discrete Wavelet Transform
EA	-	Evolutionary Algorithm
EPZS	-	Enhanced Predictive Zonal Search, a motion estimation algorithm of H.264

FG	-	Foreground
FME	-	Fast Motion Estimation
FS	-	Full Search, a motion estimation algorithm of H.264
GA	-	Genetic Algorithm
GOP	-	Group of Pictures
H.264	-	A Video Coding Standard
HVS	-	Human Visual System
I-Frame	-	Intra Frame
ISO/IEC	-	International Standards Organization, International Electrotechnical Commission
IT	-	Integer Transform
ITU-T	-	International Telecommunications Union, Telecommunication Standardization Sector
MC	-	Motion Compensation
ME	-	Motion Estimation
MF	-	Multiplication Factor
MOEA	-	Multi-Objective Evolutionary Algorithm
MOGA	-	Multi-Objective Genetic Algorithm
MOO	-	Multi-Objective Optimization
MOOP	-	Multi-Objective Optimization Problem
MPEG	-	Motion Picture Experts Group, a Committee of ISO/IEC
MPEG-4	-	A Multimedia Coding Standard
MSE	-	Mean Squared Error
MV	-	Motion Vector
NAL	-	Network Abstraction Layer

NSGA	-	Non-Dominated Sorting Genetic Algorithm
PC	-	Personal Computer
P-D	-	Power-Distortion
PPS	-	Picture Parameter Set
P-R-D	-	Power-Rate-Distortion
PSNR	-	Peak Signal to Noise Ratio, an objective quality measure
QCIF	-	Quarter Common Intermediate Format
QP	-	Quantization Parameter
QVA	-	Quantitative Visual Assessment
R-D	-	Rate-Distortion
RGB	-	Red/Green/Blue colour space
RMSE	-	Root Mean Square Error
RBSP	-	Raw Byte Stream Payload
ROI	-	Region of Interest
SIF	-	Source Input Format, a video format
SQCIF	-	Sub Quarter CIF
SVC	-	Scalable Video Coding
UMHS	-	Unsymmetrical Multi-Hexagon Search, a motion estimation algorithm of H.264
UMV	-	Un-Manned Vehicle
SUMHS	-	Simplified Unsymmetrical Multi-Hexagon Search, a motion estimation algorithm of H.264
TV	-	Tele-Vision
VCEG	-	Video Coding Experts Group
VCL	-	Video Coding Layer
VLC	-	Variable Length Coding

List of Symbols

GB	-	Giga Byte
GHz	-	Gigahertz
KB	-	Kilo Bytes
kbps	-	Kilo bits per second
kHz	-	Kilo Hertz
m	-	Meter
MB	-	Mega Byte
Mbps	-	Mega bits per second
MHz	-	Megahertz
ms	-	Milliseconds

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background to the Research

The past two decades have witnessed widespread adoption of digital video technologies such as digital television, internet video streaming, and mobile broadcasting. Digital video technologies have become an essential part of the way visual information is created, consumed and communicated [1]. International video coding standards have played a fundamental role in increasing utilisation of digital video technologies by assuring interoperability among products developed by different manufacturers. At the same time, these standards allow sufficient flexibility in optimising and moulding the technology to fit a given application and make cost-performance trade-offs best suited to particular requirements [2]. Nowadays, there is no doubt that digital video has become an integral component of entertainment, communications and broadcasting industries.

1.2 Research Challenge

The operation of video encoders requires the optimisation of numerous decision parameters to achieve the best trade-off between bit rate and quality given

the resource limitations arising from operational constraints such as memory and complexity. Optimising the performance of video codecs often involves finding solutions for multiple conflicting objectives, e.g. rate-distortion optimisation. However, solving optimisation problems with multiple conflicting objectives is a difficult process that might be computationally expensive. However, a perfect multi-objective optimisation solution that satisfies all objective functions and complies with all constraints associated with the decision variables may not exist [3].

Codecs such as H.264/AVC have provided a more enhanced coding efficiency compared to prior widely used standards such as MPEG-2. Consequently, H.264/AVC is now successful over a wide span of applications including video conferencing, broadcasting, surveillance, military applications and online video streaming [2]. The added features and functionalities within H.264/AVC have provided a marked improvement in coding efficiency. However, all of the ITU-T and ISO/IEC video coding standards have only defined the decoding process by imposing restrictions on the syntax and bitstream, while the encoding process was out of the scope of the H264/AVC standard and subsequently left undefined. This limitation has allowed a high degree of flexibility to optimize implementations of video codecs. However, it provides no guarantees for a high-quality reproduction of video streams, as it allows even crude encoding techniques to be considered conforming [1]. In addition, the aforementioned added features and functionalities and the enhancements on coding efficiency have all come at a price. For example, the focus on coding efficiency has resulted in an increased demand on system resources as a result of increased computational complexity and memory requirements.

In order to achieve high quality compressed video streams, researchers have attempted to autonomously assess visual video quality and emulate human's perception of quality. However, there has been limited research in the area of evaluating image enhancement/restoration techniques, by defining viewability, even though interest in the topic is quite old [[4], [5]]. Hence, there is a considerable need for the development of viewability measures that correlate well with human vision, are easy to implement, and computationally cheap.

1.3 Motivations for the Research

In video transmission over low-bandwidth channels, high-quality video and sufficient channel throughput should be guaranteed. However, as a result of the unprecedented growth of wireless communication technologies, competition for bandwidth resources has become fierce. This highlights a critical need for effective data compression techniques. However, there is a dual optimisation problem, wherein, the objective is to reduce the buffer and memory requirements while maintaining the quality of the transmitted video. This enables the compressed video streams to match a wide range of channel bandwidths in relation to different application requirements. Furthermore, an appropriate utilisation of memory and bandwidth resources guarantees a reduction in the end-to-end video streaming and processing delay.

1.4 Aim and Objectives of the Research

This research aims to develop an automated tool for optimising video compression to achieve an optimal trade-off between bit rate and visual quality, given maximum allowed memory and computational complexity constraints within a diverse range of scene environments.

The specific research objectives associated with the aforementioned research aim are as follows:

- Review the existing literature available on video coding standards and the associated methods for evaluating visual quality of compressed images and video sequences (Chapter 2).
- Review the existing literature available on multi-objective optimisation approaches and their potential application for enhancing the performance of video codecs (Chapter 3).
- Analyse and identify the encoding parameters that have a significant impact on CPU and memory utilisation, and rate-distortion characteristics (Chapter 4).
- Develop a novel technique for quantitatively assessing the quality of image sequences without the need for a reference image and in a way that precisely correlates to human judgement of quality (Chapter 5).

- Design a novel framework for improving the compression of images and video sequences without compromising visual quality, incorporating the coding parameters that have a significant impact on computational complexity and rate-distortion characteristics (Chapter 6).
- Implement and evaluate the overall performance of the conceptual multi-objective optimisation framework (Chapter 7).

1.5 Contributions of the Research

The specific contributions to the research are as follows:

1. A comprehensive analysis of the effect of varying a selected set of compression parameters on the efficiency of the H.264/AVC video encoder.

2. A novel technique for quantitatively assessing the visual quality of image sequences based on human judgement on quality, without the need for a reference image.

3. The development of a regression model that correlates objective quality metrics to the subjective ones for 5 different scene categories.

4. The development of a tool that quantitatively measures the quality of video sequences based on human judgement on quality

5. The development of a mathematical representation for objective and constraint functions.

6. The findings of the evaluation of the multi-objective optimisation framework

1.6 Organisation of the Thesis

The structure of this thesis is as follows: Chapter 2 provides a comprehensive introduction to video coding techniques and investigates potential optimisation areas. Chapter 3 reviews evolutionary multi objective optimisation techniques and their possible role in optimising the performance of video codecs. Chapter 4 presents the performance analysis of H.264/AVC video codec and identifies the encoding parameters that have a significant impact on CPU and

memory utilisation, and rate-distortion characteristics. Chapter 5 presents a novel technique for quantitatively assessing the quality of image sequences without the need for a reference image and in a way that precisely correlates to human judgement of quality. Chapter 6 introduces the design for the proposed conceptual model of a multi objective optimisation framework for video compression, incorporating the coding parameters that have a significant impact on computational complexity and rate-distortion characteristics. Chapter 7 details the implementation and evaluation of the conceptual model of the multi objective optimisation framework. Chapter 8 concludes the thesis with a summary of the main contributions of the research as well as areas for future research.

Chapter 2

Introduction to Video Coding

2.1 Introduction

As introduced in Chapter 1, digital video technologies have become an essential part of the way visual information is created, consumed and communicated. Nowadays, there is no doubt that digital video has become an integral component of entertainment, communications, and broadcasting industries.

Most video transmission, storage, and processing environments do not support uncompressed "raw" video due to the inherited limitations in data processing, storage, and transmission capabilities. Therefore, bandwidth-intensive raw digital video has to be reduced to a manageable size to suit these capabilities. For example, using a typical PAL video resolution of 720 x 576 pixels with a refresh rate of 25 frames per second (fps), and 8-bit colour depth per pixel requires a bandwidth of 166Mb/s. At this rate, 10 minutes of video recording requires 12.16 Gigabytes of storage. Whereas a High Definition Television (HDTV) video with a typical resolution of 1920 x 1080 pixels, a refresh rate of 60 fps, and 8-bit colour depth per pixel, requires a bandwidth of 1.99 Gb/s. At this rate, 10 minutes of video recording requires approximately 149.25 Gigabytes of storage. Handling data of this size places extreme computational and storage demands on resources. Even with

recent advances in processing power, storage, and transmission capacities, video compression will remain an essential constituent of multimedia services for many years to come.

Video compression algorithms operate by removing redundancy that exists in spatial, temporal, and/or frequency domains of digital video sequences. Spatial redundancy is significant when there is little variation in the content of an image or a video frame. Temporal redundancy is significant when there is little or no change in content between successive frames. On the other hand, redundancy in the frequency domain exists in the form of high-frequency components. Smoothing the image using a low-pass filter removes the high frequency content. The removal of some high frequency components should not affect the perceptual quality of the image sequence; this is primarily due to the lower sensitivity of the human visual system to higher frequencies [6]. However, the performance of a video compressor does not only depend on the level of redundancy in a video sequence, it also depends on whether the compression technique used for coding is lossless or lossy.

Lossless compression exploits the statistical redundancy in image and video signals and in most cases the decompressed signal is a perfect match to the original signal. However, this technique leads to a modest amount of compression, and therefore, it is rarely used for image and video compression. On the other hand, the widely used lossy compression techniques discard data in order to achieve a high compression ratio. This leads to significant decrease in file sizes, but at the expense of a considerable amount of data loss. Lossy compression will be the subject of research throughout this Thesis.

2.2 Fundamentals of Video Coding

The process of compression and decompression of a digital video signal is known as video coding. A digital video represents scenes sampled at certain points in time in the form of frames. In other words, a video sequence represents a complete visual scene at a certain point in time sampled spatially and temporally. In commercial TV systems, the sampling process is repeated at 1/25 or 1/30 second intervals in order to produce a moving video signal. A frame of digital video

typically consists of three rectangular arrays of integer-valued samples. The three sets of samples (components) are required to represent a scene in colour.

2.2.1 Terminology and Abbreviations

A brief summary of the fundamental terminology and abbreviations used in video coding are as follows:

- **Pixel:** A colour element at one position in a displayed image.
- **Luminance (or Luma):** Luminance is a measure of gray tone values computed from RGB. In this context it refers to a sample or an array representing a video brightness signal, often symbolized as Y.
- **Chrominance (or Chroma):** A sample or array representing a blue or red video colour difference signal, often symbolized as C_b and C_r, or U and V.
- **Sample:** A luma or chroma component at one position in a video frame.
- **Frame:** A set of samples representing a single time instant of a progressive video signal. A video frame consists of one array of luma samples and two arrays of chroma samples.
- **Frame rate (frame frequency):** The number of frames or images that are projected or displayed per second. Frame rate is often expressed in frames per second (fps), or simply in hertz (Hz).
- **Resolution:** The dimensions of a video frame or an image, in pixels.
- **Macroblock:** A 16 x 16 array of luma pixels (Y) and associated chroma pixels (U and V). In this thesis, the chroma components of a macroblock are assumed to each consist of 8 x 8 pixels (unless otherwise stated).
- **Block:** An M x N array of samples.

2.2.2 Sampling Formats

Components of a video scene typically fall into two categories: spatial and temporal. Spatial components include: colour, shape of objects, and texture variations within the scene. Temporal components include: object motion, movement of camera, and changes in lighting. A natural visual scene is spatially and

temporally continuous. Representing a digital visual scene incorporates sampling the scene spatially as a frame that has defined values at a set of sampling points, and temporally as a series of frames sampled at fixed time intervals. Each picture element (pixel) is represented as a set of numbers describing brightness and colour of the spatio-temporal sample [6].

Spatial sampling is based on measuring/capturing signal levels at discrete spatial points. One approach for implementing spatial sampling is to superimpose a grid on a video frame at a point in time, where sampling occurs at each of the intersection points of the grid. Choosing a coarse sampling grid reduces the resolution of the frame as the number of samples decreases. Choosing a fine grid increases the number of samples and therefore yields better resolution.

Motion in a digital video is captured by temporal sampling, where a snapshot of the scene is taken at regular time intervals. The temporal sampling rate is usually referred to as frame rate. A higher frame rate yields smoother motion. Low frame rates, below 10 frames per second (fps), are usually used for low bitrate video transmission or streaming applications. Sampling at 25 or 30 fps is typically used for television.

A video signal may be sampled in one of two basic sampling formats. The first is called progressive sampling, where a video signal is sampled as a series of complete frames. The second is called interlaced sampling where video signal is sampled as a sequence of interlaced fields (see Figure 2-1). When interlaced sampling is used, a complete video frame will contain two interleaved fields, a top field and bottom field. Each field consists of either even or odd-numbered lines (rows). Unlike progressive sampling, where the entire frame is captured at each sampling point of time, only one of the two fields is captured at each temporal sampling interval. The advantage of this method is that motion in a video will appear smoother; the reason is that in interlaced sampling, it is possible to send twice as many fields per second as the number of frames in an equivalent progressive sequence [6]. However, this can cause problems for images with sharp edges.

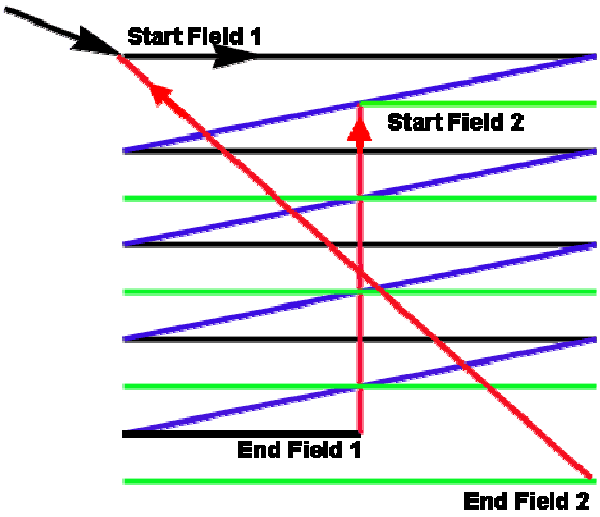


Figure 2-1: Fields in interlaced sampling

2.2.3 Colour Spaces

A colour space is a mathematical model that describes how a colour and brightness (luminance) can be represented as a finite sequence of numbers. In other words, a colour space is a method by which we are able to specify, visualise and create colour. The human visual system can define colour by its attributes such as brightness, saturation and hue. Computers define colour in different ways; colour is quantified and may, for example, be described as the amount of red, green, and blue emission needed to match the colour [7]. Colour images require at least three numbers per pixel to give an accurate representation of colour [6]; these numbers form the co-ordinates for the position of the colour within the colour space being used. Following is a brief discussion of the most commonly used colour spaces:

2.2.3.1 RGB Colour Space

In the RGB colour space, an image pixel is represented by three numbers that indicate the relative portions of red, green, and blue in that pixel. Those three components are equally important and are usually stored at the same resolution. For example, a colour image that is represented in RGB with a resolution of 704 x 576 (4QCIF) requires 1 byte of storage per colour per pixel. Thus, the whole colour image will require a total of 1.19 Mbytes of storage. The RGB colour space is easy to implement, therefore, it is very common and is used in almost every visual system, but yet, it is device-dependent and non-linear with visual perception [7].

2.2.3.2 YCbCr Colour Space

Another colour representation that is often used also has three components namely: Y, C_b, and C_r. Component Y is called *luma* (luminance) and represents brightness. The other two components, C_b and C_r, are called *chroma* (chrominance) components and represent the extent to which the colour deviates from gray towards blue and red [1]. Although human visual system (HVS) perceives colour faster than luminance, the YC_bC_r colour space is based on the fact that the HVS is more sensitive to luminance than to colour. Therefore, it is more efficient to separate luminance from the colour information and to assign higher resolution to luminance than colour without having an obvious effect on visual quality. The YC_bC_r colour space is often referred to as YUV. The terms YC_bC_r and YUV will be used interchangeably in this thesis. The conversion equations from RGB to YUV (and vice versa) can be found in literature in different forms [6]. The luminance Y can be calculated as a weighted sum of the RGB components:

$$Y = k_r R + k_g G + k_b B \tag{2.1}$$

Where k_b= 0.114, k_r= 0.299, and k_g= 0.587 [8]. The colour information (called the Chrominance) can be represented as:

$$\begin{aligned} C_b &= 0.564(B - Y) \\ C_r &= 0.713(R - Y) \\ C_g &= G - Y \end{aligned} \tag{2.2}$$

Where, C_b, C_r, and C_g represent the difference between colour intensity and mean luminance of each image sample. The sum of these chrominance components is always constant (i.e. $C_r + C_b + C_g$ = constant) and therefore, it is enough to represent a colour by the luminance Y and two chrominance components (C_b and C_r) since the third can be calculated from the other two [7].

One useful application of the YUV colour space is that any RGB image can be converted to YUV in order to reduce transmission and/or storage requirements (as will be presented in the next section), and conversely before displaying the

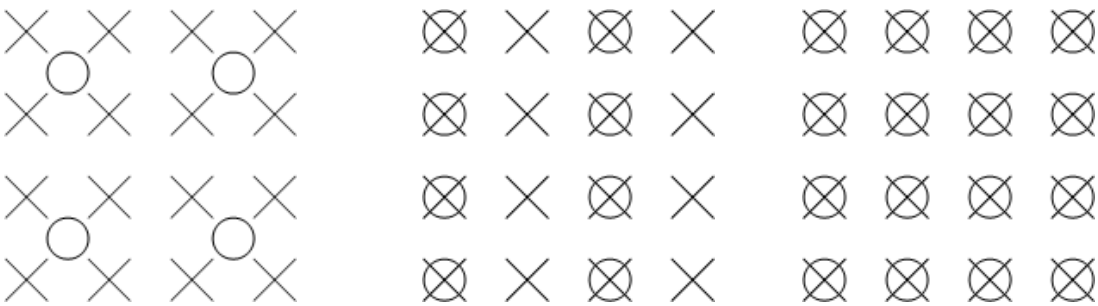
image it has to, in most cases, be converted back to RGB format using the following equations:

$$\begin{aligned} R &= Y + 1.402C_r \\ G &= Y - 0.344C_b - 0.714\,C_r \\ B &= Y + 1.772C_b \end{aligned} \tag{2.3}$$

2.2.4 YUV Sampling Formats

It follows from the definition of the YUV colour space, that the chrominance components can be sampled at lower rates (less resolution) than luminance, without compromising the overall visual quality. This leads to three possible sampling formats for the YUV colour space (Figure 2.2). The most popular format is the YUV 4:2:0 (in which chrominance (C_b and C_r) components have half the vertical and half the horizontal resolution of the luminance Y. This sampling is applied on most of the recent video applications such as video conferencing and Digital Versatile Disk (DVD).

The second sampling format is YUV 4:2:2 in which C_b and C_r are sampled at half the horizontal resolution of Y but have the same vertical resolution. This sampling format is used for high-quality video applications. The last format is the YUV 4:4:4 in which C_b and C_r are sampled at the same horizontal and vertical resolution of Y.



(a) **(b)** **(c)**

Figure 2-2: YUV sampling formats (a) 4:2:0 (b) 4:2:2 and (c) 4:4:4

To calculate the required bit rate to sample each pixel, let us assume that we have a group of four pixels, in the 4:4:4 sampling format, a group of four pixels will

contain 12 samples each coded at 8 bits, therefore, 12 x 8 = 96 bits will be required. Therefore, an average of 96/4 = 24 bits per pixel are needed to encode each pixel. If the same method is applied for the 4:2:0 sampling format, a sampling rate of 12 bits per pixel is required. Back to the previous example, an image with a resolution of 704x576 pixels encoded using 4:2:0 sampling format will require a total of 594 Kbytes of storage, i.e. half the size needed for the same image encoded using RGB sampling.

2.2.5 Digital Video Formats

Most of the modern video compression standards capture and convert video frames to a set of intermediate formats prior to compression and transmission, this basic format is referred-to as the Common Intermediate Format (CIF). CIF is used to standardize the horizontal and vertical resolutions in pixels of YUV video sequences. It was first proposed as a part of the H.261 standard [9] developed to support video-conferencing over ISDN networks, and was further extended in H.263 [10] and H.264 [11] standards. CIF specifies the resolution per video frame at 352x288 luminance pixels. Other CIF formats are defined by their resolution in reference to the full CIF format. Table 2-1 displays the main video frame formats and a brief description of their applications, the choice of format depends on the application in hand, available storage, and/or transmission capacity.

Table 0-1: Most popular common intermediate formats

Format	Luminance resolution	Sample application
Sub-QCIF	128 x 96	Mobile multimedia
Quarter CIF (QCIF)	176 x 144	Desktop video conferencing
CIF	352 x 288	Video conferencing,
4CIF	704 x 576	Standard definition TV, DVD-Video
16CIF	1408 x 1152	High Definition TV (HDTV)

2.3 Visual Quality Assessment

Video data being compressed and transmitted through communication channels is susceptible to distortion and degradation of quality. Most of the applied video compression techniques are lossy; they are based on removing redundancy in the spatial, temporal and frequency domains [6]. Therefore, substantial compression is achieved at the expense of quality. On the other hand, transmitted video data is also susceptible to various types of bit-error rates, packet losses, or even delays, all of which are factors of video data degradation.

In order to evaluate and compare the performance of different video display and communication systems, it is necessary to judge the visual quality of the video being processed. Since most video services target human observers, the judgement on visual quality has to be relevant to the way the human visual system perceives the viewability of a video sequence. This in turn brings other challenges which lie in the nonlinear behaviour of the human visual system, and the variety of factors, such as subjectivity, that can affect measuring visual quality. This makes it a difficult task and often leads to imprecise results.

There has been limited research in the area of evaluating image enhancement/restoration techniques- by defining viewability; even though interest in the topic is quite old [4, 5, 12]. Pappas and Safranek [13] state that: “Even though we use the term image quality, we are primarily interested in image fidelity, i.e., how close an image is to a given original or reference image”. They examine objective criteria for image quality that are based on models of the HVS, they also detail three models that were proposed by Lubin [14], Teo and Heeger [15], and Daly [16] and give comparative results. All of these models first perform multi-resolution frequency analysis of images, followed by contrast sensitivity, use of a masking model and finally error pooling which determines the quality of enhancement. It should be noted that Daly and Lubin’s models are exceptionally computationally complex and difficult to use for real applications [16].

2.3.1 Subjective Quality Assessment

Human's visual quality assessment is intrinsically "subjective". Our ability as human beings to assess the visual quality of an image or video is influenced by many factors such as spatial and temporal fidelity, level of interaction with the scene, viewer's state of mind, viewing environment, and how comfortable the viewing environment is [6]. Two users' visual performance could match well in terms of their ability to pick out interesting objects, but not in terms of grading image quality. For this reason, designing viewability measures that are quantitative yet correlate well with the visual perception of different human experts remains a challenging task.

In order to set a standardised benchmark for subjective visual quality assessment, the International Telecommunications Union (ITU) has proposed a set of test procedures defined in ITU-R Recommendation BT.500-11 [17]. This recommendation sets the guidelines for the subjective assessment test conditions such as the viewing distance, the test duration, and the observers' recruitment.

2.3.2 Objective Quality Assessment

The complexity and expense of subjective quality assessment, and usual variability between human observers have made it attractive to develop automatic quality assessment techniques using mathematical and computational algorithms that can predict perceived image and video quality automatically. Wang et al [18] has defined the purpose of objective quality assessment as to "*design quality metrics that can predict perceived image and video quality*".

Most of the recent objective quality assessment techniques are based on computing the quality of an image or video in reference to the original image, and therefore referred to as Full-Reference quality assessments [19]. Among those techniques are the Mean Error Squared (MSE) and Peak Signal to Noise Ratio (PSNR):

$$MSE = \frac{1}{N}\sum_{i-1}^{N}(x_i - y_i)^2 \tag{2.4}$$

Where N is the number of pixels in the frame, and x_i and y_i are the number of pixels in the original and compressed frames respectively. And

$$PSNR = 10\log_{10}\frac{L^2}{MSE} \tag{2.5}$$

Where L is the dynamic range of pixel values (L= 255 for monotonic images).

However, PSNR and MSE are criticised for not correlating very well with perceived (subjective) quality assessment. Moreover, as with all full-reference assessment techniques, PSNR and MSE cannot function if the original image does not exist. Therefore, it is highly desirable to develop quality measures that can assess image and video quality without the need to refer to the original image [18].

2.4 State-of-the-Art Video Coding Techniques

The growing interest in digital image and video applications over the past two decades has made video coding a very active field of research and development. Many coding techniques have been proposed and developed by researchers in academia and industry under the umbrella of international standardisation bodies; among these are the International Organisation of Standardization, International Electro-technical Commission (ISO/IEC), and the International Telecommunications Union, Telecommunications Standardization Sector (ITU-T). The ISO/IEC Motion Picture Experts Group (MPEG) has developed the MPEG series: MPEG-1 [20], MPEG-2 [21], MPEG-4 [22], MPEG-7 [23], and MPEG-21 [24]. The ITU-T Video Coding Experts Group (VCEG) has led the work to standardise the H.26x series of standards (H.261 [9], H.262 [25], H.263 [10], and H.264 [11]).

2.4.1 H.264 / MPEG-4 (Part 10) Advanced Video Coding

The ITU-T H.264 / MPEG-4 (Part 10) Advanced Video Coding is usually referred to as H.264/AVC. It was developed in 2003 by the Joint Video Team (JVT), consisting of IUT-T VCEG and ISO/IEC MPEG. H.264/AVC is one of the most powerful state-of-the-art video coding standards. The design of H.264/AVC has provided a more enhanced coding efficiency compared to prior widely used standards such as MPEG-2. It is now successful over a wide span of applications that include video conferencing, broadcasting, surveillance and military application, and online video streaming [2]. The basic video coding design in H.264 is based on a conventional block based motion-compensated hybrid video coding concept, however with some important improvements over prior standards. Such improvements are found in the form of enhanced prediction capability, enhanced entropy coding methods, small block-size exact-match integer transform, etcetera. The enhanced algorithms utilised within the H.264/AVC standard can achieve up to a 50% bit-rate saving to provide a compressed video with perceptual quality equivalent to that of prior standards [26].

2.4.1.1 Structure of H.264 Codec

To address the need for customisability and flexibility of H.264/AVC across a broad variety of applications, and to ensure an efficient integration of network adaptation and video coding, the H.264/AVC structure is formed of two conceptual layers (see Figure 2-3). A video coding layer (VLC) provides an efficient representation of video content, and a network abstraction layer (NAL) converts the VCL video representation into a format suitable for enabling a seamless integration with specific transport layers or storage media. For circuit-switched transmission such as H.320, MPEG-2, and H.324/M, the NAL delivers the coded video as an ordered stream of bytes with headers attached so that the structure of the bit stream can be identified to the decoder. For packet switched networks like RTP/IP and TCP/IP, coded video packets are delivered without those headers [27].

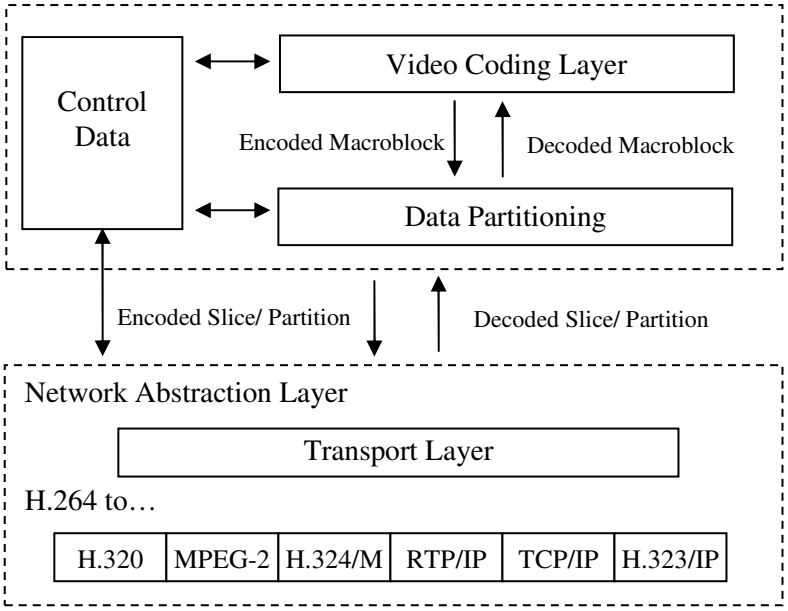


Figure 2-3: H.264/AVC conceptual layers [26]

In H.264/AVC, each picture can be compressed as one or more slices; each slice can be divided into macroblocks that consist of 16x16 luma samples with their corresponding chroma components. Furthermore, each macroblock can be divided into sub-macroblocks, which are used for motion-compensation prediction. For a more improved coding efficiency, those prediction blocks can be partitioned into 16x16, 16x8, 8x16 macroblocks, and 8x8, 8x4, 4x8, and 4x4 sub-macroblocks (see Figure 2-4).

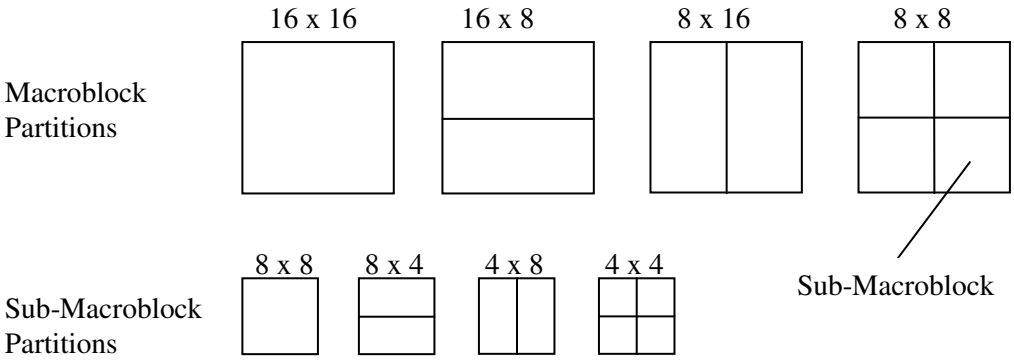


Figure 2-4: Partitioning of a macro and sub-macroblocks for motion compensated prediction [27]

The macroblock is the basic entity of the encoding or decoding process. In 4:4:4 format, each macroblock consists of 16x16 region of luma samples and two

other 16x16 chroma samples. In 4:2:2 format, each macroblock consists of one 16x16 luma samples and two corresponding 8x16 chroma samples. In 2:2:0 format, each macroblock consists of one 16x16 luma samples and two corresponding 8x8 chroma sample arrays [2]. It is worth noting that the terms “sample” and “pixel” are being used interchangeably in the context of this thesis.

2.4.1.2 H.264/AVC Profiles

To facilitate inter-operability between various application domains, three basic feature sets called profiles were defined in H.264/AVC. Each profile describes a set of coding tools or algorithms that are available within the standard to produce a bitstream that conforms to the requirements of the specified syntax, i.e. the binary codes and values that make up a conforming bitstream [6]. Those basic profiles are the Baseline, Main, and Extended profiles. The Baseline profile is mainly designed to minimise complexity and provide flexibility for use over a broad range of network environments with limited computing capabilities. The other two profiles were designed with more emphasis on coding efficiency capability and greater network robustness [2]. The contributions of this thesis are mainly based on the Baseline profile, details of coding tools used in this profile will follow in section 2.5.

2.5 Coding Tools

Unlike other coding standards, H.264 does not explicitly define a codec, but rather defines the syntax and semantics of the encoded bitstream and the method of decoding this bitstream, giving the freedom to the manufacturers to compete in cost and other hardware requirements. However, all standardised video coding techniques share the same hybrid video coding structure. Figure 2-5 shows a generalised structure of a hybrid video encoder.

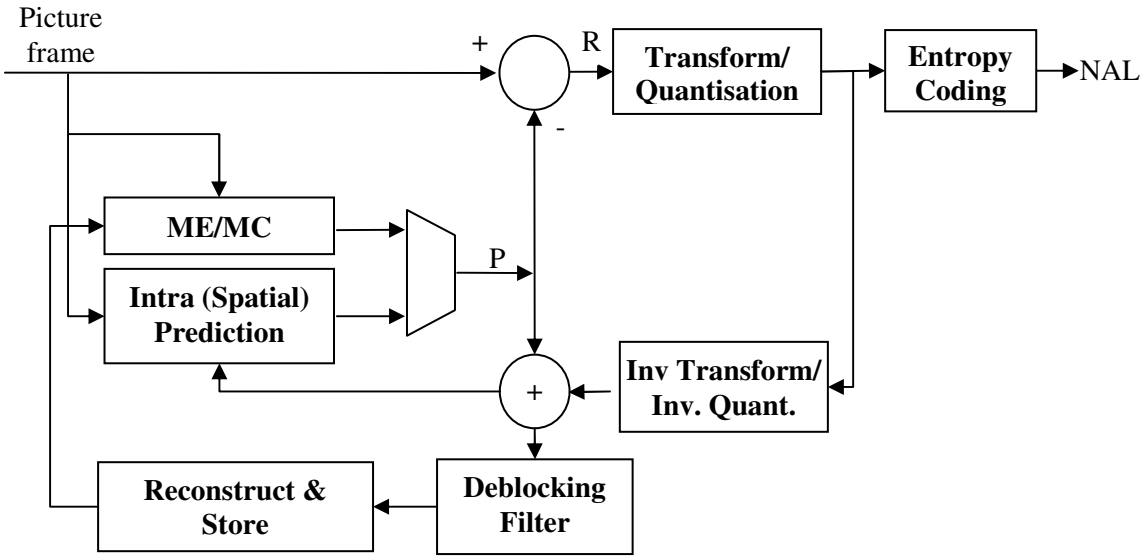


Figure 2-5: High-level encoder architecture

The input picture is partitioned into one or more slices and subsequently into macroblocks, each of which is either spatially or temporally predicted. The resulting prediction block (*P* in Figure 2-5) is subtracted from the original block to produce a residual (difference) block '*R*'. The residual block is then transformed using integer transform, and the transform coefficients are quantized and finally entropy coded. The resulting entropy coded data is passed to the NAL for transmission or storage. In motion compensated prediction, a copy of the encoded macroblock is reconstructed and stored in memory to be used in the prediction of macroblocks of subsequent frames. For this purpose, the quantised coefficients are inverse-transformed and added to the prediction signal. The resulting constructed macroblock is filtered in order to reduce the block-artefacts. The decoder (see Figure 2-6) receives the NAL data and initially uses entropy-decoding to obtain the quantized coefficient, "C". This data then follows a path similar to that described in the reconstruction part of the encoder, to finally obtain the reconstructed frame.

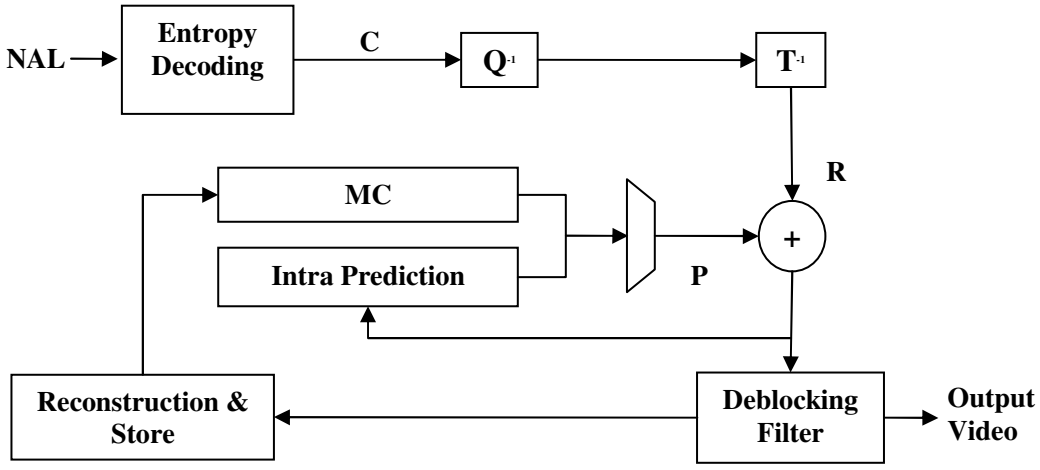


Figure 2-6: Hybrid video decoder

In general, slices of a video frame might be compressed using some/all of the following coding tools:

- Intra (spatial) prediction - block based.
- Inter (temporal) prediction - block based motion estimation and compensation.
- Interlaced coding features (Frame-field adaptation and field scan).
- Residual colour transform for efficient RGB coding.
- Scalar quantization.
- 8x8 or 4x4 integer inverse transform.
- Deblocking filter (within motion compensation loop).
- Coefficient scanning (Zigzag or field).
- Lossless Entropy coding.
- Error resilience tools.

Depending on its type, the above coding tools may or may not be used for each slice. A slice can be one of the following types: I (Intra), P (Predicted), B (Bi-predicted), SP (Switching P), or SI (Switching I). Pictures, which may contain different slice types, fall into two categories: reference pictures, used in inter-frame prediction, and non-reference pictures.

Within an index slice (I-slice), pixel values are first predicted from their neighbouring pixel values. After spatial prediction, the residual information is transformed then quantised (see Figure 2-7). The quantisation process supports

perceptual-based quantisation scaling matrices to optimise the quantisation process according to the visibility of the specific frequency associated with each transform coefficient. The quantised coefficients of the transform are scanned (zigzag or field scan) and then compressed using entropy coding. Temporal Prediction is only used for P and B macroblocks and not used for intra macroblocks. This is the main difference between I, P, and B macroblocks [2].

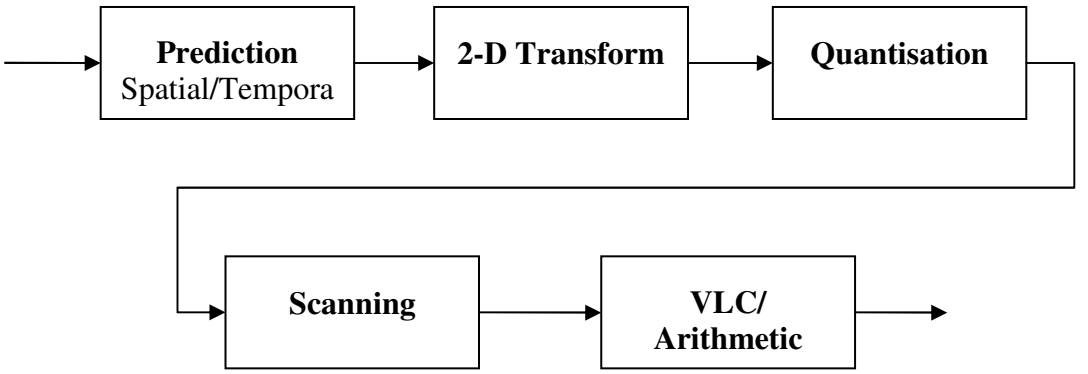


Figure 2-7: A block diagram for a video encoder [2]

2.5.1 Intra Spatial Prediction

In intra prediction, each prediction block is generated from the spatially neighbouring blocks that have been already coded within the same frame. H.264 provides three basic classes of intra spatial prediction, namely: Full-macroblock (INTRA-16x16), 8x8 luma, and 4x4 luma (INTRA-4x4) prediction. In Full macroblock prediction, pixel values for luma and chroma samples of the entire macroblock (16x16 pixels) are predicted from the previously coded neighbouring macroblocks. To perform full-macroblock prediction, the encoder selects one of four different prediction types: (i) horizontal, (ii) vertical, (iii) DC, and (iv) planner. In horizontal and vertical prediction, pixel values of a macroblock are predicted from pixels to the left of or above the macroblock, respectively. In DC prediction, pixel values of a macroblock are predicted by averaging the luma values of neighbouring pixels. In planner prediction, a curve fitting equation is used to form a prediction macroblock based on three parameters to approximate/match the neighbouring pixels. Those parameters are: brightness, slope in the horizontal direction, and slope in the vertical direction.

Alternatively, the encoder may select 4x4 luma intra prediction as the basis for predicting the pixel values of a macroblock. In this case, the selection is done on a macroblock-by-macroblock basis. In this prediction mode, the values of each 4x4 block of luma samples are predicted from the neighbouring pixels above or left of a 4x4 block. In INTRA-4x4, a macroblock (i.e. 16x16 pixels) is divided into sixteen 4x4 sub-blocks and the luma signal for each of the sub-blocks is predicted individually. A total of nine possible prediction modes are used based on nine different directional ways of performing the prediction. Figure 2-8 illustrates the nine prediction directions.

In 2003, a new set of extensions to the H.264 standard known as Fidelity Range Extensions (FRExt) were approved. In FRExt profiles, 8x8 intra prediction can be selected. This prediction mode uses the same concepts as 4x4 prediction, however incorporates block size of 8x8 rather than 4x4 [2].

On the other hand, the prediction type for chroma samples is selected independently of the prediction type for the luma samples. Chroma intra prediction always operates using full-macroblock prediction. This is due to the fact that the size of chroma arrays for the macroblock are different in different chroma formats (i.e. 4:2:0, 4:2:2, and 4:4:4).

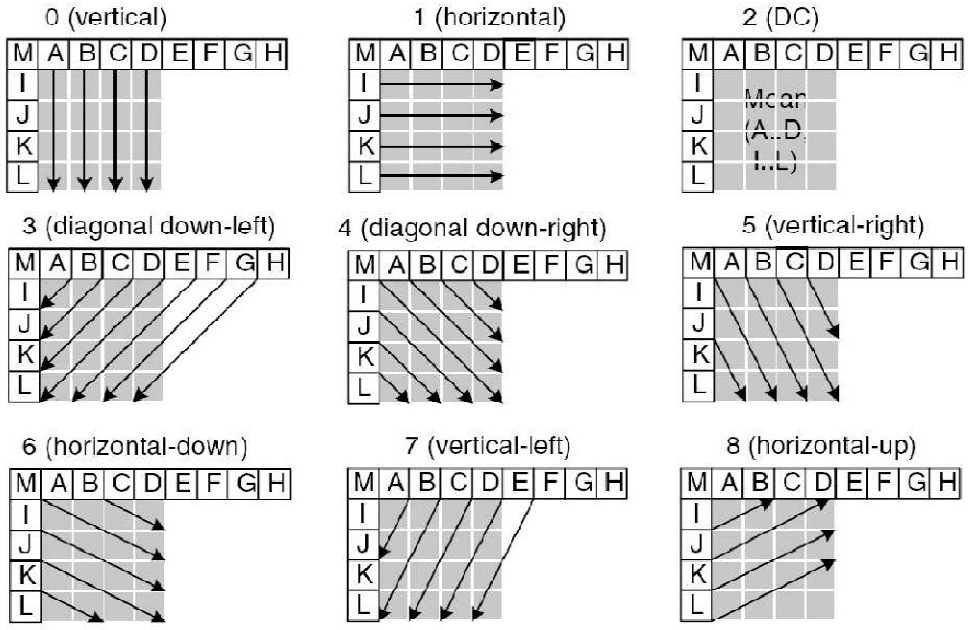


Figure 2-8: INTRA 4x4 prediction modes [6]

2.5.2 Inter Prediction (Motion Compensated Prediction)

Inter prediction, also known as block based motion compensation, is used to reduce the temporal redundancy in successive frames based on predicting macroblocks from a previously transmitted reference frame. For instance, H.264/AVC adopts block-based motion estimation and compensation for removing the redundancy between frames. Within this approach, each M x N block in the current frame is compared with blocks of similar size within a predefined search region of the reference frame. This aims to obtain the closest match for the M x N block from the corresponding reference frame. The technique of searching for the closest match is known as Motion Estimation (ME), which is discussed later in this section. The matching block is then subtracted from the current block to produce a residual block *R* that is encoded and transmitted along with the corresponding motion vector difference (MVD) describing the residual between the current motion vector and a predicted motion vector.

Inter prediction takes place in P-slices (predicatively-coded slices). Motion can be estimated at full-macroblock level (16x16) or by dividing the macroblock into "macroblock partitions" which corresponds to luma sizes of 16x16, 16x8, 8x16, and 8x8, and sub-macroblocks which corresponds to luma sizes of 8x8, 8x4, 4x8, or 4x4 (see Figure 2.4). For each sub-macroblock partition, a distinct motion vector can be transmitted. Each motion vector is coded and transmitted along with the choice of partitions [6]. Motion can be estimated from pictures that lie either in the future or in the past in display order. The selection of which reference frame is used is done on a macroblock partition level. To estimate motion, pixel values are first interpolated to achieve quarter-pixel accuracy for luma and up to one-eighth pixel accuracy for chroma. After interpolation, block-based motion compensation is used [2].

If the motion characteristics of a macroblock indicate that its motion can be predicted effectively from the motion of neighbouring macroblocks, and contains no none-zero quantised transform coefficients, then this macroblock is flagged as skipped. Motion vector and reference frame indexes representing the estimated motion are compressed. The compression of a motion vector is done by taking the

median of the motion vectors of three neighbouring partitions, and then the difference from this median and the value of the current motion vector is obtained and entropy coded [2]. The prediction of the current frame content in the P-slice is also achieved with the help of weighted prediction, where weights can be applied to the motion compensated prediction before it is used to predict the current frame [6].

Unlike P-slices the process of temporal prediction in B-slices (or B-Frames) is slightly different, where two motion estimation vectors are produced per macroblock partition. Those motion vectors can be estimated from any reference frame (I-Frame) in the future or the past. The weighted prediction concept is also used in case of B-slice, although further extended to enable some encoder adjustments to the weighting coefficients used in the weighted average between the two predictions that apply to bi-prediction [2].

2.5.3 Transform Coding and Quantisation

After spatial prediction, transform coding is applied to code the prediction error signal (Residual block "R") in order to reduce the spatial redundancy of the prediction error signal. In other words, transform coding is used to reduce the statistical correlation of the input signal.

In the past, all compression standards applied two dimensional 8x8 Discrete Cosine Transform (DCT). In H.264/AVC the size of these transforms is 4x4 and in special cases 2x2. The use of 4x4 transform instead of 8x8 has three advantages: Firstly, it enables the encoder to efficiently adapt the prediction error coding to the boundaries of the moving objects. Secondly, it enables the encoder to match the transform block size with the smallest block size of the motion compensation. Thirdly, it enables the encoder to adjust the transform to the local prediction error signal [27].

There are three different types of transforms. The first is a 4x4 transform. This is applied to all samples of all error prediction blocks of both luma and chroma components. This type uses Hadamard transform and could be used with intra prediction or motion compensated prediction; its transformation matrix is called H1 (see Figure 2-9). The second type is also a 4x4 transform. It applies Hadamard

transform with matrix H2 (see Figure 2-9) in conjunction with H1 if the macroblock is predicted using Intra_16x16. It is used to transform the 16 DC coefficients of the luminance signal. The third type applies Hadamard transform with a 2x2 H3 matrix (see Figure 2.9) to transform 4 DC coefficients of each chrominance component [27].

$$H_1 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix} \qquad H_2 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} \qquad H_3 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

Figure 2-9: Hadamard transform matrices used in H.264

2.5.3.1 Quantisation

Quantisation reduces the precision by which a sample (or group of samples) is represented. In general, quantisation aims to reduce the amount of data needed to encode the data representation. All the coefficients of a macroblock are quantised by a scalar quantiser. The basic quantiser is in the form:

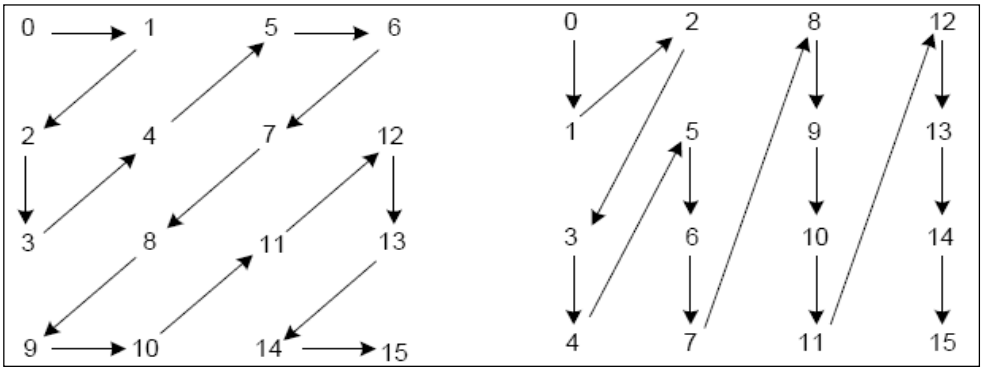
$$Z_{ij} = round(\frac{Y_{ij}}{Q_{step}}) \tag{2.6}$$

Where, Y_{ij} is a coefficient of the transform described above. Q_{step} is a quantiser step size and Z_{ij} is a quantised coefficient. The quantization step size is indexed by a Quantisation Parameter (QP) which supports 52 different quantisation steps. The step size doubles every 6 increments of QP. The required data rate decreases by about 12.5% when the QP increments by 1[28]. Among all other coding tools, quantisation is typically the only one that inherently involves some loss of fidelity [1]. The wide range of QP makes it possible for the encoder to efficiently control the trade-off between the bitrate and quality [6].

2.5.4 Scanning

When a macroblock is compressed using a 4x4 transform, each 4x4 block of quantised coefficients is mapped to a 16 element array. In the case of frame mode compression, the quantised coefficients of the transform are scanned in zigzag form

(Figure 2-10a), this scan ordering is designed to maximise the number of consecutive zero-valued coefficients and to order the highest variance coefficients first (Sullivan et al, 2004). On the other hand, in the case of field mode compression, the quantised coefficients of the transform are scanned the way shown in figure 2.10b.



(a) **(b)**

Figure 2-10: Coefficient scanning order in (a) Frame and (b) Field modes [2]

2.5.5 Entropy Coding

Entropy coding is a lossless coding technique, in which data elements are replaced with coded representations. Two modes of entropy coding are used in the H.264/AVC standard: Context Adaptive Variable Length Coding (CAVLC) and Context Adaptive Binary Arithmetic Coding (CABAC). The CAVLC is a low complexity technique, while CABAC is computationally a more demanding algorithm. Both techniques represent substantial improvements in terms of coding efficiency compared to old techniques of statistical coding. Entropy coding, along with predictions, transformation and quantisation, can reduce data size significantly.

CAVLC is the baseline entropy coding method of H.264/AVC. The idea of VLC, also known as Huffman coding, is that when data elements occur with unequal frequencies, very short codes will be assigned to the most frequent elements while longer codes will be assigned to the less frequent elements. In typical conditions, CAVLC can provide bit rate reductions of 2-7% compared to the traditional statistical coding techniques such as VLC [27]. CABAC is the alternative

entropy coding mode of H.264/AVC where a significantly improved coding efficiency is achieved at the cost of additional complexity. As depicted in Figure 2-11, the key elements of CABAC are: binarisation, context modelling, and arithmetic coding.

CABAC usually encodes a broader range of syntax elements than CAVLC. Among the several syntax elements coded with CABAC are: the macroblock type, the intra prediction modes, motion vectors, reference frame indexes, and residual transform coefficients, whereas the transform coefficients on their own are adaptively coded with CAVLC [2]. Typically, CABAC provides bit rate reductions of 5-15% compared to CAVLC [27]. More details on CABAC can be found in [29].

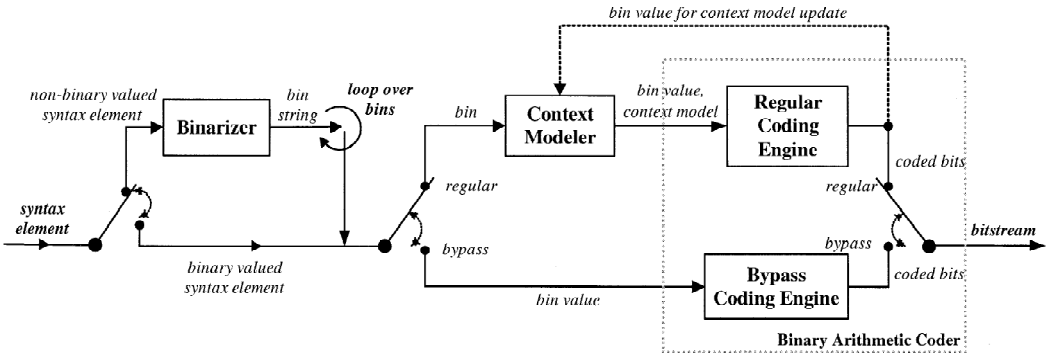


Figure 2-11: CABAC encoder block diagram [29]

2.6 Optimisation Areas for H.264 Video Codec

As identified in Section 2.4, the growing interest in digital image and video applications has made video coding a very active field of research and development. Modern video coding techniques have provided a more enhanced coding efficiency compared to prior widely used standards such as MPEG-2. The added features and functionalities within H.264/AVC, as discussed in Section 2.5, have provided a marked improvement in coding efficiency. However, all of the ITU-T and ISO/IEC video coding standards have only defined the decoding process by imposing restrictions on the syntax and bitstream, while the encoding process was out of the scope of the H264/AVC standard, and subsequently left undefined. This limitation has allowed a high degree of flexibility to optimize implementations in a manner appropriate to specific applications (balancing compression quality, implementation

cost, etc.). However, it provides no guarantees for a high-quality reproduction of video streams, as it allows even inefficient encoding techniques to be considered conforming [1]. In addition, the aforementioned added features and functionalities, as well as the enhancements on coding efficiency have all come at the expense of increased demand on system resources due to increased computational complexity and memory requirements.

The operation of video encoders requires the optimisation of numerous decision parameters to achieve the best trade-off between bit rate and quality given the resource limitations arising from operational constraints such as memory and complexity. There has been a significant amount of research on the aforementioned optimisation problem. Sullivan and Wiegand [1], stated that one area of particular interest has been the Lagrangian optimisation methods (e.g. [30], [31], [32]). Some other studies have focused on reducing the complexity while minimising the loss in quality, while others have developed sophisticated encoder optimisation strategies with little regard for encoding complexity (e.g. [33], [34], [35], [36], [37], [38])

2.7 Conclusion

This chapter has introduced the fundamental concepts of video coding, followed by a discussion on visual quality assessment for compressed video sequences highlighting both the subjective and the objective assessment methods. State of the art video coding techniques were discussed giving particular emphasis to the H.264/AVC video coding standard. Various video coding tools were presented and discussed. The chapter concludes with highlighting the need for enhancing the performance of video coding techniques through the optimisation of the various coding parameters. More details on the optimisation of the performance of video codecs, in particular the H.264/AVC, are presented throughout this thesis.

Most of the examined video standards do not explicitly define a codec, but rather defines the syntax and semantics of the encoded bitstream and the method of decoding this bitstream; this ensures interoperability, giving the freedom to the manufacturers to compete in cost and other hardware requirements. However, the

visual quality aspect has been left un-standardised, with no guarantees of the end-to-end reproduction quality of video sequences.

Optimising the performance of video codecs often involves finding solutions for multiple conflicting objectives, e.g. rate-distortion optimisation. A number of video compression optimisation models have been proposed in literature, but very few have solved for multiple objectives. More discussion on the optimisation methods for video codecs and the use of multi-objective optimisation models are presented in Chapter 3.

Chapter 3

Evolutionary Multi-Objective Optimisation Techniques

3.1 Introduction

Chapter 2 has highlighted how digital video technologies have become an integral part of the way we handle visual information. This has been seen in many application scenarios ranging from broadcast and terrestrial transmission to Internet video streaming.

The basic challenge of video codec design maybe presented as conveying the source data using the lowest bit rate possible whilst maintaining the video stream at specified reconstruction fidelity, or it may be posed as delivering source data with the highest fidelity possible within an available bit rate. In either case, a fundamental trade-off is made between fidelity and bit rate. The ability of the codec to optimise this trade off is referred to as its coding efficiency. It is also referred to as rate distortion performance.

Thus, video codecs are primarily characterised in terms of the distortion of the decoded video, and the throughput of the channel. Furthermore, there are additional factors that play essential roles in influencing the performance of a video codec; such factors include the delay, such as buffering and processing delays, and the complexity of the video codec in terms of capacity and memory access requirements.

This chapter reviews the existing optimisation methods for video coding including algorithm based and parameter based optimisation approaches. This is followed by a detailed introduction to the principles of multi-objective optimisation and a review of its existing approaches including aggregation, population, and Pareto-based approaches. The review then concludes with an overview of the application of multi-objective optimisation for enhancing the performance of video coding.

3.2 Optimisation methods for Video Coding

As illustrated in Chapter 2 (Section 2.6), many optimization methods have been proposed in the literature, some studies have focused on reducing the complexity while minimising the loss of quality, while others have focused on developing encoder optimisation strategies with little regard for encoding complexity. These optimisation approaches can be classified into two categories, algorithm-based optimizations and parameter-based optimizations. The algorithm-based optimization methods focus on the direct performance optimization of a given algorithm. Alternatively, parameter-based optimization methods optimize given objectives through the optimal selection of coding parameters. A comprehensive literature review on optimization of video coding has been conducted by [39]. The review highlights that most optimization research works have focused on algorithmic enhancements/improvements as compared to relatively few that have focused on parameter-based optimisation. The review has also highlighted that a number of optimization studies [40], [41], [42], [43], [44], [45], [46], [47] have focused attention on the H.264 video codec. These approaches have mainly focused on algorithmic improvements to enhance the performance of the H.264 video codec

with little emphasis on parameter based performance enhancement. The following sections provide a brief review on algorithm and parameter based optimisation.

3.2.1 Algorithm-Based Optimisation

As identified by [39], there has been a substantial amount of research conducted on the optimisation of video coding. This research has focused on optimising the performance of video codecs based on single and multiple objectives. For example, [46] has proposed a single objective optimisation algorithm to reduce the computational complexity for the H.264 encoder. On the other hand, [40] has proposed a single objective algorithmic enhancement to reduce the number of memory accesses during the decoding process, leading to the optimisation of memory usage by the H.264 video decoder.

A large amount of research has been conducted on two-objective optimisation approaches. For example, [45], [42], [44], [37], [38], [34], [33], have evaluated the cost of using various possible coding modes and the corresponding motion vectors to achieve the best trade off between and distortion and bit rate (rate-distortion optimisation). Other approaches have focused on three and four-objective optimisation techniques, where various combinations of objectives including power, rate, distortion, memory, and complexity are analysed in order to optimise the performance of video codecs [48], [45], [44], [39].

3.2.2 Parameter-Based Optimisation

An area to receive less attention in video codecs performance optimisation research involves the optimisation of coding parameters. Only a few examples in literature have focused on parameter-based performance optimisation [49], [39]. Kwon *et al*. [49] have proposed a parameter-based method for the joint optimization of computational complexity and distortion in H.263 video coding, while Li [39] has proposed a joint complexity-memory-rate-distortion optimization of H.264 video codec.

In parameter-based optimisation, the selection of various combinations of coding parameters can compromise the performance of video codecs due to the

selection of inappropriate coding parameters and/or parameter values. Hence, the selection of the right parameter set and the optimum values are of utmost importance. Although parameter based optimisation approaches for the H.264/AVC video codec have been proposed in the literature, these methods largely focus on the joint optimisation of complexity and distortion. From the authors review, only one study [39] has suggested that other factors such as bit-rate and memory usage should be considered in this optimization model.

3.3 Multi-Objective Optimisation

In mathematics, the definition of optimisation might be found to have several interpretations, but all refers to finding the minima and maxima of a function, or in other words, finding one or more "optimum" values (solutions) for one or more objective functions [50]. In engineering and computer sciences, the definition of optimisation tends more towards improving the system to reduce resources' consumption, e.g. cost, bandwidth or memory requirements... etc.

Most real-world engineering and scientific problems have multiple conflicting objectives. In most cases, solving optimisation problems with multiple conflicting objectives is a difficult process that might be computationally expensive [50]. However, it should be noted that a perfect multi-objective optimisation solution that satisfies all objective functions, and complies with all constraints associated with the decision variables, may not even exist [51].

3.3.1 What is Multi-objective Optimisation?

Osyczka (1985) [52] has defined multi-objective optimisation as the process of "finding a vector of decision variables which satisfies constraints and optimises a vector function whose elements represent the objective functions". These functions form a mathematical description of performance criteria which are usually in conflict with each other and finding such a solution which would give the values of all the objective functions acceptable to the designer."

In single-objective optimisation, the search space is well defined and usually yields a unique optimal solution. In contrast, multi-objective optimisation problems have several possible contradicting objectives to be simultaneously optimised. Therefore, rather than obtaining a single optimal solution, a whole set of possible optimal solutions might be obtained. Consequently, it is up to a decision maker to pick the solution out of a set of optimal trade-offs between the conflicting objectives [53].

In multi-objective optimisation problems, a number of objective functions are to be minimised or maximised, and the optimal set of solutions must satisfy a number of constraints. For example, a multi-objective optimisation algorithm minimises/maximises k objective functions $F(X) = (f_1(X), \ldots, f_k(X))$ subject to m constraints ($g_i(X) \leq/\geq 0$, $i = 1,\ldots, m$) where X is an n-dimensional decision variable vector [51].

In other words, we are interested in finding the vector $X^* = [x_1^*, x_2^*, \ldots, x_n^*]^T$ that satisfies the equality/ inequality constraints

$$g_i(X) \geq 0, \; i = 1,\ldots, m \tag{3.1}$$

$$g_i(X) = 0, \; i = 1,\ldots, p \tag{3.2}$$

And optimises the vector function

$$F(X) = [\, f_1(X), f_2(X), \ldots, f_k(X) \,]^T \tag{3.3}$$

Where $X = [x_1, x_2, \ldots, x_n]^T$ is the decision variables vector [54].

3.3.2 Goals of multi-objective optimisation

There are two main goals of multi-objective optimisation:

1. To guide the search towards the Pareto optimal front. In other words, to find a set of solutions as close as possible to the Pareto front.

2. To find a diverse set of solutions to achieve a well distributed trade-off front.

The first goal states that the search should converge to the true Pareto optimal front. The approaches to this convergence will be discussed in the next section. The second goal states that the solutions in the Pareto optimal front should be sparsely spaced, therefore, among the objective space; we can get a good set of trade-off solutions. Diversity can be assured either in the decision variable space or in the objective space, or in both. In most cases, diversity in one space guarantees the diversity in the other space. Furthermore, two solutions are found to be diverse if their Euclidean distance is large [50].

3.4 Basic Principles of Multi-Objective Optimisation

3.4.1 Pareto optimality

As discussed in the previous section, when dealing with multi-objective optimisation problems, we usually look for trade-offs rather than a single optimal solution. Therefore, the concept of optimality is different in this case. One of the most common notions used to describe this set of optimal solutions is Pareto optimality. This notion was formulated by Vilfredo Pareto in the 1890s [55].

Assuming that our optimisation problem is a minimisation one, a vector of decision variables $X^* \in F$ is called a Pareto optimal if there does not exist a vector $X \in F$ such that $f_i(X) \leq f_i(X^*)$ for all $i = 1, 2, \ldots, k$ and $f_j(X) < f_j(X^*)$ for at least one j. The vectors X* corresponding to Pareto optimal solutions are called *non-dominated* solutions [53] [54]. A curve that connects all of these Pareto optimal solutions is called a *Pareto Front* as shown in Figure 3-1.

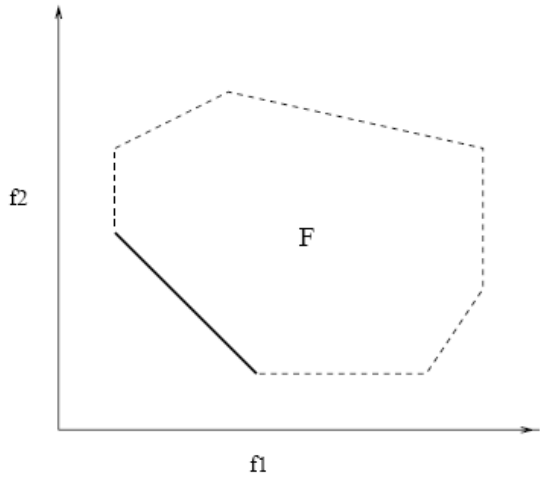


Figure 3-1: an example of a minimisation problem with two objective functions. The Bold line is the Pareto front [54]

3.4.2 Approaches to Multi-Objective Optimisation

As discussed in the previous section, the convergence of solutions towards the Pareto optimal front is the key goal of multi-objective optimisation. This process of convergence involves some evolutionary techniques such as fitness assignment and selection. For about two decades, various evolutionary approaches have been introduced to multi-objective optimisation. Evolutionary algorithms have made the simultaneous search for multiple solutions possible. Hence, there has been a growing interest in solving multi-objective optimisation problems using EA's [56].

The first introduction of a Multi-Objective Evolutionary Algorithm (MOEA) was in the mid of 1980's and aimed to solve problems in machine learning. Afterwards, the MOEAs where roughly divided into two categories: Aggregation and non-aggregation approaches. The non-aggregation approaches were in turn divided into: Population-based and Pareto-based approaches. These approaches are discussed in more details in the following sections.

3.4.2.1 Aggregation approaches

These are the simplest approaches to multi-objective optimisation. They are based on combining all objectives into a single objective, using any arithmetic

operation such as addition or multiplication. One disadvantage of this approach is that scalar fitness information needs to be passed to the genetic algorithm (GA) in order to function. This implies that the behaviour of each objective function should, to some extent, be known [54].

One of the most quoted examples of this approach is the Weighted Sum approach, in which, all the weighted objective functions are added linearly together. This transforms the problem into a scalar optimization problem of the form:

$$\sum_{i=1}^{k} w_i f_i(\vec{x}) \tag{3.4}$$

Where w_i are the weighting coefficients [54].

3.4.2.2 Population-Based Approaches

These approaches have been developed to overcome the difficulties of the aggregation approaches. They are based on diversifying the population of an EA. In population based approaches, the selection process does not include the concept of Pareto dominance [57]. The most famous example of this approach is called Vector Evaluated Genetic Algorithm (VEGA) which was proposed by Schaffer (1984) [58] and that was the first real implementation of an MOEA. In VEGA, the GA was modified by performing independent selection cycles to each objective function with the help of crossover and mutation [50]. The main disadvantage of this approach is that the Pareto dominance is not considered in the selection process.

3.4.2.3 Pareto-Based Approaches

Pareto-Based approaches were developed to overcome the drawback caused by the absence of Pareto dominance from VEGA algorithms. The basic idea behind the Pareto-based approaches is to find members of the population that are not dominated by other members of the same population. This set of “non-dominated” members will be assigned the highest rank and preserved, while another set of non-

dominated members will be determined from the remaining population and assigned the next highest rank [54].

One of the most important Pareto-based algorithms is the Non-Dominated Sorting Genetic Algorithm (NSGA) proposed for the first time by Srinivas and Deb (1994) [59] . This algorithm is based on several layers of hierarchal classification of the individuals. The selection process is preceded by ranking the population on the basis of non-domination. Then the set of non-dominated individuals is ranked with a dummy fitness value. Then this group is preserved and another layer of non-dominated individuals is ranked.

NSGA has been criticised for the lack of elitism, the computational complexity, and the choice of the sharing parameter σ_{share}, which in turn leads to two problems: First, the chosen value of σ_{share} determines the performance of the sharing function in maintaining diversity among solutions. Second, the overall complexity of the sharing function increases as each solution must be compared with all other solutions [60].

3.5 Elitist Non-Dominated Sorting Genetic Algorithm (NSGA-II)

To overcome the aforementioned drawbacks of NSGA, Deb et al. (2002) [60] proposed a modified version called NSGA-II, which is more efficient, uses elitism, incorporates an improved sorting algorithm, and no sharing parameter needs to be specified a priori. NSGA-II also uses the same explicit diversity-preserving mechanism defined in [61]. In this section, NSGA-II is discussed in some detail as it forms part of the core of the research detailed in this thesis.

Figure 3-2 illustrates the pseudo-code for the NSGA-II as described by [60]. The illustrated algorithm is based on evolutionary processes for finding the optimal set of solutions for identified objective functions. The algorithm is first initialised by defining the population size, the total number of generations, and the number of decision variables. Once the population is initialised, it is sorted into fronts based on non-domination. For each individual p, two measures are calculated: first, the number of individuals, n_p, that dominate p, and second, the set of individuals (S_p)

that *p* dominates. The population initialisation and sorting of the population is summarised in the following algorithm:

- for each individual p in the main population *P*:
  1. Initialize $S_p = \Phi$. *The set of individuals dominated by p.*
  2. Initialize $n_p = 0$. *Individuals that dominate p*.
  3. for each individual *q* in *P*:

     If *p* dominates q then

     Add *q* to the set *Sp* i.e. $S_p = S_p \; U \; \{q\}$

     Else if *q* dominates *p* then

     Increment the domination counter i.e. $n_p = n_p + 1$
  4. If $n_p = 0$ then p belongs to the first front; Set the rank (fitness) of individual p to one ($p_{rank} = 1$). Update the first front set by adding *p* to front one i.e. $F1 = F1 \; U \; \{p\}$
- This is carried out for all the individuals in main population P.
- Initialize the front counter to one. i = 1
- Following is carried out while the i^{th} front is nonempty i.e. $F_i \neq \Phi$
  1. Q = Φ. The set for storing the individuals for $(i + 1)^{th}$ front
  2. for each individual p in front F_i
     - for each individual q in S_p (S_p is the set of individuals dominated by p)
       a. $n_q = n_q$-1, decrement the domination count for individual q.
       b. if $n_q = 0$ then none of the individuals in the subsequent fronts would dominate q. Hence set $q_{rank} = i + 1$. Update the set Q with individual q i.e. Q = Q U q
  3. Increment the front counter by one.
  4. Now the set Q is the next front and hence F_i = Q.

Figure 3-2: pseudo-code illustrating the operation of the NSGA-II

As discussed before, two main factors should be taken into consideration when dealing with an MOEA: convergence towards the optimal set of solutions, and diversity (spread) of solutions. In addition to fitness assignment, and to preserve diversity, NSGA-II incorporates a new parameter called “crowding distance”. The crowding distance requires information on the density of individuals surrounding a

particular point in the population. This is done by calculating the distance between two points on either side of the point of interest along each of the objectives [60].

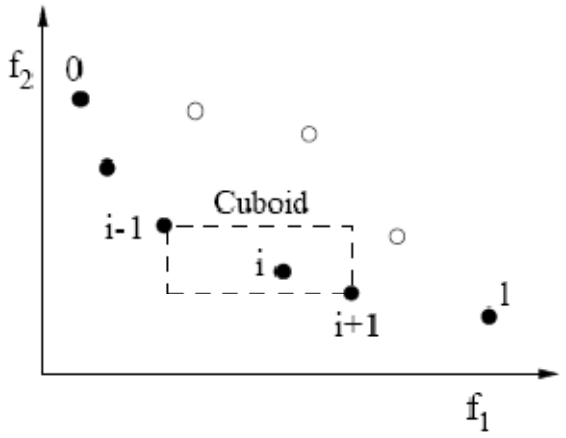


Figure 3-3: Calculation of crowding distance [60]

The crowded-distance operator ensures that for two solutions with different non-domination ranks, the crowding distance has no effect. However, when dealing with solutions in the same front, those located in a lesser crowded region are preferred. Therefore, the diversity among non-dominated solutions is ensured by crowding distance operators [50].

Based on rank and crowding distance, binary tournament selection selects parents from the population. Then, crossover and mutation operators generate off-springs which are added to the current population to form 2N individuals. The new population is sorted again based on non-domination rank and the best N individuals are selected based on their ranks and crowding distance [61].

3.6 The application of MOEAs on video coding and transmission

There is relatively little research literature available on the application of multi-objective optimisation on video coding and wireless transmission problems.

One of the main goals of any wireless systems engineer is to achieve transmission at the highest throughput with the maximum quality regardless of whether talking about service quality or viewability quality (in case of video and broadcasting). One solution could be to increase the buffer occupancy levels for each user while increasing the overall channel throughput. Other approaches such as Leaky Bucket, have implemented an algorithm to check that data transmission conform to a defined limit on bandwidth [85]. However, wireless communication channels are still very limited in bandwidth, and therefore a way to get the best possible trade-off between size and quality has to be found.

Therefore, a dual optimization problem is faced, in which the objective is to try and reduce the buffer and memory requirements, while maintaining transmission at high quality levels. In the case of video streaming or broadcasting, the objective is to look forward to transmitting high quality videos with the minimum possible bitrate, or, to have an adaptive coding scheme, in which, frames with high importance are coded at a higher bitrate than low importance frames. Chapter 2 has discussed the H.264 video compression technique and adaptive bitrate coding in some detail.

3.7 Conclusion

This chapter has reviewed the existing optimisation methods for video coding including algorithm based and parameter based optimisation approaches. This was followed by a detailed introduction to the principles of multi-objective optimisation and a review of its existing approaches including aggregation, population, and Pareto-based approaches. The review concluded with an overview of the application of multi-objective optimisation for enhancing the performance of video coding.

From this review, the significant importance of parameter based optimisation has been identified. However, it has been shown that the application of parameter-based optimisation can largely compromise the performance of video codecs due to the selection of inappropriate parameters and/or parameter values. In

order to address this gap in the existing approaches, this thesis proposes the development of a parameter based, multi-objective framework for enhancing the performance of the H.264/AVC video codec.

The following chapters detail more in-depth analysis of the H.264 video codec, followed by the design, implementation, and evaluation of the aforementioned optimisation framework.

Chapter 4

Performance Analysis of the H.264 Video Codec

4.1 Introduction

Chapter 2 has demonstrated that the growing interest in digital image and video applications over the past two decades has made video coding a very active field of research and development. Codecs such as H.264/AVC (see Chapter 2, Section 2.4) have provided a more enhanced coding efficiency compared to prior widely used standards such as MPEG-2. It is now successful over a wide span of applications including video conferencing, broadcasting, surveillance and military applications, and online video streaming [2]. The added features and functionalities within H.264/AVC (see Chapter 2, Section 2.5) have provided a marked improvement in coding efficiency. However, all of the ITU-T and ISO/IEC video coding standards have only defined the decoding process by imposing restrictions on the syntax and bitstream, while the encoding process was out of the scope of the H264/AVC standard and subsequently left undefined. This limitation has allowed a high degree of flexibility to optimize implementations in a manner appropriate to specific applications (balancing compression quality, implementation cost, etcetera).

However, it provides no guarantees for a high-quality reproduction of video streams, as it allows even crude encoding techniques to be considered conforming [1]. In addition, the aforementioned added features and functionalities and the enhancements on coding efficiency have all come at a price. For example, the focus on coding efficiency has resulted in an increased demand on system resources as a result of increased computational complexity and memory requirements.

Coding efficiency is also dependent on a range of different parameters that can be used to set the functionality of the H.264 video CODEC. In this chapter, coding parameters, which significantly affect coding efficiency, are identified. This analysis is based on the H.264 Main profile. The rest of this chapter is organized as follows. The video test sequences used in our experiments are introduced in Section 4.2. A brief overview of the coding parameters used in our experiments is given in Section 4.3. A comprehensive analysis of the encoder's computational complexity is presented in Section 4.4, while Sections 4.5 and 4.6 provide an in-depth analysis for the rate-distortion characteristics and memory utilisation of the H.264 video encoder. Finally Section 4.7 concludes this chapter.

4.2 Video test sequences

Coding efficiency and the effectiveness of video encoders depends to a great extent on the content of the source video. In this analysis video sequences relating to five different scene categories were chosen with distinct content and motion characteristics so that the results could reflect some generality. The selected video categories were considered from the Minerva Video Benchmark [62], including videos of news, landscapes, traffic, sports, and videos captured by day and night-vision cameras mounted on an unmanned vehicle. This dataset reflects the range of characteristics in the general population of compressed videos. It was ensured that no particular sub-category was under- or over-presented. Image format for all of the selected video sequences is the standardised CIF resolution (368x272 pixels). Each sequence is in 4:2:0 sampling format (see Section 2.2.4) and has around 248 frames.

"News" and "Landscape" video sequences are characterised by minimal motion in the background and simple motion of the foreground, with two different

light environments. The sequences "Traffic" and "Sports" show fast motion on both foreground and background. While the UMV video shows a moving background and foreground as a result of moving the vehicle on which the camera is fixed. The above video sequences represent a wide range of videos with different properties and behaviours, from moderate to high movement; from low to highly detailed scenes, and from fixed to changing background. Table 4-1 shows a selected frame of each video sequence.

Table 4.1: Sample frames from the video test sequences belonging to the 5 scene categories

(a) News

(b) Landscape

(c) Traffic

(d) Sports

(e) UMV

4.3 Video Coding Parameters

This section investigates compression parameters that have a significant impact on the encoder's computational complexity and memory utilisation. This analysis is based on H.264/AVC JM Reference Software [63]. For clarity, the investigated coding parameters are explained below and presented in Table 4-2.

- **Resolution**: Video frames are captured and converted to a set of intermediate digital video formats (see Chapter 2, Section 2.2.5) prior to compression and transmission.

- **Number of Intra Frames**: H.264/AVC allows for the use of multiple reference frames. In this case, the video encoder chooses between a number of previously decoded frames and uses this choice to reconstruct each macroblock in the next frame. It is worth noting that different macroblocks in the same frame can be based on different reference frames.

- **Use of Fast Motion Estimation**: This parameter defines which motion estimation algorithm to be used during the encoding of a video stream. This involves the analysis of previous and next frames to identify blocks that have moved location during the encoding process.

- **Quantisation Parameter (QP)**: This parameter controls the trade-off between quality and bit rate in the sense that a QP increment by 1 results in 12.5% reduction of bit-rate [6]. Three different values for QP were selected for our experiments, namely: 30, 35, and 40. Those values for the QP were selected based on the observed variations on visual quality of the compressed videos. This is discussed in more details in Chapter 6.

- **Intra-Frame Period**: This parameter determines how often a reference frame (I-frame) appears in the video sequence. Two different values of Intra-frame period are used in the experiments as shown in Table 4-2. Values for the Intra frame period are chosen based on empirical experiments and are proven to have the most effect on the compressed videos' size and quality.

- **Number of B-frames**: B-frames are commonly referred to as bi-directional interpolated prediction frames. This parameter sets the number of consecutive B-frames to be inserted within a video sequence.

- **Search Range**: This parameter defines the search window size used by the motion estimation algorithm for an inter prediction macroblock.

Table 4-2: Investigated coding parameters and respective ranges of values

Coding Parameter	Value Range	Default
Resolution	QCIF, CIF	QCIF
Number of Reference Frames	1-5	1
Fast Motion Estimation	0-3	0 (Disabled)
I-Frame Period	2-3	0 (only first frame)
Number of B-Frames	1-2	1
QP for I-Slice	0-51	28
QP for P-Slice	0-51	28
QP for B-Slice	0-51	28
Inter Block Search	0-1	1 (on)
Intra Block Search	0-1	0 (off)

4.4 Computational complexity for the H.264/AVC Encoder

As presented in the introduction of this chapter, the H.264/AVC video coding standard guarantees improved coding efficiency over existing video coding standards through added features and functionalities. However, such features and functionalities also entail additional complexity in encoding and decoding. The computational complexity of the coding algorithms directly affects the cost effectiveness of the development of a commercially viable H.264/AVC-based video solution.

To estimate the computational complexity of an H.264/AVC encoder implementation, it is important to understand its two major components [64]: time complexity and space (or storage) complexity. Time complexity is measured by the approximate number of operations required to execute a specific implementation of an algorithm. This can be achieved by estimating the number of CPU cycles required to perform key encoding functions. Storage complexity is measured by the approximate amount of memory required to implement an algorithm. Storage complexity and memory utilisation will be discussed in further details in section 4.5.

In this section, the computational complexity of the H.264/AVC encoder is studied and analysed in the context of software implementation on a PC with an Intel P4-2800MHz processor. A number of experiments were carefully designed to identify the encoding parameters that have a significant impact on CPU utilisation. Throughout this analysis, it has been assumed that the network does not introduce any data loss or delay. Therefore, the quality of the video received at the decoder is assumed to be the same as that at the encoder terminal.

In order to estimate the time complexity of the H.264 encoder and to gather accurate information about processor utilisation, Intel's VTune Performance Analyser was used to carry out code profiling. The profiler enables the collection of details such as run-time data and time spent on each function and sub-routine of the H.264 encoder.

A systematic approach is followed to quantifying the time complexity of an H.264/AVC main profile encoder. The basis of this approach is to determine the number of basic operations (cycles) required by the processor to perform each of the key encoding routines. By mapping these computational requirements to the processing capabilities of the processing unit, an estimate of the encoder's time complexity can be defined. The actual time spent on each function is calculated using the following equation:

$$T = \frac{C_{function}}{F_{processor}} \tag{5.1}$$

Where T is the actual time required to execute each function, measured in seconds. $C_{function}$ is the number of CPU cycles needed for each function, and $F_{processor}$ is processor's speed measured in MHz. In experiments to follow, each video sequence is coded at 30fps with QP initially set equal to 30. The coding parameters are varied within their full range as depicted in table 4-2. Default values for the chosen set of parameters are used as benchmarks for comparison with the resulting processor utilisation.

4.4.1 Resolution and Number of Reference Frames

Experiments have shown that each of the parameters shown in Table 4-2 have an impact on the computational complexity of the encoder. For instance, a video sequence coded with a CIF resolution requires four times the time required to code a video sequence with a QCIF resolution (quarter the resolution).

Table 4-3: The effect of using multiple reference frames on the processing time for various video categories

Number of I-Frames	Processing time (in seconds) for various video categories at different resolutions (QCIF/CIF)				
	News	**Landscape**	**Traffic**	**Sports**	**UMV**
1	50.10 /	47.25/	51.38/	58.75/	61.75/
	200.4	189.0	205.5	235.0	247.1
2	85.17/	80.33/	87.34/	99.88/	105.0/
	340.7	321.1	349.2	400.3	420.0
3	120.2/	113.4/	123.3/	141.0/	148.2/
	481.0	453.1	493.4	565.0	593.0
4	155.3/	146.5/	159.2/	182.1/	191.4/
	621.2	586.0	637.1	728.5	765.7
5	190.4/	179.6/	195.3/	223.2/	234.6/
	761.5	718.2	780.1	893.8	938.6

The use of multiple reference frames can improve compression efficiency and/or video quality [6]. However, this comes at a price; when the number of reference frames is increased, the processing time required to encode the additional reference frames would increase. This is clearly demonstrated in Table 4-3.

The "News" video sequence was coded at 30fps and the QP was set to 30. Intel's VTune Performance Analyser was used to estimate the processing time required to execute main functions. The results are shown in Table 4-4. The percentages shown in the table represent the ratio between the processing time for each function and the total time needed to encode the video sequence. Chapter 2, Section 2.5 has discussed in details the generalised structure of a hybrid video encoder, and Figure 2.5 showed the different coding tools which can be tested separately for their processor utilisation.

Table 4-4: Profiling results of "*News*" video sequence for different motion estimation algorithms

Coding Tools	Processor Utilisation		
	FS	**UMHexagonS**	**EPZC**
Intra Prediction	2.82%	3.97%	4.79%
ME/ MC	77.56%	57.24%	59.2%
Transform and Quantisation	1.91%	3.51%	3.16%
Deblocking Filter	0.49%	2.14%	1.42%
Reconstruction and Store	3.64%	10.36%	7.11%
Entropy and Other Functions	13.58%	22.78%	24.32%
Total Processor Utilisation	100%	100%	100%
Total Coding Time (Seconds)	201.23	98.91	104.55

4.4.2 Motion Estimation and Compensation

Table 4-4 depicted the profiling results of "News" video sequence and the processor utilisation for the different motion estimation algorithms. The Full Search (FS) is included to provide a reference processor utilisation for the other widely used motion estimation algorithms: The Unsymmetrical-cross Multi-Hexagon-grid Search (UMHexagonS) and the Enhanced Predictive Zonal Search (EPZS). It is obvious as table 4-4 shows, that motion estimation and compensation (ME & MC) are the most computational intensive processes, followed by entropy coding, intra prediction, reconstruction, and other remaining functions such as inter prediction, coefficient scanning, and error resilience tools.

Thus, it can be concluded that motion estimation and compensation contribute to a significant encoding time, especially when using the full search algorithm (FS). The use of a more advanced motion estimation algorithms such as UMHexagonS can correspond to reduction of total coding time to around 50% on average compared to fast full search algorithms [65]. As presented in Table 4-2, the motion estimation and compensation processes are controlled by setting the fast motion estimation parameter (useFME) to 0, 1, 2, or 3 (0: disable FME, 1: UMHexagonS, 2: Simplified UMHexagonS, and 3: EPZS).

4.4.3 Group of Pictures Structure

Consecutive frames within a coded video sequence constitute a Group of Pictures (GOP). A GOP always begins with an I-Frame followed by several B- and P-Frames. Table 4-5 shows the effect of varying different values for "I-frame period" and "number of B-frames" on the GOP structure. Parameters' values were varied over the ranges presented in Table 4-2.

Intra-picture prediction aims to improve the compression efficiency of the intra-coded pictures and intra macroblocks. Although, intra prediction contributes to around 2.5% of the total computation time (Table 4-4), it can result in considerable savings when spatial correlation is significant and the motion in the video sequence is minimal [66].

Table 4-5: The effect of varying selected coding parameters on the GOP structure

I-Frame period	Number of B-frames	GOP Structure
2	1	IBPBP...
2	2	IBBPBBP...
3	1	IBPBPBP...
3	2	IBBPBBPBBP...

4.4.4 Quantisation Parameter

Quantization is controlled by a parameter that varies from 0 to 51. QP is used to derive the equivalent quantisation step size, which directly controls the bit rate of the encoded video stream. It controls the trade-off between quality and bit rate. As previously mentioned, a QP increment by 1 results in a 12.5% reduction in bit-rate and therefore a reduction in processing time. As QP increases, quantisation step size increases, in practice, quantisation step size doubles for every increase of 6 in QP. Figure 4-1 shows a video frame from the "News" video sequence compressed at two different QP values.

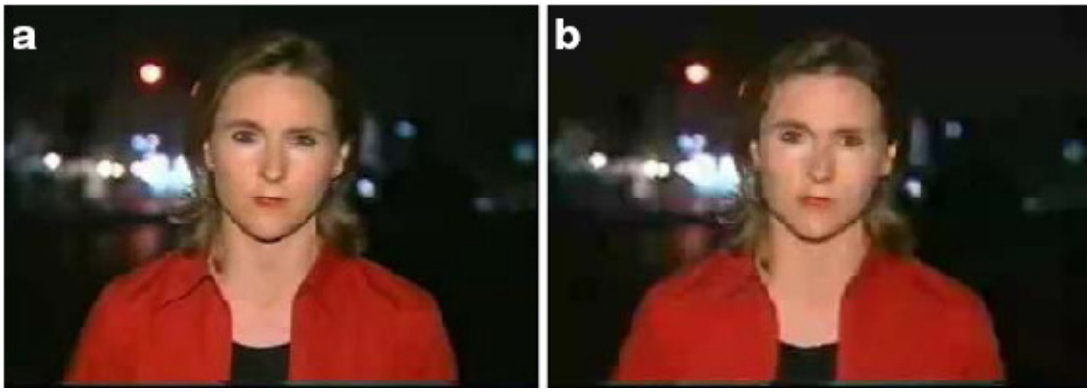


Figure 4-1: Sample image frame compressed with different compression parameters. (a) QP=30, I-Frame=2, B-Frame=2; (b) QP=40, I-Frame=2, B-Frame=2

4.4.5 Search Modes

As demonstrated in Chapter 2, Section 2.5, the tree-based decomposition adopted by H.264 to partition a macroblock into smaller sub-blocks of specified sizes serves for a better adaptation to motion estimation. With four choices of partitioning modes for macroblocks and another four choices for sub-macroblocks,

these partitions result in a large number of possible block decompositions each of which requires a separate motion vector [67]. For example, if a macroblock is coded using Inter8x8 mode, and each 8x8 sub-macroblock is coded using Inter4x4 mode, then 16 motion vectors will be coded and transmitted for this macroblock.

4.5 Rate Distortion Analysis

In video compression, rate *R* is usually expressed as the number of bits per data sample (e.g. kb/s), while distortion *D* is expressed as the variance of the difference between input and output signals. However, since most lossy compression techniques operate on video sequences that will be perceived by human observers, the distortion measure should preferably be modelled based on human perception. In which case, the R-D theory may be expressed as the following: in lossy compression the target is to lower the bit-rate by allowing some acceptable distortion of the signal. In other words, rate distortion theory either calculates the minimum transmission bit-rate R for a required picture quality, or, calculates the best stream quality possible for a given maximum bit rate. This is illustrated in Figure 4-2.

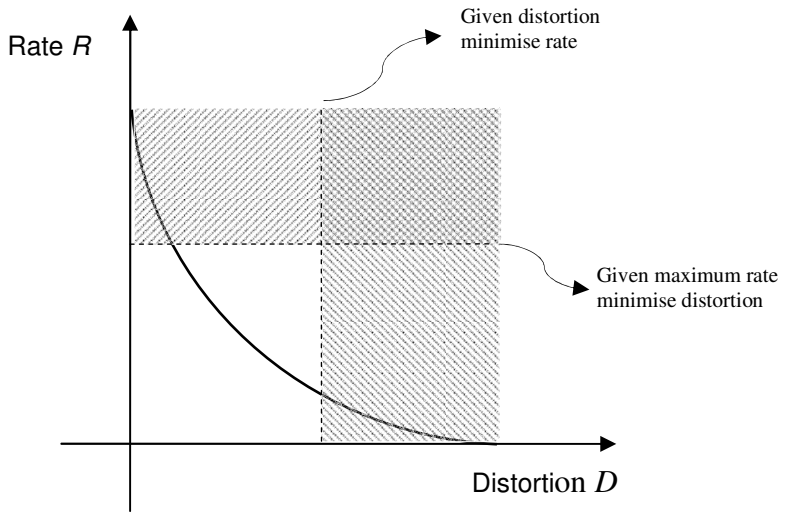


Figure 4-2: Rate-distortion theory

Figure 4-2 constitute the basis for a constrained rate distortion optimisation problem, where the cost function D is constrained by R or the cost function R is

constrained by D. Choosing encoding settings that yield the highest quality output image requires making several encoding decisions. However, this has the disadvantage that the choice might require more bits whilst giving relatively little or no quality benefit. A common example for this issue is in motion estimation [68], where encoding the motion vector to a higher precision during motion estimation might enhance quality; however, the enhancement might not be worth the extra bits necessary to achieve the respective level of quality. To overcome this conflict, a multi-objective optimisation framework will be presented in Chapter 5.

General rate-distortion optimisation techniques solve the above mentioned problem by introducing a video quality metric, which measures both the variance between the input and output signals, and the bit cost for each possible decision outcome. Such conventional approaches use unconstrained Lagrangian cost function (see Figure 4-3) to solve constrained optimization problem instead of cost function D with constrained R, or R with constrained D. The quality metric is measured by multiplying the bit cost by the Lagrangian multiplier (λ) in Figure 4-3; this represents the relationship between quality and bit cost for a particular quality level. In order to maximize the PSNR video quality metric, mean squared error is used to measure the deviation from the source.

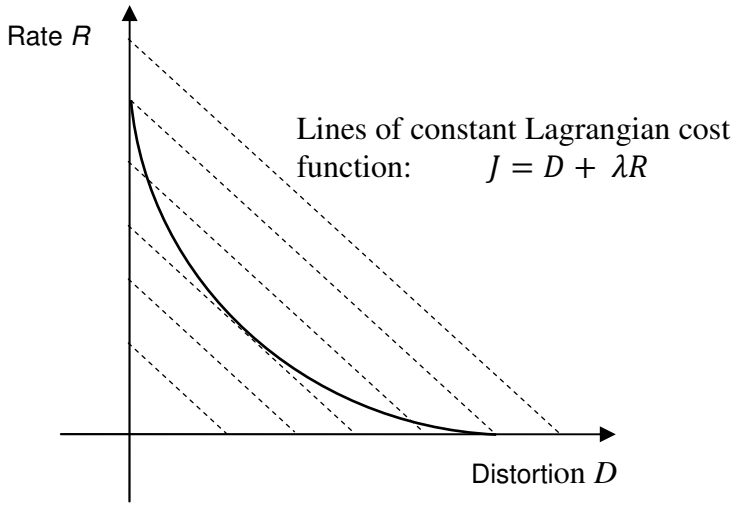


Figure 4-3: Rate-Distortion characteristics in relation to unconstrained Lagrangian cost function

In H.264/AVC, the entropy encoder makes it more challenging to calculate the bit cost. The encoder requires the optimisation algorithm to pass each block of

the video stream to the entropy encoder to measure its actual bit cost. This optimisation process starts with a transformation followed by quantisation and then entropy coding. The penalty of this process is an increase in processing time as illustrated in Table 4-6. Therefore *R-D* optimisation is conventionally used in the final steps of motion estimation in H.264/AVC.

Table 4-6: Processing time (in seconds) for various video sequences

Video Sequence	Processing Time (in seconds)	
	Without R-D	With R-D
News	200.0	412.0
Landscape	189.0	390.0
Traffic	205.0	407.0
Sports	235.0	488.0
UMV	247.0	503.0

To investigate the effect of various compression parameter choices on video quality, video sequences listed in Table 4-6 were compressed and the PSNR values were calculated and averaged for each parameter setting (see Figure 4-4). The benchmark for this set of experiments is based on QCIF video sequences coded with 1 reference frame. The ME algorithm was set to full search (FS), Quantisation Parameter was set to 30, and the intra frame period was set to 0. It is worth mentioning that typical values for PSNR in lossy video compression vary between 30 and 50 dB, where higher is better.

While the change in resolution from QCIF to CIF results in a four-fold increase in bit rate, it is clear from Figure 4-4 that it does not contribute to a significant PSNR enhancement (~1%). On a similar note, as the number of previous reference frames is varied from 1-5, PSNR values are only increased by less than 1%, and this is associated with a modest reduction of bit rate. The choice between different ME algorithms is also associated with a minimal effect on both bit rate and PSNR (see Figure 4-4).

In H.264, quantization is controlled by a parameter that varies from 0 to 51. As presented in the previous section, QP derives the quantisation step size, which directly controls the bit rate of the encoded video stream and controls the trade-off between quality and bit rate. As mentioned earlier; in theory, a QP increment by 1 contributes to 12.5% reduction in bit rate [6]. Varying the QP across a range between 30 and 40 contributes to a significant change in the quality of the compressed video. This is reflected clearly in Figure 4-4, where setting the QP value at 30, enhances the PSNR level to around 36.4dB while increasing the QP value to 40 brings the PSNR level down to around 32.3dB, i.e. ~11% decrease.

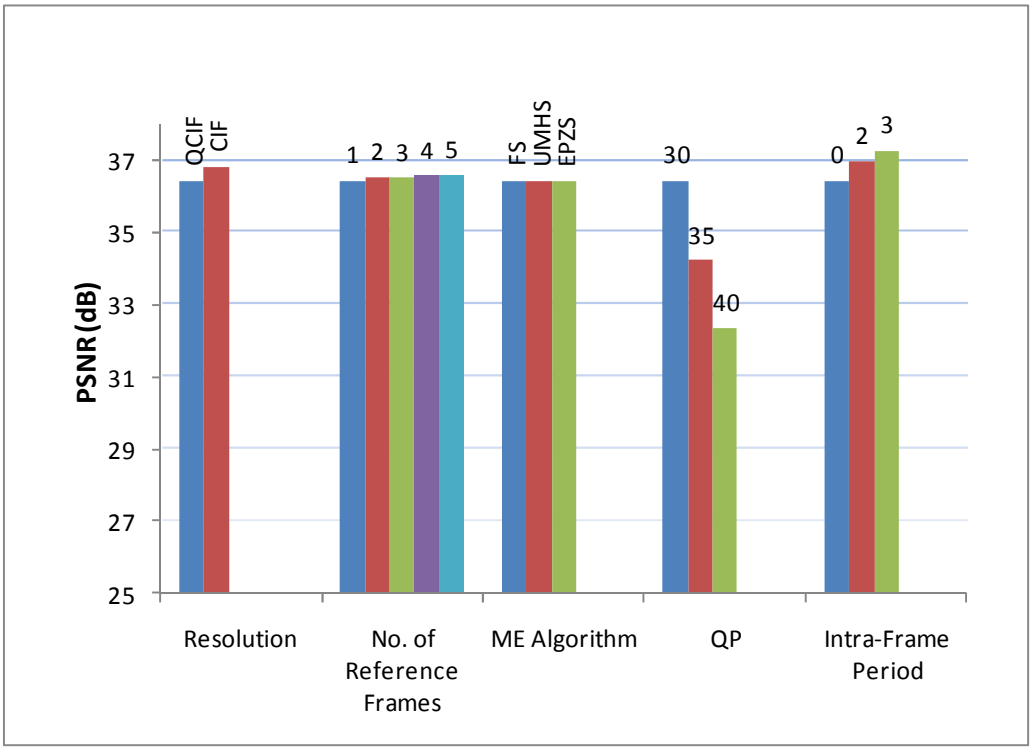


Figure 4-4: the effect of various coding setting for selected compression parameters on the PSNR

The "Intra-Frame Period" parameter determines how often a reference frame (I-frame) appears in the video sequence. In this experiment, three different values of Intra-frame period were used, namely, 0, 2, and 3, where 0 indicates that a reference frame exists only as the first frame in a GOP. It is clear from Figure 4-4 that PSNR is enhanced by around 2% when adding two more I-frames to the GOP.

4.6 Memory Utilisation

As detailed earlier in this chapter, the added features and functionalities to the modern video encoders and the enhancements on coding efficiency have all come at a price. The costs of modern video encoders include an increased demand on system resources as a result of increased computational complexity and memory requirements. The encoding parameters directly affecting the computational complexity were discussed in details in the previous sections of this chapter.

In H.264, all variables that are required throughout the encoding process are stored in what is known as the global memory of the encoder. This part of the memory can be classified into two categories: dynamic and static. The dynamic memory is allocated to encoding parameters such as resolution, quantisation, number of B-frames, etcetera. These parameters are also known as variable encoding parameters. The static memory is allocated to fixed variables such as intra-prediction probability tables. In the case of this research, a *C* implementation of the H.264/AVC codec is adopted. Consequently, three levels of memory management exist. Firstly, memory can be *statically* allocated for the lifetime of the codec's run time. Secondly, memory can be allocated *automatically* for the life time of a given function. Finally, memory can be allocated *dynamically* and can persist for the lifetime of multiple function calls [69].

Moreover, as with most other video coders, the memory system is the bottleneck of H.264/AVC encoding process. This is because it utilizes the neighbouring pixels to create a reliable predictor, leading to a dependency on a long past history of data [70], requiring architectures with large memory and high bandwidth. Additionally, video coding applications incorporate memory-intensive algorithms that require multiple large buffers. The control of these algorithms depends on a number of factors including the choice of coding parameters.

In this section, experiments were designed to test the effect of different coding parameters on the demand placed on memory resources. As each video sequence was encoded, the demands placed on the systems memory resources were recorded. A sample of the outputs of the experiments is provided in Table 4-7. As

depicted, varying different coding parameters had a significant impact on memory demands. For example, incrementing the QP value by 2 has contributed to a 20% reduction (149kB) in the size of the encoded file.

Table 4-7: The effect of varying coding parameters on the memory demands of an H.264 video encoder

QP	No. Of Reference frames	I-Frame Period	No. Of B-Frames	File Size (MB)
28	1	0	1	0.745
30	1	0	1	0.596
28	2	2	1	0.910
28	1	0	2	0.704

4.7 Conclusion

This chapter has provided a comprehensive analysis for the effect of varying a selected set of compression parameters on the efficiency of the H.264/AVC video encoder. From the analysis, the encoding parameters that have a significant impact on the computational complexity, rate-distortion characteristics, and memory utilisation have been identified; these are QP, I-Frame Period, and the Number of B-Frames. It was demonstrated that incrementing the QP by 1 contributes to a 12.5% decrease in bit rate and a 0.4dB decrease in PSNR. The other two coding parameters control the GOP structure. It was found that although intra prediction contributes to around 2.5% of the total computation time, it results in considerable savings when spatial correlation is significant and the motion in the video sequence is minimal. Moreover, the effect of adding more I and B-Frames is evident on the PSNR, where the enhancement is at least 0.25dB for each added frame.

In Chapter 5, the visual quality of video sequences, compressed using the identified compression parameters, is assessed. In Chapter 6 and 7, a framework is developed and implemented to improve the compression of video sequences based on optimising the selection of values for the identified compression parameters.

Chapter 5

Visual Quality Assessment of Image and Video Sequences

5.1 Introduction

The most consistent way of assessing the quality of an image or video sequence is through subjective evaluation; this is based on the fact that the human eye is the ultimate vision sensor. However, the complexity and expense of subjective quality assessment, and occasionally variability between human observers, have made it attractive to develop automatic quality assessment techniques using mathematical and computational algorithms that can predict perceived image and video quality automatically. However, most of the recent objective quality assessment techniques are based on computing the quality of an image or video with reference to the original (reference) image.

It is evident that human judgment on image quality across different people is not uniform. Two users' visual performance could match well in terms of their ability to pick out interesting objects, but not in terms of grading image quality. For

this reason, designing viewability measures that are quantitative yet correlate well with the visual perception of different human experts remains a challenging task.

There has been limited research in the area of evaluating image enhancement/restoration techniques, by defining viewability, even though interest in the topic is quite old ([4], [5]). Pappas and Safranek [13] state that: "Even though we use the term image quality, we are primarily interested in image fidelity, i.e., how close an image is to a given original or reference image. It is very hard to develop objective metrics that evaluate image quality without a reference image, even though the Human Visual System is very good at doing that". They examine objective criteria for image quality that are based on models of human visual system, and detail three models proposed by Lubin [14], Teo and Heeger [15], and Dally [16] and give comparative results. All of these models first perform multi-resolution frequency analysis of images, followed by contrast sensitivity, use of a masking model and finally error pooling which determines the quality of enhancement. It should be noted that Daly and Lubin's models [16] are exceptionally computationally complex and difficult to use for real applications. Hence, there is a considerable need for the development of viewability measures that correlate well with human vision, are easy to implement, and computationally cheap.

The concept of image or video viewability is not easily defined. Even though we all visually infer images as of high or low quality, it is not very easy to define what is viewable and what is not. The most primitive measure of viewability is based on image contrast. Several measures have been proposed to measure image contrast, and in particular to include the concept of target and background.

This Chapter presents a novel technique for quantitatively assessing the quality of image sequences without the need for a reference image and in a way that precisely correlates to human judgement on quality. This paves the way to Chapter 6, where a framework that incorporates multi-objective optimisation algorithms to optimise the quality metrics of compressed videos that are transmitted over low-bandwidth communication channels. The model was trained on a video dataset that involved 600 videos of 5 different categories (see Section 4.2). The validation of the

performance of this model shows that it highly correlates to the human subjective quality assessment. The work presented in this chapter is published in [71] and [72].

5.2 Theoretical Framework

A variable can take several, perhaps many, values across a range. The value is often numerical but not necessarily so. Some variables are familiar in concept but measuring them numerically seems very difficult, strange, or even impossible to achieve, as in the case of perceived visual quality. It is still not well known how such "feelings" can be assessed, which are related to personal preferences and vary significantly from one person to another. One major task in attempting to assess such psychological variables is often to move from categorical variable (e.g. like/don't like) to measured variable (e.g. degree of liking). Moreover, If we are to work with variables related to visual quality measures, then we must be able to specify them precisely, partly because we want to be accurate in the measurement of their change, and partly because we wish to communicate with others about our findings, so that it is possible for other researchers to replicate them using the same measurement procedures.

As introduced in Chapter 2, it is necessary to judge the visual quality of the video being processed in order to evaluate and compare the performance of different video display and communication systems. However, since most video services target human observers, the judgement on visual quality has to be relevant to the way the human visual system perceives the viewability of a video sequence. This in turn brings other challenges which lie in the nonlinear behaviour of the human visual system, and at the same time, the variety of factors that can affect measuring visual quality.

The task in the process of visual quality assessment is to train the developed model to measure the quality of compressed video sequences in a way that correlates very well to the human judgment on quality. Figure 5-1 shows the process of multiple regression analysis that is used to find the correlation between the human judgment on quality and the objective viewability measures. As indicated in

Figure 5-1, the process starts with compressing the video samples based on all possible combinations of an identified set of compression parameters.

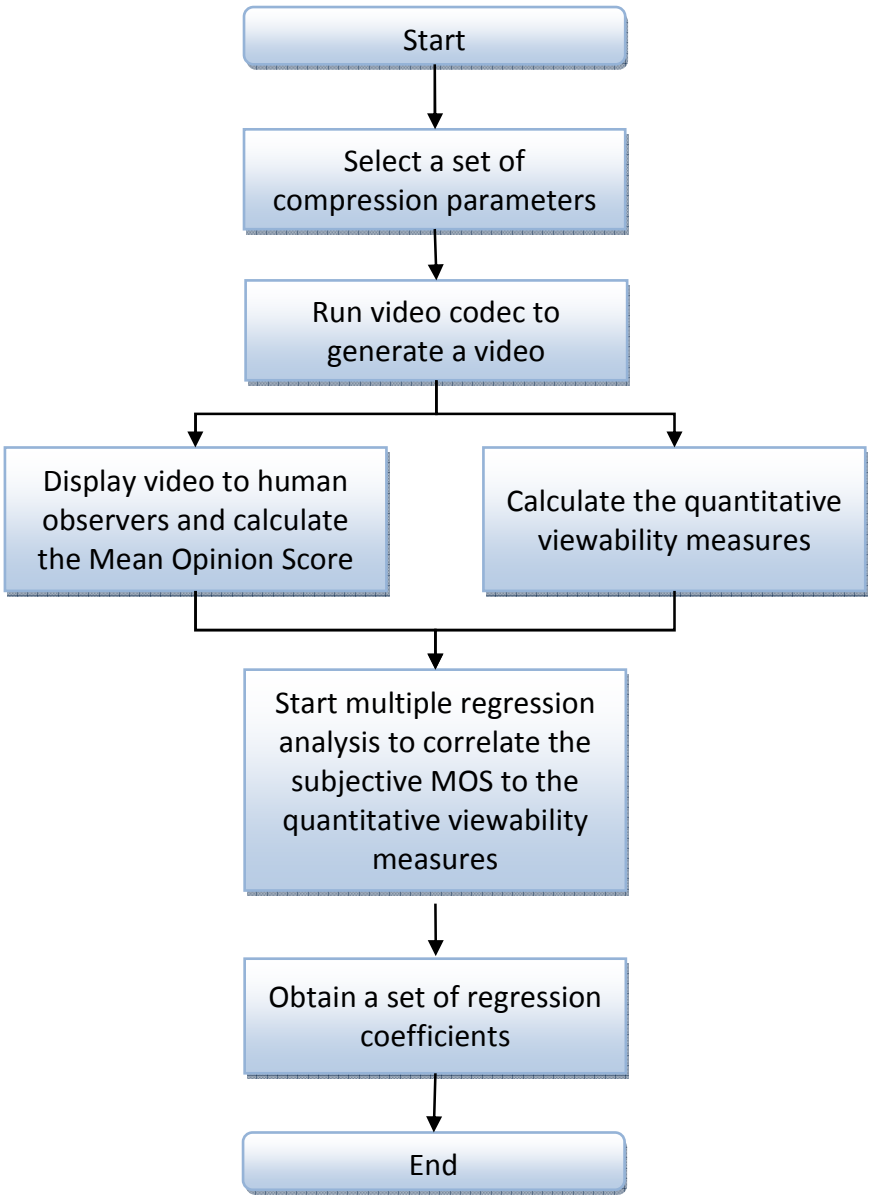


Figure 5-1: A multiple regression model that correlates the qualitative human judgment on quality to the quantitative viewability measures

Chapter 4 has investigated the effect of various compression parameters on the quality of the compressed video stream as well as their effect on the performance of the operating system such codecs operate within. The effective parameters were identified, and then varied within a fixed range with successive

levels of compression. These are: *Quantisation Parameter*, *Intra-Frame period*, and *number of B-frames*. The latter two decide the Group of Pictures structure, and the QP controls the trade-off between quality and bit rate. Table 5-1 shows a subset of video sequences that are generated based on 12 different combinations of compression parameters.

Quality of the compressed videos is then assessed based on subjective and qualitative metrics. Multiple regression analysis is then used to correlate the qualitative and quantitative measures. As will be detailed in the following sections, 33 independent variables (viewability measures) are mapped to one dependent variable (observed quality rank). The outcome of this mapping process is a vector of regression coefficients that is used in further work to predict the qualitative viewability measures from the quantitative counterparts.

Table 5-1: A subset of video sequences compressed based on 12 different combinations of compression parameters

	QP	I-Frame Period	B-Frame
video 1 (original)	28	0	1
Video1.1	30	2	1
Video1.2	30	2	2
Video1.3	30	3	1
Video1.4	30	3	2
Video1.5	35	2	1
Video1.6	35	2	2
Video1.7	35	3	1
Video1.8	35	3	2
Video1.9	40	2	1
video1.10	40	2	2
video1.11	40	3	1
video1.12	40	3	2

5.3 Visual Quality Assessment

The following sections describe a series of experiments that were conducted to analyse the relationship between qualitative and quantitative visual assessment techniques. In order to cover the qualitative side of the assessment, human participants (observers) have taken part in a number of focus groups. The aim was to calculate their mean opinion score in relation to the observed quality of video

sequences. In order to cover the quantitative side of the assessment, a number of quantitative viewability measures were identified and applied on the compressed video sequences. The correlation between the aforementioned viewability measures was analysed using multiple linear regression. After conducting these experiments, the system was capable of predicting the visual quality of image sequences based on human visual perception of quality and without the need for a reference image.

5.3.1 Subjective Quality Assessment

A focus group approach was undertaken to conduct the visual quality assessment. Krueger and Casey [73] define a focus group as "a carefully planned series of discussions designed to obtain perceptions on a defined area of interest in a permissive, nonthreatening environment". The focus group approach was adopted due to two characteristics identified by [74], which lend themselves to this research. Firstly, the aim of focus groups is to undertake an in-depth exploration of a particular subject/theme, in the case of this research, the assessment of visual quality of compressed video sequences. Secondly, a focus group allows the resources associated with undertaking a questionnaire based study to be minimised.

There were a number of practical issues which have been considered before conducting the focus groups, as identified by [74]. These include:

- ***Number of groups:*** Repeating a focus group with different people several times is desirable to minimise group bias. It has been recommended that a minimum of three focus groups should be conducted [75], however resource constraints such as time can be limiting factors.

- ***Size of groups:*** It has been recommended to conduct focus groups with size six to ten participants [76], though examples can be found of both smaller and larger group sizes.

- ***Level of moderator involvement:*** The role of the moderator is to guide the focus group participants and not to influence their behaviour.

- ***Selecting participants:*** Selection of participants is dependent upon the objectives of the study in question. Whether to use natural grouping or to use

stratifying criteria in order to select people who do not know each other is arguable.

- ***Asking questions:*** There is no standard approach on how focus group questions should be structured. According to [75], some researchers favour to use one or two general questions to encourage debate, with the moderator participating when necessary, while others prefer, as in this study, to use more structured questions.

As previously discussed, three focus groups were used to measure the average quality rating (Mean Opinion Score) assigned by participants to a number of videos, each compressed to differing degrees. It was felt that conducting three focus groups would overcome the impact of single group bias, and provide the necessary feedback. The focus groups were conduced sequentially; each session was conducted in a computer lab at Loughborough University, using conventional desktop computers. All the computers used in the study consisted of the same specification of hardware and software to avoid introducing bias. Moreover, the environment and the lighting conditions were consistent for all trials. Due to the difficulty in obtaining participants, each group consisted of 15 participants, considered to be "experts" (Research Associates and Research Students) in the fields of computer vision, digital signal processing, and image processing. The participants consisted of an equal number of males and females, aged between 22 and 35 years. No participants with visual impairment were selected for the study.

The focus groups commenced with the facilitator giving a short ten minute introduction, informing the participants of the purpose of the focus group, plan for the session, and how the information collected during the session would be used. After the initial introduction the participants were asked to view and rate the visual quality of the compressed video sequences by filling in a questionnaire (See Appendix A). The video dataset consisted of 600 video sequences distributed equally amongst five video categories (News, Traffic, Sports, Landscape, and UMVs). The duration of each video sequence was limited to 10 seconds. It has been ensured that no particular sub-category is under- or over-presented.

For each video, the participants first viewed the original (uncompressed) version; this was regarded as the benchmark video. Subsequently, they viewed the compressed versions of each original video. They were asked to rate the observed quality on a scale of 1 to 10 (10 being the best quality and 1 the worse). The compressed videos were played to observers in random order. The Joint Quality Rank (JR) for each video was calculated as the average of the ratings provided by the all participants as shown in Table 5-2.

Table 5-2: Calculating the Joint Quality Rank (JR)

Video	Obs.1	Obs.2	…	Obs.15	Joint Rank
Video1.1	9.5	9	...	7	9.17
video1.2	8	8	...	9	8.67
video1.3	5	7	...	8	8.33
video1.4	7	8	...	6	7.00
video1.5	7	7	...	8	7.33
video1.6	6	9	...	5	5.66

On completion of each video category, participants were given a twenty minute break. The task was then repeated for the remaining video categories.

After the focus group sessions, the questionnaires were collected and analysed. Table 5-3 summarises the results of a subset of the analysis. As depicted, it was found that the change of observed quality, as a result of varying the selected compression parameters, is proportional to the size of the compressed videos. It is clear from Table 5-3 that there is a noticeable trade-off between video size and quality. For example, a 60% reduction in video size could be achieved at the cost of an 8% reduction of quality, and a 78% reduction of size at the cost of a 30% reduction of quality.

Table 5-3: A subset of compressed video sequences showing the effect of the varying compression parameter on the size and the observed quality

	QP	I-Frame Period	B-Frame	size	JR
video 1 (original)	28	0	1	745 KB	10.00
Video1.1	30	2	1	298 KB	9.17
Video1.2	30	2	2	256 KB	8.67
Video1.3	30	3	1	251 KB	8.00
Video1.4	30	3	2	229 KB	7.67
Video1.5	35	2	1	164 KB	7.00
Video1.6	35	2	2	131 KB	6.83
Video1.7	35	3	1	133 KB	6.67
Video1.8	35	3	2	114 KB	6.33
Video1.9	40	2	1	95 KB	6.00
video1.10	40	2	2	74 KB	5.33
video1.11	40	3	1	75 KB	5.17
video1.12	40	3	2	62 KB	5.00

5.3.2 Objective Quality Assessment

Following the subjective quality assessment, the quality of video sequences was computed using quantitative measures, as proposed in a survey by Singh et al in [77]. These measures were chosen based on discussions with human screening experts at airports. The survey showed that the most important factors that contribute to the visual perception of the scene can be represented by 11 measures. The proposed measures are summarized in Table 5-4 and can be grouped based on the following factors: (a) edges and sharp details (V_1,V_2,V_3,V_4) (b) the amount of dark area in an image and the level of brightness (V_5) (c) well-defined uniformly textured objects, that are in contrast with their surrounding environment (V_6,V_7,V_8,V_9,V_{10},V_{11}).

Table 5-4: Summary of the proposed viewability measures [77]

Viewability measure	Viewability measure description.
Cumulative edge strength (V_1)	This is the average edge strength per pixel of the whole image, represented by $Avg_{\lvert\vec{\nabla}C\rvert}$: $$\lvert\vec{\nabla}C\rvert = \left[\frac{1}{2}\left\{g_{xx} + g_{yy} + \sqrt{(g_{xx} - g_{yy})^2 + 4g_{xy}^2}\right\}\right]^{1/2}$$ Where the gradient (g) for each colour channel can be calculated as follows: $$g_{xx} = \left(\frac{\partial R}{\partial x}\right)^2 + \left(\frac{\partial G}{\partial x}\right)^2 + \left(\frac{\partial B}{\partial x}\right)^2$$ $$g_{yy} = \left(\frac{\partial R}{\partial y}\right)^2 + \left(\frac{\partial G}{\partial y}\right)^2 + \left(\frac{\partial B}{\partial y}\right)^2$$ $$g_{xy} = \left(\frac{\partial R}{\partial x}\right)\left(\frac{\partial R}{\partial y}\right) + \left(\frac{\partial G}{\partial x}\right)\left(\frac{\partial G}{\partial y}\right) + \left(\frac{\partial B}{\partial x}\right)\left(\frac{\partial B}{\partial y}\right)$$
Amount of edge pixel (V_2)	This calculates the total number of edge pixels where edge strength magnitude is greater than the average edge strength of the image pixels, then calculates the proportion of these pixels to the whole image.
Histogram area(V_3)	The edge-strength values of a particular image are distributed into sub-groups. Then the frequency of elements of a particular sub-group is plotted against the edge-strength in the form of a histogram. This measure represents the area under the curve of the histogram.
Edge contrast (V_4)	All of the edge pixels are first determined using Sobel edge detection operator [78]. The non-edge pixels in the neighbourhood of each edge pixel are identified. Then the average of the Euclidean distance between the edge pixel and these neighbours is calculated, this represents the contrast value for the edge pixel. The 'contrast matrix' is generated for all the pixels in the image. Contrast of the image is calculated by averaging the 'contrast matrix'.

Proportion of dark pixels (V_5)	The proportion of very dark pixels in the image is calculated by counting the number of pixels that have their RGB values less than 100. $$V_5 = \frac{Number\_of\_dark\_pixels}{Total\_pixels}$$
Uniformity of texture in edge removed image (V_6)	The same algorithm used to determine Edge contrast is used here but this time on after removing the edges from the image where most of the objects from within should be smooth with a uniform texture. A highly viewable image will have sharp contrast at the edges, and uniform texture otherwise.
Difference in colour levels within a neighbourhood (V_7)	For each pixel in the image, the average Euclidean distance between the pixel and its eight neighbours is found then averaged across all pixels. This serves as a measure of contrast.
Mean pixel intensity (V_8)	This gives information on the overall brightness of the region. A bright region will have high mean pixel intensity and a dark region will have low mean pixel intensity.
Standard deviation of pixel intensity (V_9)	This describes the spread of the pixel intensity values. A high variance indicates a high contrast image whereas a low variance indicates a low-contrast image has. The standard deviation of pixel intensity also characterises the distribution's width or variability around the mean.
Skewness (V_{10}) and kurtosis of pixel intensity (V_{11})	The skewness measures the symmetry of pixel intensity distribution around its mean. Kurtosis measures the relative flatness of a distribution relative to a normal distribution.

The viewability measures listed in Table 5-4 are computed for the entire set of the training video sequences. Our dataset consisted of a total of the same 600 video sequences used in the previous subjective assessment, distributed equally

amongst the five categories. The selected video categories were considered from the Minerva Video Benchmark [62], including videos of news, landscapes, traffic, sports, and videos captured by day and night-vision cameras mounted on an unmanned vehicle. This dataset reflects the range of characteristics in the general population of compressed videos. It was ensured that no particular sub-category was under- or over-presented. The remaining sections detail the analysis that was carried out on the "News" test sequences. Similar analysis was performed on the other four video categories.

To calculate the values for viewability measures, an automated *quantitative visual assessment (QVA)* tool was developed based on the image viewability measurement technique proposed by [77]. The QVA tool was used to calculate the 11 quantitative measures defined in Table 5-4 for each frame per video sequence. These are calculated as follows: given a video consisting of frames$(F_1, \dots, F_n)$, for each frame we calculate the viewability measures $(V_1, \dots, V_m)$, where $m = 11$. Hence, for each viewability measure, the mean μ, median *k*, and standard deviation σ are calculated (see Table 5-5). The calculations on the 11 objective measures give a vector of measurements rather than a single estimate of video quality. Therefore, the revised viewability metrics form a vector of size$(1\ x\ 3m)$ as shown in the following equation:

$$(\mu_1, \dots, \mu_m, k_1, \dots, k_m, \sigma_1, \dots, \sigma_m) \tag{5.1}$$

Table 5-5: The mean, median, and std. deviation for each viewability measure

	V1-Mean	V1-Median	V1-Std.Dev.	…	V11-Mean	V11-Median	V11-Std.Dev	Joint Rank
Video1-1	0.0256	0.0190	0.0109	…	-1.7001	-1.6360	0.1650	**9.17**
video1-2	0.0254	0.0190	0.0109	…	-1.7002	-1.6360	0.1652	**8.67**
video1-3	0.0254	0.0190	0.0110	…	-1.7004	-1.6370	0.1655	**8.00**
video1-4	0.0253	0.0190	0.0110	…	-1.7005	-1.6370	0.1652	**7.67**
video1-5	0.0240	0.0180	0.0108	…	-1.7005	-1.6370	0.1652	**7.00**
video1-6	0.0242	0.0180	0.0107	…	-1.7005	-1.6370	0.1652	**6.83**
…	…	…	…	…	…	…	…	…
video12-11	0.0240	0.0180	0.0108	…	-1.7005	-1.6370	0.1652	**5.17**
Video12-12	0.0242	0.0180	0.0107	…	-1.7005	-1.6370	0.1652	**5.00**

5.4 Development of the Visual Quality Assessment Model

The main motivation of the research is to develop a framework that is capable of measuring visual quality of videos and image sequences quantitatively in a manner that correlates to the human judgement on perceived quality. Coolican (2004) [79] recommends that in order to prove the credibility of a model such as the visual quality model, it is essential to conduct reliability, validity, and standardisation checks. Reliability refers to measures that are consistent across different tests. Validity refers to experiments that measure what they are intended to measure. Finally, the standardisation check proves that the measures are applicable to a population of people and not just the sample participating in a study. The following sections address the aforementioned recommendations in the process of the development of the visual quality model.

5.4.1 Mapping

For real-life applications, it is not possible for human observers to provide video quality assessments on a large scale. Therefore, there is a need for a dynamic automated system that evaluates the visual quality of a video, translating it into a measure between 1 and 10 that matches human judgment on visual quality. This can be achieved by developing a mapping scheme that maps the vector in equation (5.1) to the Joint Rank in Table 5-5.

The data set presented in Table 5-5, represents a univariate set of data, in which there are 33 independent variables and one dependent variable (hence, called univariate). The dependent variable in this case is the Joint Rank. In order to learn more about the video data set, multiple regression analyses were conducted. For this purpose, multiple linear regression was applied where the system is trained with input data represented by the viewability measures vector, equation (5.1), and the output of the system is a predicted Joint Rank (JR) that mimics human judgment on perceived visual quality. The regression process can be described as:

$$JR = \beta_o + \beta_1 V_1 + \beta_2 V_2 + \cdots + \beta_{3m} V_{3m} + \varepsilon \qquad (5.2)$$

Where JR is the predicted variable; ($V_1 = \mu_1,\ V_2 = k_1,\ V_3 = \sigma_1, \dots, V_{31} = \mu_{11},\ V_{32} = k_{11},\ V_{33} = \sigma_{11}$), β_m is the mth coefficient of the mth predictor V_m, and ε is the residual term (the difference between predicted and observed value of *JR*). During the training phase, multiple regression analysis finds the optimal weight vector $(\beta_1, \beta_2, \dots, \beta_{3m})$ that minimises the difference between the observed and predicted output.

5.4.2 Training the Regression Model

Table 5-6 shows the outcome of a multiple regression data fitting process. The analysis was conducted using SPSS version 14, a well established statistical analysis tool. The column labelled *R* represents the correlation between the observed and predicted values of *JR*. A large value of *R* (closer to 1) represents a high correlation between the predicted and the observed values of the outcome. For example, for Model 1, which represents *News* videos, *R* is 0.95. This represents a situation in which the model predicts the observed data with very low error. R^2, also called the coefficient of determination, is a measure of how well the regression line approximates the real data points. An R^2 of 1.0 indicates that the regression line perfectly fits the data. The adjusted R^2 gives an idea of how well the model can be generalised and represents the amount of variation in the dependent variable that is accounted for by the model. In the case of *News* videos, R^2 is 0.874, which means that the independent variables (predictors) account for 87.4% of the variation in the dependent variable (the Joint Rank).

Table 5-6: Regression Model Summary

Video Sequence	Model	R	R^2	Adjusted R^2
News	1	.950	.902	.874
Traffic	2	.913	.864	.848
Sports	3	.946	.895	.899
Landscape	4	.953	.910	.904
UMV	5	.897	.853	.731

This is followed by the analysis of the variance (ANOVA), which tests whether the model is significantly better at predicting the outcome than using the mean as a best guess [80]. The results are shown in Table 5-7

Table 5-7: Analysis of Variance (ANOVA) results

	Model	Sum of Squares	Mean Square	F
News	Regression	204.324	8.514	32.5
	Residual	22.245	.262	
	Total	226.569		
Traffic	Regression	198.352	7.139	29.1
	Residual	10.158	.203	
	Total	208.51		
Sports	Regression	245.26	8.753	37.0
	Residual	20.252	.218	
	Total	265.512		
Landscape	Regression	192.984	7.689	23.4
	Residual	29.362	.196	
	Total	222.346		
UMV	Regression	235.135	8.296	19.1
	Residual	21.32	.239	
	Total	256.455		

The F-ratio represents the ratio of the improvement in the prediction as a result of fitting the model relative to the inaccuracy that still exists in the model. For example, the *News* model has an F-ratio of 32.53, which is considered highly significant ($P < 0.001$). In general, the value of the F-ratio will be higher than 1 if the improvement due to fitting the regression model is much higher than the inaccuracy within the model.

It is obvious that the variables $(V_1, V_2, \dots, V_{33})$ in equation (5.2) will not be equally important in the mapping process. Next, the relative importance of different quantitative features in predicting visual quality of videos that correlate the best with human judgment is evaluated. Table 5-8 shows the coefficients of the regression model, where the first part displays the estimates for the un-standardised values of β. These values could be substituted in regression equation (5.2). The values of β explain the relationship between each predictor and the dependent variable (*JR*). A positive value of β indicates a positive relationship between the

predictor and the outcome, while a negative coefficient represents a negative relationship.

Each coefficient in Table 5-8 has an associated standard error value which is used to determine whether or not the coefficients differ significantly from zero. Moreover, the standard error value indicates the extent to which a coefficient value would vary across different samples. The t-statistic is a measure of whether the predictor is making a significant contribution to the model. If the associated value of significance (the column labelled *sig*.) is less than 0.05 then the predictor is making a significant contribution to the model, i.e. the smaller value of sig., the larger the value of the t-statistic, and the greater the contribution of the predictor [80].

Table 5-8: Regression coefficients for the 33 dependent variables used for mapping objective to subjective quality estimates. Important coefficients are highlighted*

	Un-standardized Coefficients		Standardized Coefficients	t	Sig.
	β	Std. Error	β		
β_0 *	-67.899	31.450		-2.159	.034
β_1 *	112.504	131.581	2.167	.855	.395
β_2 *	47.959	17.850	4.890	2.687	.009
β_3 *	362.391	171.568	5.766	2.112	.038
β_4	.082	.106	.685	.771	.443
β_5 *	-17.498	68.248	-.584	-.256	.798
β_7	.085	.113	1.817	.757	.451
β_{10}	-7.717	16.692	-1.297	-.462	.645
β_{12}	-275.32	158.40	-4.690	-1.742	.085
β_{14}	-318.94	169.662	-5.001	-1.880	.064
β_{15}	-.016	.068	-.148	-.241	.810
β_{16}	-.082	.554	-.284	-.149	.882
β_{17}	22.766	66.559	.852	.342	.733
β_{18}	.244	.136	5.148	1.792	.077
β_{20}	.822	.220	7.795	3.746	.000
β_{22}	33.110	7.368	9.107	4.494	.000
β_{24}	106.089	67.256	2.333	1.577	.118
β_{25}	212.947	438.428	.586	.486	.628
β_{26}	.032	.100	.164	.323	.748
β_{27}	-.398	.756	-.840	-.526	.600
β_{28}	58.526	77.342	.712	.757	.451
β_{30}	-.333	.437	-1.165	-.762	.448
β_{31}	-1.192	1.253	-2.279	-.952	.344
β_{32}	100.078	40.671	4.585	2.461	.016

Since a hierarchical model is used, some predictors were excluded from the first stage of regression; the excluded variables are shown in Table 5-9. The table also gives estimates for β values and t-statistics for each variable. Furthermore, it also provides the partial correlation, which indicates how much contribution the excluded variable would have made if included in the model.

Table 5-9 Excluded variables

Excluded	Beta In	t	Sig.
β_{7}	-8.439	-1.888	.062
β_{8}	2.146	.646	.520
β_{9}	14.721	2.917	.005
β_{11}	-.863	-.096	.923
β_{13}	-17.220	-2.566	.012
β_{19}	2.916	.732	.466
β_{21}	-5.669	-.850	.398
β_{23}	11.446	3.088	.003
β_{29}	-12.632	-2.864	.005

5.4.3 Testing and Validating the Regression Model

This section evaluates the differences between observed and predicted values of *JR*. Table 5-10 gives information on the standardised residuals (i.e. the residuals divided by an estimate of their standard deviation) and the un-standardised residuals for 110 videos of each video category. These residuals should be as minimal as possible, and ideally as close to zero as possible.

Table 5-10: Residuals statistics

	Min	Max	Mean	Std. Dev.	N
Predicted Value	4.2153	9.3231	7.2395	1.369	110
Residual	-1.225	1.34711	.00000	.4517	110
Std. Predicted Value	-2.209	1.522	0.000	1.000	110
Std. Residual	-2.395	2.633	0.000	.883	110

Figure 5-2 indicates that the un-standardised residuals of the model are normally distributed across the data set. It shows that approximately 80% of the residuals lie between +/-0.47.

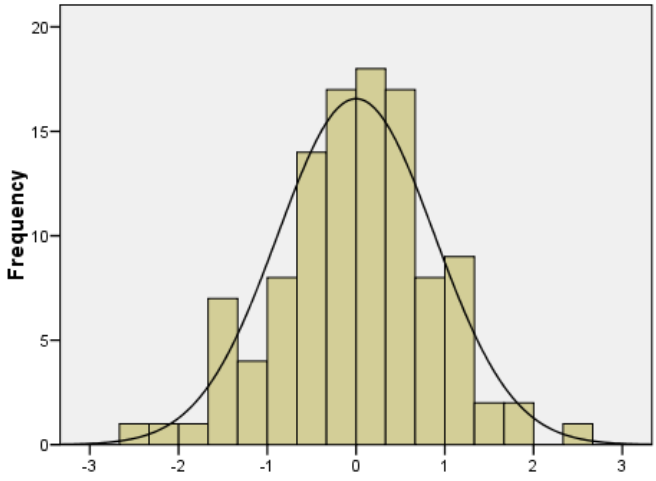


Figure 5-2: A histogram shows the distribution of the standardised residuals across 110 videos

The aforementioned regression analysis was conducted on a 12 fold cross-validation. In each fold, 110 video sequences (out of 120) were used for training and the remaining 10 were used for testing the reliability of the model. The aim of the validation process is to test the model's ability to minimise the difference between the observed and the predicted quality measures. The reference for the validation experiments are the Joint Rank values obtained as described in Section 5.3.1. Validation experiments were conducted as follows: First, 10 compressed videos were selected randomly from each category of video sequences. The QVA tool was then used to calculate the 11 quantitative viewability measures per frame per video. The mean μ, median *k*, and standard deviation σ were calculated for each viewability measure per video (similar to Table 5-5). The outcome of this process is a column vector of quantitative viewability measures per video. Second, the regression coefficients per video category obtained in Section 5.4.2 are used as scaling factors for the prediction. Equation 5.3 depicts the process of obtaining the predicted Joint Rank:

$$JR_{predicted} = [\beta_1, \dots, \beta_n].[V_1, \dots, V_n]^T \quad (5.3)$$

Where, β_n and V_n are the n^{th} un-standardised coefficient and the n^{th} corresponding viewability measure, respectively. Table 5-11 – Table 5-15 show the observed JR, Predicted JR, and the absolute difference between the observed and the predicted values for the 10 validation test video sequences per scene category.

Table 5-11: Validation of the model for *News* video test sequences

Validation Video	Observed JR	Predicted JR	|Error|
$Video_{news\_1}$	6.83	8.58	1.75
$Video_{news\_2}$	7.67	8.14	0.47
$Video_{news\_3}$	5.5	6.4	0.9
$Video_{news\_4}$	8.33	7.72	0.61
$Video_{news\_5}$	4.17	5.54	1.37
$Video_{news\_6}$	9.17	8.7	0.47
$Video_{news\_7}$	8.94	9.31	0.37
$Video_{news\_8}$	9.33	9.01	0.32
$Video_{news\_9}$	6.49	7.12	0.63
$Video_{news\_10}$	5.77	4.91	0.86

Table 5-12: Validation of the model for *Sports* video test sequences

Validation Video	Observed JR	Predicted JR	|Error|
$Video_{sports\_1}$	9.1	8.51	0.59
$Video_{sports\_2}$	6.23	6.95	0.72
$Video_{sports\_3}$	5.47	5.78	0.31
$Video_{sports\_4}$	8.46	6.74	1.72
$Video_{sports\_5}$	3.74	4.88	1.14
$Video_{sports\_6}$	5.15	4.24	0.91
$Video_{sports\_7}$	7.78	8.72	0.94
$Video_{sports\_8}$	8.94	8.59	0.35
$Video_{sports\_9}$	8.67	7.13	1.54
$Video_{sports\_10}$	5.19	4.5	0.69

Table 5-13: Validation of the model for *Traffic* video test sequences

Validation Video	Observed JR	Predicted JR	|Error|
$Video_{traffic\_1}$	8.33	9.5	1.17
$Video_{traffic\_2}$	7.73	7.33	0.4
$Video_{traffic\_3}$	8.17	8.21	0.04
$Video_{traffic\_4}$	5.33	5.78	0.45
$Video_{traffic\_5}$	5.67	4.4	1.27
$Video_{traffic\_6}$	9.1	8.09	1.01
$Video_{traffic\_7}$	6.73	6.49	0.24
$Video_{traffic\_8}$	5.5	5.76	0.26
$Video_{traffic\_9}$	8.5	7.98	0.52
$Video_{traffic\_10}$	8.3	7.77	0.53

Table 5-14: Validation of the model for *Landscape* video sequences

Validation Video	Observed JR	Predicted JR	|Error|
$Video_{landscape\_1}$	8.13	9.34	1.21
$Video_{landscape\_2}$	5.47	4.8	0.67
$Video_{landscape\_3}$	6.73	7.37	0.64
$Video_{landscape\_4}$	8.5	8.63	0.13
$Video_{landscape\_5}$	4.83	5.86	1.03
$Video_{landscape\_6}$	6.83	7.99	1.16
$Video_{landscape\_7}$	9.13	8.03	1.1
$Video_{landscape\_8}$	7.17	7.83	0.66
$Video_{landscape\_9}$	5.67	4.18	1.49
$Video_{landscape\_10}$	4.17	4.9	0.73

Table 5-15: Validation of the model for *UMV* video sequences

Validation Video	Observed JR	Predicted JR	|Error|
$Video_{umv\_1}$	4.63	6.42	1.79
$Video_{umv\_2}$	5.17	7.5	2.33
$Video_{umv\_3}$	7.67	7.91	0.24
$Video_{umv\_4}$	7.15	8.93	1.78
$Video_{umv\_5}$	5.33	4.19	1.14
$Video_{umv\_6}$	4.79	5.71	0.92
$Video_{umv\_7}$	5.5	6.48	0.98
$Video_{umv\_8}$	7.93	8.62	0.69
$Video_{umv\_9}$	8.17	7.08	1.09
$Video_{umv\_10}$	4.83	4.11	0.72

The validation data shows that the proposed model has predicted the joint Rank for the compressed video test sequences to a close degree. Table 5-16 shows a summary for the analysis of the validation experiments. It is noted that the model has successfully predicted the visual quality for the test videos. Moreover, the average difference between the predicted and the observed Joint Rank values for most of the scene categories was found to be less than one.

Table 5-16: Summary of the analysis of the validation experiments

	Min Difference	Max Difference	Average
News	0.32	1.75	0.775
Sports	0.31	1.72	0.891
Traffic	0.04	1.27	0.589
Landscape	0.13	1.49	0.882
UMV	0.24	2.33	1.168

Figure 5-3 depicts the average observed and predicted Joint Ranks for the video data set. It proves that the model has successfully predicted the quality of the video sequences to a high level of accuracy.

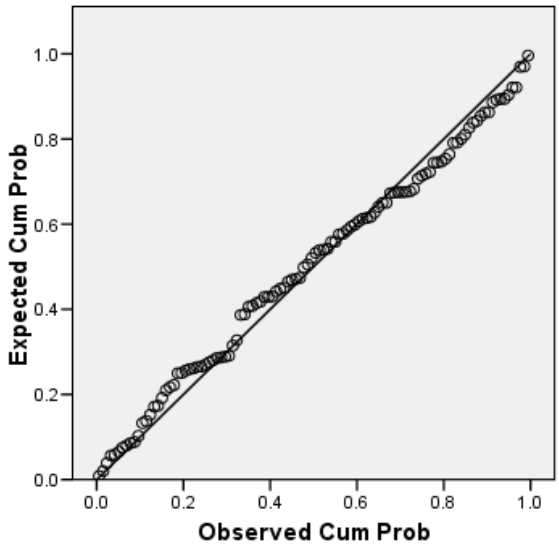


Figure 5-3: Normal P-P plot of Regression Standardised Residual

5.5 Conclusion

This chapter has presented a novel technique for quantitatively assessing the quality of image sequences without the need for a reference image. This technique was designed to precisely mimic human visual perception of quality. Within the process of visual quality assessment, the task was to train the developed model to measure the quality of compressed video sequences in a way that correlates very well to the human judgment on quality.

A model was developed to find the correlation between the human judgment on quality and a set of objective viewability measures. The model was trained on a video dataset that involved 600 compressed videos of 5 different categories. Compression parameters were varied within a fixed range with successive levels of compression, as identified in Chapter 4. The visual quality of the compressed videos was then assessed based on qualitative metrics during a number of focus groups. Multiple regression analysis was used to correlate the qualitative and quantitative

measures. The outcome of this correlation process was a vector of regression coefficients that was used to predict the qualitative viewability measures from the quantitative counterparts. This chapter was concluded with an evaluation of the differences between the observed and predicted values of visual quality.

The evaluation has shown that the proposed model has predicted the visual quality for the compressed video test sequences to a close degree, with a small average variance between the predicted and observed Joint Rank values of less than one. This high correlation suggests that there is significant potential for accurately mimicking human visual quality perception using an automated tool. The model developed in this chapter will be used in Chapter 6, where a multi-objective optimisation framework is proposed in order to optimise the quality metrics of compressed videos.

Chapter 6

Multi-objective Optimisation Framework for Video Compression

6.1 Introduction

The literature review (see Chapter 2) has highlighted the important role that international video coding standards have played in spreading digital video technology. These standards allow enough flexibility in optimising the video technology to fit a given application and make the cost-performance trade-offs best suited to particular requirements (see Figure 5-1).

In video transmission over low-bandwidth channels, high-quality video and sufficient channel throughput should be guaranteed. However, as a result of the unprecedented growth of wireless communication technologies, competition for bandwidth resources has become fierce. This highlights a critical need for effective data compression techniques [81]. However, as discussed in Chapter 3, there is a dual optimisation problem, wherein, the objective is to reduce the buffer and memory requirements while maintaining the quality of the transmitted video. Moreover, solving optimisation problems with multiple conflicting objectives is a

difficult process that might be computationally expensive. However, a perfect multi-objective optimisation solution that satisfies all objective functions and complies with all constraints associated with the decision variables may not exist [3].

Hence, the objective of this chapter is to present a novel framework for improving the compression of images and video sequences acquired from image sensors, without compromising visual quality. This framework incorporates the coding parameters that have a significant impact on memory, computational complexity and rate-distortion characteristics, as identified in Chapter 4. The work presented in this chapter is published in [82].

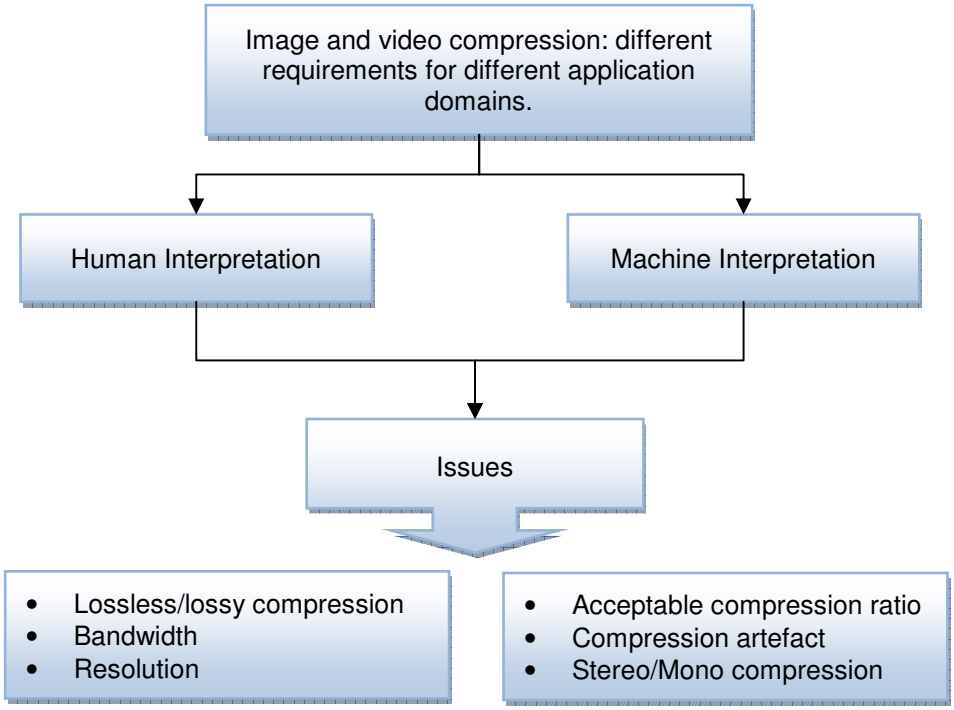


Figure 6-1: Different video compression requirements relative to the application in hand

6.2 Theoretical Framework

As mentioned earlier, this chapter presents a framework (see Figure 5-2) that aims to obtain a set of compression parameters that yields the highest image quality, whilst satisfying the bandwidth requirements of the respective application. In order to address these conflicting objectives, a novel multi-objective optimisation

framework is proposed. The framework mimics the natural evolution process to drive the search within a given population towards the optimal set of solutions. Consequently, the outcome of this framework is a set of all feasible solutions that represent the best trade-offs between the conflicting objectives

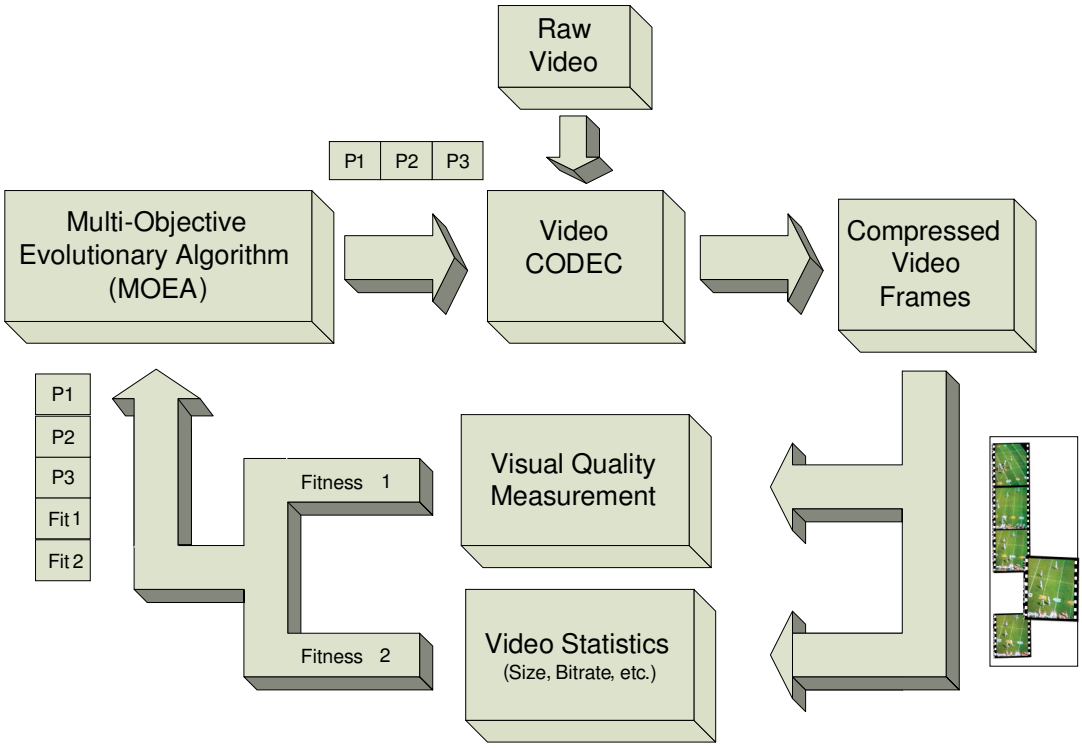


Figure 6-2: Multi-objective optimisation framework for video compression

The following sections review the breakdown of the individual components of the framework depicted in Figure 5-2.

6.2.1 Compression Algorithm

As discussed in Chapter 2, video compression techniques are based on removing redundancy in the spatial, temporal, and frequency domains, resulting in a reduction in the perceived quality [6]. The performance of a video CODEC is controlled by a set of parameters which can be varied within a predefined range. Chapter 4 has investigated the compression parameters that have a significant impact on the encoder's performance in terms of computational complexity and memory utilisation. The three most-effective compression parameters identified from Chapter 4 are: *Quantisation Parameter*, *Intra-Frame Period*, and *Number of B-Frames*. The latter two decide the Group of Pictures structure.

Figure 6-3 outlines the data flow within the proposed compression algorithm. Once the framework (Figure 6-2) is initialised, the video CODEC encodes the raw video based on a default set of compression parameters. The reconstructed video is then split into individual frames. In the following iterations of the framework, compression parameters are fed back into the CODEC from the MOEA.

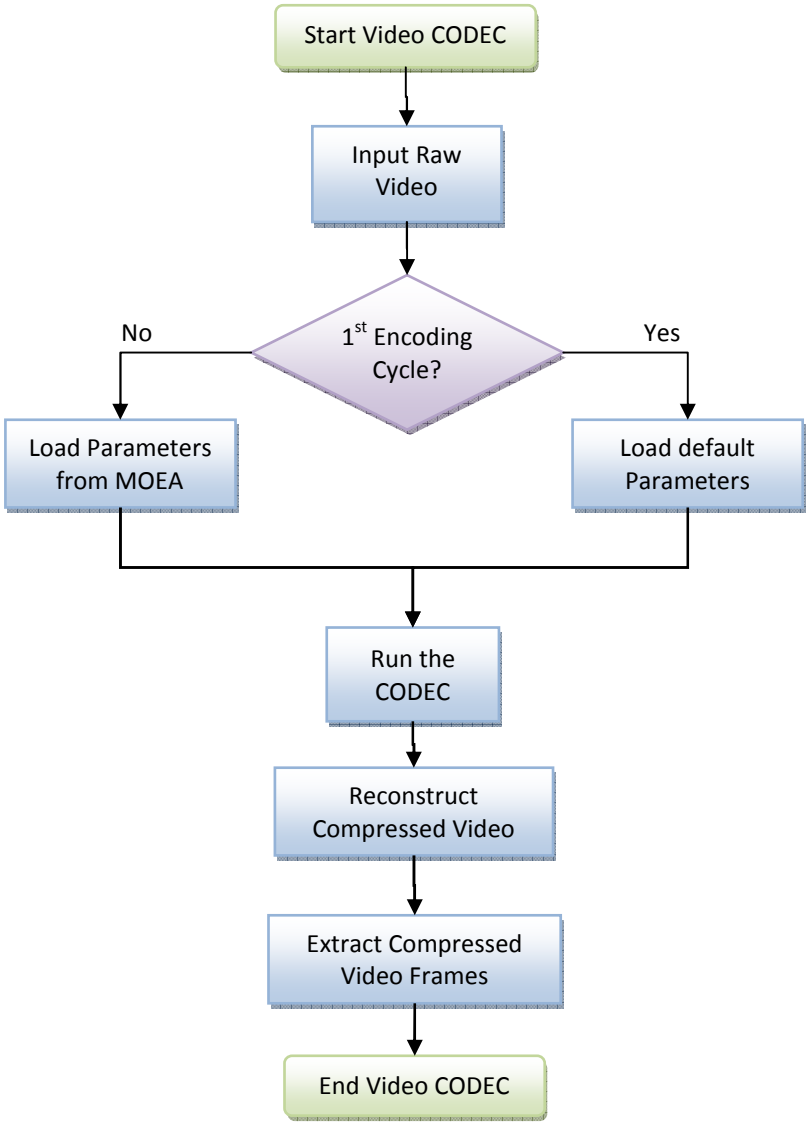


Figure 6-3: Data flow within a video CODEC

The video CODEC adopted for the proposed framework is the H.264/AVC JM Reference Software [63]. For a thorough review of the H.264/AVC CODEC please refer to Chapter 4, Section 4.3.

6.2.2 Visual Quality and other Fitness Measures

The concept of image or video viewability is not easily defined. Even though we all visually infer images as of high or low quality, it is not very easy to define what is viewable and what is not in a reliable manner. For that purpose, an automated system that measures a quantitative value for the quality of the compressed video frames is developed. This quality measure (Fitness1) is fed back to the MOEA –along with other statistical measures (Fitness2). Both measures indicate the fitness of the selected chromosome and are used to sort population into fronts based on non-domination. The MOEA, in turn, generates a new set of parameters. This process is repeated until the entire population is ranked. The following sections provide further detail on this process.

6.2.3 Multi-Objective Evolutionary Algorithm

As discussed in Chapter 3, a number of multi-objective evolutionary algorithms have been suggested. For the proposed framework, the NSGA-II algorithm is adopted. NSGA-II consists of four modules [61], including a fast non-dominated sorting module, density estimation module, crowded comparison operator, and a main loop module.

Figure 6-4 illustrates the data flow within the proposed multi-objective optimisation framework. This iterative approach starts with the initialisation of a population of chromosomes, where each chromosome represents a unique combination of compression parameters. Quantitative ranges of compression parameters are predefined and individual sets of compression parameters are coded as chromosomes. Therefore, our search space is the population of chromosomes representing all possible solutions. The output of this iterative loop is an optimised set of solutions that covers the trade off space between the objective; e.g. maximising the quality and reducing the size of an encoded video.

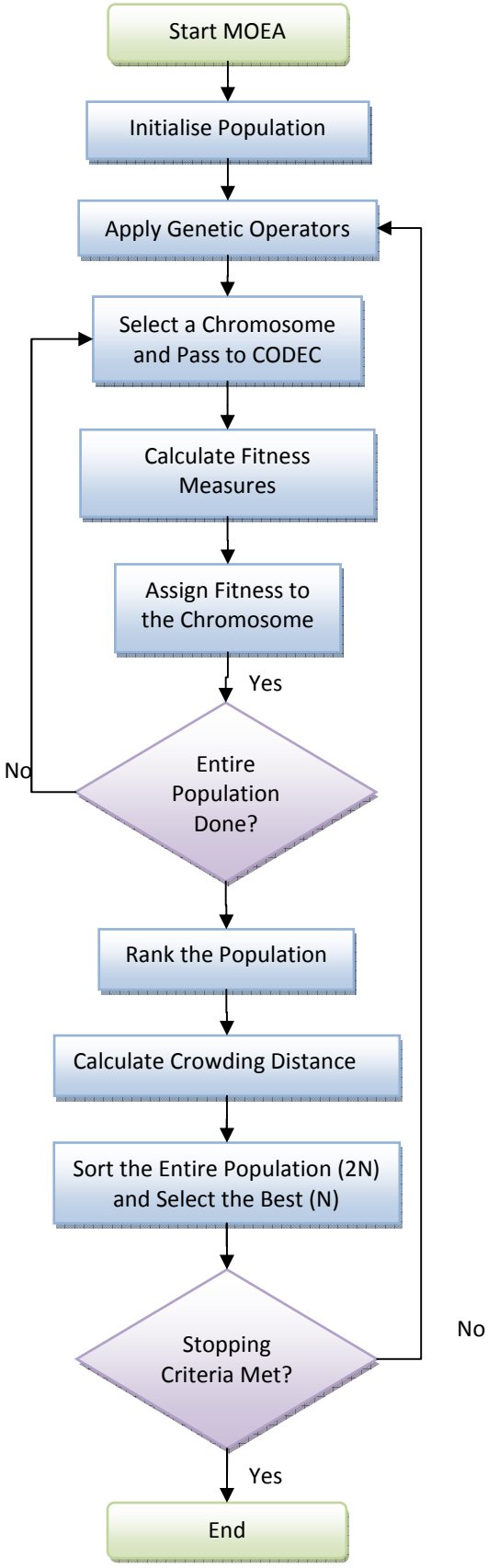


Figure 6-4: Data flow within the MOEA

The population is initialised by randomly generating N chromosomes, each representing a set of compression parameters. This population is sorted into fronts based on non-domination. An equal rank (fitness) will be assigned for individuals residing in the same front. For example members of the first front are given a fitness of 1 and those of the second front will be given a fitness value of 2, and so on, until all the population is ranked.

In the following iterations, binary tournament selection will be applied to select parents for mating. This selection is based on the rank (fitness) and crowding distance, which guarantees the diversity of selected chromosomes. Genetic operators (i.e. Crossover and mutation) are then applied to generate offsprings for the second iteration. At the end of this process, solutions converge towards the Pareto optimal front, which consists of a set of diverse optimal solutions, covering a wide range of choices for the decision maker.

6.3 Problem Formulation

As discussed earlier in the literature review, the driver for this research is the need to improve the compression of images and videos acquired from autonomous vehicles. Vision sensors in such autonomous vehicles are used to gather data about the context and status of their operating environment. This highlights one application scenario where the need for effective data compression is evident. Moreover, as soon as multiple autonomous vehicles are operated in a convoy or co-operating team, there is a need to share video information to get an accurate picture of situation awareness, whilst making efficient use of the limited bandwidth. One of the major constraints of using vision sensors with autonomous vehicles is that onboard power and weight constraints limit the maximum data processing (CPU), memory, and transmission rates (bitrate) that can be supported. Therefore, an important solution to this problem relies on the application of effective data compression schemes.

The problem lies in achieving highly compressed videos without compromising visual quality. To deal with such conflicting objectives and to accommodate cost-performance trade-offs, a multi-objective optimisation framework is proposed (see Figure 6.2). In mathematics, the definition of

optimisation might be found to have several interpretations, but all refer to finding the maxima and minima of a function. In other words, all these interpretations refer to finding one or more "optimum" values (solutions) for one or more objective (fitness) functions [50][60].

Chapter 4 has investigated the effect of different encoding parameters on CPU and memory utilisation, and rate-distortion characteristics. It emerged that optimising rate and distortion poses two contradicting objectives that are constrained by memory and CPU resources. Therefore, this problem can be considered as a multi-objective constrained optimisation problem, where both objectives (F_{rate} and $F_{distortion}$) are to be minimised under constrained resources (G_{memory} and G_{CPU}). The decision space consists of three dimensional decision variable vectors (X_i) coded as "chromosomes" (X_1,..., X_n) each representing a unique set of decision variables (x_i). Where, the decision variables represent the following compression parameters: *Quantisation Parameter*, *Intra-Frame Period*, and *Number of B-Frames*. The identified compression parameters' values are varied within finite ranges ($x_{il} \leq x_i \leq x_{iu}$) (see Chapter 4, Section 4.4).

The proposed multi-objective optimisation solution minimises the components of a vector *F(X)*, subject to an identified constraints. The general form of this optimisation problem can be formulated as follows:

$$\text{Minimise} \qquad F(X) = (F_{rate}(X), F_{distortion}(X))^{\mathrm{T}} \tag{6.1}$$

$$\text{Subject to} \qquad G(X) = \left(G_{memory}(X), G_{CPU}(X)\right) \leq (M, P) \tag{6.2}$$

Where, M and P represent the memory and processing constraints respectively. Therefore, the aforementioned problem consists of three decision variables ($i = 3$), two constraints ($m = 2$), and two objectives ($k=2$).

6.4 Obtaining Objective Functions

The function *F(X)* depicted in equation 6.1 represents a combination of two objectives, namely, rate and distortion. The task of this section is to incorporate these objectives into a mathematical expression that defines how well the data fits into the objective space. This mathematical relationship between the decision variables (x_i) and each of the above objectives is called an *objective (fitness) function*. The objective function and the constraints placed upon the problem (see Equation 5.2) must be deterministic and able to be expressed in linear form.

As discussed in Chapter 3, the process of integrating multiple objectives into a single function is referred to as aggregation. This process consists of adding the different objective functions together after multiplying them with their corresponding weighted coefficients (see Equation 3.4).

The following sections describe a regression-based approach that is adopted to obtain the objective functions for both rate and distortion. A number of experiments were conducted using all the possible combinations of the aforementioned compression parameters to produce a data set for the regression model. Next, the data set was regressed using SPSS to obtain the coefficients of each objective function in its polynomial form.

The video data set described in Chapter 4 consisted of 50 sample videos distributed across five categories. Videos within the same category shared the same parameter settings (see Table 6-1). Each of the video sequences was compressed using one of the 12 different parameter combinations described in Chapter 5, Section 5.2. Therefore, the final video dataset consisted of a total of 600 video sequences, with an average of 250 frames per sample.

Table 6-1: Ranges for decision variables used in the regression experiments

Video Sequence	Decision Variables		
	QP	I-Frame Period	Number of B-Frames
News	30-40	0-3	1-2
Landscape	30-40	0-3	1-2
Traffic	30-40	0-3	1-2
Sports	30-40	0-3	1-2

Average values for rate and distortion were recorded during the compression process and the corresponding values were used in the SPSS regression model In order to model the relationship between objective functions and the coding parameters. It is noted that the compression of all video sequences was carried out using the H.264/AVC JM Reference Software [63].

6.4.1 Rate Fitness Function

In order to model the relationship between the decision variables and the *bitrate* objective, a regression model is developed. In this model, all possible combinations of input variables (i.e. compression parameters) are mapped to the corresponding average bitrate values (see Equation 6.3) obtained following the compression of each video sample.

$$Bitrate_{avg} = [\beta_n] \,.\, [\mathrm{X_i^J}]^{\mathrm{T}} \tag{6.3}$$

Where, β_n is a vector representing all the "n" regression coefficients, X_i is a three dimensional decision variable vector, and J is a three dimensional vector of integers representing the power of each of the decision variables.

This set of experiments consisted of 480 video samples, distributed across four categories (News, Landscape, Traffic, and Sports). Each of the video samples on average comprised 250 frames. Table 6-2 shows the average bitrate in kbit/s for a subset of the video samples obtained during the compression experiments for different combinations of decision variables.

Table 6-2 Average bitrate (in kbit/s) for each video sample

Decision Variables			Average bitrate (in kbit/s) per video sample			
QP (x_1)	I-Frame (x_2)	B-Frame (x_3)	News	Landscape	Traffic	Sports
30	2	1	298	309	354	331
30	2	2	256	278	311	294
30	3	1	251	277	313	290
30	3	2	229	228	263	269
35	2	1	164	187	210	203
35	2	2	131	137	158	149

Multiple regression analysis is employed to find the optimal weight vector (β_1,β_2, …, β_n) that minimises the difference between the observed and predicted output. From all the terms comprising the polynomial representing the fitness function, only higher order terms are of particular significance. Table 6-3 lists the coefficients for the significant terms of the fitness polynomial for the "News" video sequences. The integers listed under columns x_1, x_2, and x_3 represent the powers of the relevant decision variable.

Table 6-3: The coefficients for the significant terms of the *rate* fitness polynomial for the "News" video sequences

QP (x_1)	I-Frame (x_2)	B-Frame (x_3)	Coefficients (β_n)
0	0	0	-57.879
30	2	1	59.023
30	2	2	47.959
30	3	1	242.319
30	3	2	43.62
32	2	1	0.086
32	2	2	22.766
32	3	1	-4.014
32	3	2	-174.382
35	2	1	-183.483
35	2	2	-.017
35	3	1	-.086
35	3	2	31.193
37	2	1	.149
37	2	2	.702
37	3	1	33.100
37	3	2	58.16
39	2	1	-113.256
39	2	2	.046
39	3	1	-.962
39	3	2	99.126
40	2	1	-.353
40	2	2	-3.192
40	3	1	0.32

Table 6-4 shows the outcome of the multiple regression data fitting process. As in the previous analysis in Chapter 5, this analysis was conducted using SPSS version 14. The column labelled *R* represents the correlation between the observed and predicted values of *bitrate*. A large value of *R* (closer to 1) represents a high correlation between the predicted and the observed values of the outcome. For example, for Model 1, which represents *News* videos, *R* is 0.947. This represents a situation in which the model predicts the observed data with very low error. R^2, also called the coefficient of determination, is a measure of how well the regression line approximates the real data points. An R^2 of 1.0 indicates that the regression line perfectly fits the data. The adjusted R^2 gives an idea of how well the model can be generalised and represents the amount of variation in the dependent variable that is accounted for by the model. In the case of *News* videos, R^2 is 0.892, which means that the independent variables (predictors) account for 89.2% of the variation in the dependent variable.

Table 6-4: The regression model summary for the five video categories

Video Category	Model	R	R^2
News	1	.947	.892
Landscape	2	.905	.870
Traffic	3	.913	.861
Sports	4	.930	.869

From Table 6-3, a general fitness function for the *Rate* can be estimated by the weighted sum of coefficient and its corresponding decision variables. The *Rate* fitness polynomial can be represented as follows:

$$F_{rate\_news}(X_{news}) = \beta_o + \sum_{i=1}^{n\_news} \beta_i x_1^a x_2^b x_3^c \tag{6.4}$$

Where, *n_news* represents the number of significant terms in the fitness function, and *a*, *b*, and *c* are the corresponding powers of the decision variables x_1, x_2, and x_3, respectively, for the *News* video sequences.

Similarly, fitness functions for the other sets of video sequences (*Landscape*, *Traffic*, *Sports*, and *UMV*) can be derived as follows:

$$F_{rate\_landscape}(X_{landscape}) = \beta_o + \sum_{i=1}^{n\_landscape} \beta_i x_1^a x_2^b x_3^c \tag{6.5}$$

$$F_{rate\_traffic}(X_{traffic}) = \beta_o + \sum_{i=1}^{n\_traffic} \beta_i x_1^a x_2^b x_3^c \tag{6.6}$$

$$F_{rate\_sports}(X_{sports}) = \beta_o + \sum_{i=1}^{n\_sports} \beta_i x_1^a x_2^b x_3^c \tag{6.7}$$

Where, *n_landscape, n_traffic,* and *n_sports,* represent the number of significant terms for the fitness functions that correspond to the respective video category, and *a*, *b*, and *c* are the corresponding powers of the decision variables x_1, x_2, and x_3, respectively, for the video sequences belonging to the respective video category.

6.4.2 Distortion Fitness function

Section 5.4 has detailed the development of a framework that is capable of measuring visual quality of videos and image sequences quantitatively in a manner that correlates to the human judgement on perceived quality. It detailed the design and implementation of a dynamic automated system that evaluates the visual quality of a video, translating it into a measure between 1 and 10 that matches human judgment on visual quality. This was achieved by developing a mapping scheme that maps the vector in equation (5.1) to the Joint Rank in Table 5-5.

The data set presented in Table 5-5, represents a univariate set of data, in which 33 independent variables were mapped to one dependent variable. The dependent variable in this case was the Joint Rank (JR), for information on the derivation of JR, see Chapter 5, Sections 5.3 and 5.4. In this research, the distortion

is considered to be equal to the difference between the predicted quality measure (JR) and the maximum possible visual quality (i.e. 10). The objective of this research is to optimise the distortion fitness function by minimising this difference, see equation 6.8.

$$Distortion = 10 - JR \tag{6.8}$$

As previously discussed in Chapter 5, multiple regression analyses was conducted to calculate the JR. The regression process can be described as:

$$JR = \beta_o + \beta_1 V_1 + \beta_2 V_2 + \cdots + \beta_{3m} V_{3m} + \varepsilon \tag{6.9}$$

Where JR is the predicted variable; ($V_1 = \mu_1,\ V_2 = k_1,\ V_3 = \sigma_1, \ldots, V_{31} = \mu_{11},\ V_{32} = k_{11},\ V_{33} = \sigma_{11}$), β_m is the mth coefficient of the mth predictor V_m, and ε is the residual term (the difference between predicted and observed value of *JR*). During the training phase, multiple regression analysis finds the optimal weight vector $(\beta_1, \beta_2, \ldots, \beta_{3m})$ that minimises the difference between the observed and predicted output.

6.5 Obtaining Constraint Functions

It has been shown throughout this thesis that video codecs require architectures with large memory and high processing capabilities. Furthermore, video codecs are based on data-intensive algorithms that require an efficient use of onboard resources. It follows from Chapter 4 that the efficiency of these algorithms depends on the choice of different sets of compression parameters.

As discussed in Section 6.3, rate and distortion optimisation poses two contradicting objectives that are constrained by memory and processing resources. Hence, this is regarded as a multi-objective constrained optimisation problem, where both objectives (F_{rate} and $F_{distortion}$) are to be minimised under constrained resources (G_{memory} and G_{CPU}), where memory and CPU constraints are determined by the system performance requirements.

6.5.1 Computational Complexity Constraints

Computational complexity of the H.264 codec was analysed in Section 4.6. It was demonstrated that the codec's processing requirements depend on the choice of compression parameters. However, in this section, computational complexity requirements are regarded as a constraint that a solution to the multi-objective optimisation framework must satisfy. The computational complexity constraint will be referred to as the *processing* constraint.

In order to obtain the *processing* constraint function, a regression-based approach, similar to that demonstrated in Section 6.4, is adopted. A number of experiments were conducted using all the potential combinations of the compression parameters to produce a data set for the regression model. Next, the data set was regressed using SPSS to obtain the scaling coefficients of each objective function in its polynomial form.

Similarly, each of the 10 original news video sequences was coded using 12 different combinations of coding parameters. Table 6-5 shows the processing time (in seconds) for a subset of the "News" video sequences.

Table 6-5: Processing time (in seconds) for a subset of "News" video sequences coded using different combinations of decision variables

Decision Variables			Processing Time (in seconds)
QP (x_1)	I-Frame (x_2)	B-Frame (x_3)	
30	2	1	200
30	2	2	250
30	3	1	481
30	3	2	507
35	2	1	181
35	2	2	223

Multiple regression analysis was then employed to find the optimal weight vector ($\beta_1,\beta_2, \ldots, \beta_n$) that minimises the difference between the observed and predicted output. Decision variables vectors represented the independent variables, and the processing time vector was used as the single dependent variable. From all the terms comprising the polynomial representing the fitness function, only higher

order terms are of particular significance. Table 6-6 lists the coefficients for the significant terms of the fitness polynomial for the "*News*" video sequences.

Table 6-6: The coefficients for the significant terms of the *complexity* constraint polynomial for "News" video sequences

QP (x_1)	I-Frame (x_2)	B-Frame (x_3)	Coefficients (β_n)
0	0	0	-27.325
30	2	1	106.096
30	2	2	69.236
30	3	1	-109.301
30	3	2	41.55
32	2	1	31.193
32	2	2	29.714
32	3	1	-13.036
32	3	2	96.382
35	2	1	-183.483
35	2	2	.717
35	3	1	-.086
35	3	2	68.56
37	2	1	.112
37	2	2	-6.31
37	3	1	33.100
37	3	2	98.16
39	2	1	-123.256
39	2	2	5.046
39	3	1	-.62
39	3	2	91.16
40	2	1	-2.753
40	2	2	-79.89
40	3	1	33.46

Table 6-7 shows the outcome of the multiple regression data fitting process. The column labelled *R* represents the correlation between the observed and predicted values. A large value of *R* (closer to 1) represents a high correlation between the predicted and the observed values of the outcome. For example, for Model 1, which represents *News* video test sequence, *R* is 0.933. This represents a situation in which the model predicts the observed data with very low error. R^2, also called the coefficient of determination, is a measure of how well the regression line

approximates the real data points. An R^2 of 1.0 indicates that the regression line perfectly fits the data.

Table 6-7: Fitness results for the computational complexity analysis

Video Category	Model	R	R^2
News	1	.933	.781
Landscape	2	.914	.810
Traffic	3	.909	.836
Sports	4	.951	.793

Based on the preceding analysis, a general constraint function for the *processing* complexity can be estimated by the weighted sum of coefficient and its corresponding decision variables. The *processing* constraint fitness polynomial can be represented as follows:

$$G_{\text{complexity}}(\text{X}) = \beta_o + \sum_{i=1}^{n} \beta_i \text{x}_1\text{x}_2\text{x}_3 \leq P \tag{6.10}$$

Where $G_{complexity}(X)$ represents the constraint function in its standard form, x_1, x_2, and x_3 are the three decision variables, and P is the maximum processing capability that can be supported by the system.

6.5.2 Memory Constraints

Memory utilisation of the H.264 codec was analysed in Section 4.6. It was demonstrated that the codec's memory requirements depend on the choice of compression parameters. In this section, memory requirement is regarded as a constraint that a solution to the aforementioned optimisation framework must satisfy.

In order to obtain the *memory* constraint function, a regression-based approach, similar to that demonstrated in Section 6.5.1, is adopted. A number of experiments were conducted using all the potential combinations of the compression parameters to produce a data set for the regression model. Next, the data set was regressed using SPSS to obtain the scaling coefficients of each objective function in its polynomial form.

As described before, each of the 10 original *News* video sequences was coded using 12 different combinations of coding parameters. Table 6-8 shows the frame buffer size (in Kbytes) for a subset of the *News* video sequences.

Table 6-8: Frame buffer size (in Kbytes) for a subset of "News" video sequences coded using different combinations of decision variables

Decision Variables			Fame Buffer Size (Kbytes)
QP (x_1)	I-Frame (x_2)	B-Frame (x_3)	
30	2	1	97.79
30	2	2	256
30	3	1	251
30	3	2	229
35	2	1	164
35	2	2	131

Similar to the analysis conducted in Section 6.5.1, multiple regression analysis was employed to find the optimal weight vector ($\beta_1,\beta_2, \ldots, \beta_n$) that minimises the difference between the observed and predicted output. Decision variables vectors represented the independent variables, and the processing time vector was used as the single dependent variable. From all the terms comprising the polynomial representing the fitness function, only higher order terms are of particular significance. Table 6-9 lists the coefficients for the significant terms of the fitness polynomial for the "*News*" video sequences.

Table 6-9: The coefficients for the significant terms of the *memory* constraint polynomial for "News" video sequences

QP (x_1)	I-Frame (x_2)	B-Frame (x_3)	Coefficients ($\boldsymbol{\beta_n}$)
0	0	0	-67.899
30	2	1	112.504
30	2	2	47.959
30	3	1	362.391
30	3	2	.082
32	2	1	-17.498
32	2	2	.085
32	3	1	-7.717
32	3	2	-275.382
35	2	1	-318.984
35	2	2	-.016
35	3	1	-.082
35	3	2	22.766

37	2	1	.244
37	2	2	.822
37	3	1	33.110
37	3	2	106.089
39	2	1	212.947
39	2	2	.032
39	3	1	-.398
39	3	2	58.526
40	2	1	-.333
40	2	2	-1.192
40	3	1	100.078

Table 6-10 shows the outcome of the multiple regression data fitting process. The column labelled *R* represents the correlation between the observed and predicted values. A large value of *R* (closer to 1) represents a high correlation between the predicted and the observed values of the outcome. For example, for Model 1, which represents *News* video test sequence, *R* is 0.933. This represents a situation in which the model predicts the observed data with very low error. R^2, also called the coefficient of determination, is a measure of how well the regression line approximates the real data points. An R^2 of 1.0 indicates that the regression line perfectly fits the data.

Table 6-10: Fitness results for the computational complexity analysis

Video Category	Model	R	R^2
News	1	.950	.902
Landscape	2	.913	.864
Traffic	3	.946	.895
Sports	4	.953	.910

Based on the preceding analysis, a general constraint function for the *processing* complexity can be estimated by the weighted sum of coefficient and its corresponding decision variables. The *memory* constraint fitness polynomial can be represented as follows:

$$G_{\text{memory}}(\text{X}) = \beta_o + \sum_{i=1}^{n} \beta_i \text{x}_1\text{x}_2\text{x}_3 \leq M \tag{6.11}$$

Where $G_{memory}(X)$ represents the constraint function in its standard form, x_1, x_2, and x_3 are the three decision variables, and M is the maximum memory resources that can be supported by the system.

6.6 Conclusion

This chapter has presented a novel framework for improving the compression of images and video sequences acquired from image sensors without compromising visual quality. The aim of this framework is to obtain highly compressed videos while retaining their visual quality. To deal with such conflicting objectives and to accommodate for the cost-performance trade-offs, a multi-objective optimisation framework was proposed.

Two objective functions relating to rate and distortion were formulated. These objective functions are to be minimised in a memory and CPU resource constrained environment. Therefore, two functions were formulated relating to memory and CPU constraints, which are determined by the system performance requirements. The decision space for this optimisation framework consists of three-dimensional decision variable vectors, encoded as chromosomes, each representing a unique set of decision variables (i.e. compression parameters).

The next chapter illustrates details the implementation of the multi-objective optimisation framework introduced in this chapter.

Chapter 7

Implementation and Evaluation of the Optimisation Framework

7.1 Introduction

This chapter discusses the implementation and evaluation of the multi-objective optimisation framework. Firstly, the unit testing of the individual components of the framework is summarised. This is followed by the integration testing of the individual components of the frameworks. The chapter concludes with the validation of hypotheses, introduced in Chapter 1, against the framework developed and discussed in Chapter 6, followed by a conclusion.

7.2 Implementation

This section details the implementation of the conceptual optimisation framework described in the previous chapters, as depicted in Figure 7-1. All of the elements of the framework were implemented on a PC with a single core Intel P4-2800MHz processor, with 2GB of memory and 200GB of storage capacity.

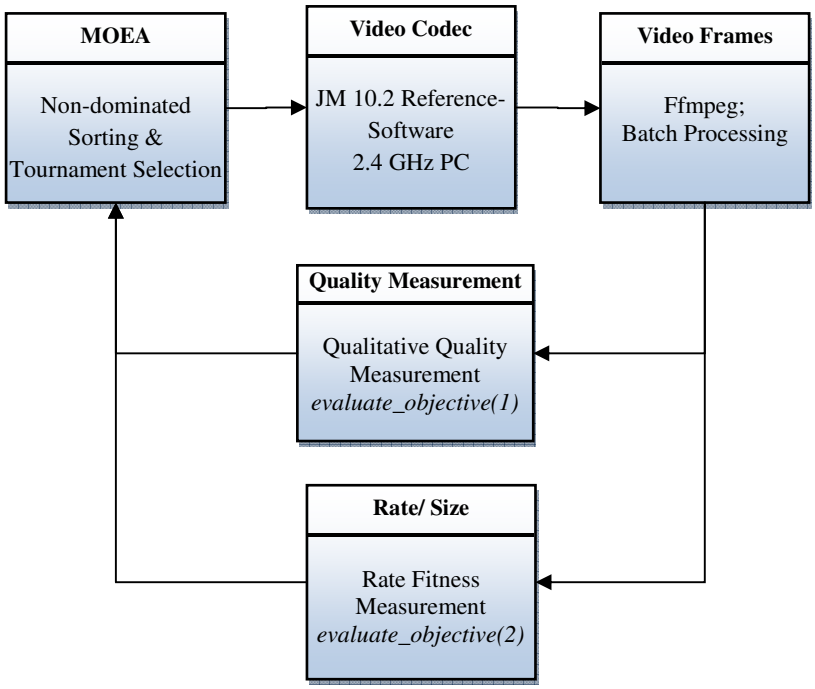


Figure 7-1: Implementation of the optimisation framework

7.2.1 Video Codec

The video codec adopted for the framework is the industry standard H.264/AVC codec - JM 10 reference software. The use of the freely available open source JM 10 software allowed for the easy configuration of the software to meet the frameworks specific requirements.

From the video codec's available resources, an executable called "*lencode*" and a configuration file called "*encoder.cfg*" were utilised. The configuration file is preconfigured with the parameters identified in Chapter 4. Figure 7-2 depicts a selection of the modified parameters from the configuration file. These parameters include non-performance parameters such as the number of frames to be coded and the frame resolution, as well as performance related parameters such as Period of I-Frames and the value quantisation parameter.

```
###############################################################################
# Files
###############################################################################
InputFile                   = "news1_12."      # Input sequence
InputHeaderLength           = 0     # If the inputfile has a header, state it's length in byte here
StartFrame                  = 0     # Start frame for encoding. (0-N)
FramesToBeEncoded           = 250   # Number of frames to be coded
FrameRate                   = 30.0  # Frame Rate per second (0.1-100.0)
SourceWidth                 = 368   # Frame width
SourceHeight                = 272   # Frame height
TraceFile                   = "trace_enc.txt"
ReconFile                   = "news1_12_rec.yuv"
OutputFile                  = "news1_12.264"

###############################################################################
# Encoder Control
###############################################################################
IntraPeriod                 = 2    # Period of I-Frames (0=only first)
QPISlice                    = 30   # Quant. param for I Slices (0-51)
QPPSlice                    = 30   # Quant. param for P Slices (0-51)
NumberReferenceFrames       = 3    # Number of previous frames used for inter motion search (1-16)

###############################################################################
# B Slices
###############################################################################
NumberBFrames               = 2    # Number of B coded frames inserted (0=not used)
QPBSlice                    = 30  # Quant. param for B slices (0-51)
```

Figure 7-2: A sample from the video codec configuration file "encoder.cfg"

The *lencode* executable uses the aforementioned configuration file to encode the video. Figure 7-3 depicts the outcome summary for the encoding process. It displays some of the important settings that were used to set up the encoder, including: image format, total encoding time, sequence type, and the motion estimation scheme. Then it provides a summary of the average data for all the frames in the coded video sequence. This summary includes: the signal-to-noise ratio (SNR) for the Y, C_b, and C_r channels, and the bit rate in kbit/s.

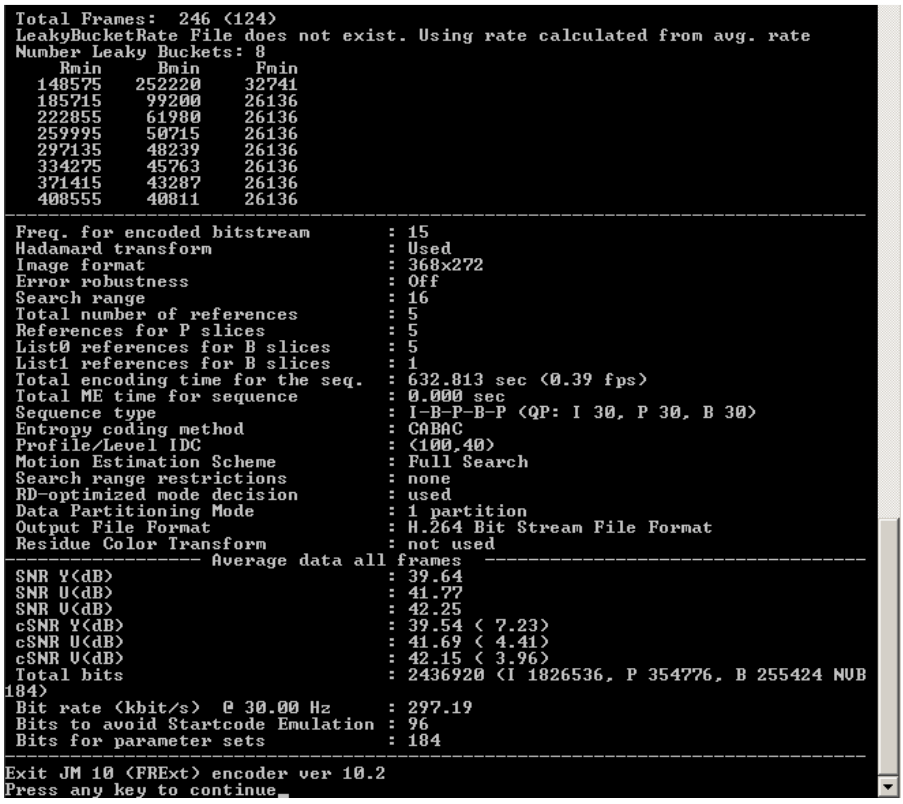


Figure 7-3: A screenshot of the encoding processing output

Chapter 5 presented the development of an automated human-based model to assess the visual quality of video sequences. In order to train the model and tune its performance to successfully assess and predict the visual quality of video sequences of different scene types, each of the 50 raw video sequences (distributed across 5 scene categories) were coded at 12 different compression parameter settings. Therefore, the preparation of the video dataset required the compression of 600 videos. To automate this process, a script was developed (see Figure 7-4), which loads all the raw videos, reconfigures the configurations file, and sequentially compresses the videos using the configuration parameters.

The encoding of the 600 video sequences took in excess of 104 hours on a 2.8 GHz Intel Pentium-4 PC, with an average processing time of 630 seconds per video sequence. The reconstructed video sequences were arranged into groups corresponding to their scene category and were made ready for the training of the regression model.

```
String VideoLocation[];
int NumberOfVideos = 0;
int CompressionParamters[][][];
int NumberOfParameterSets = 0;
int VideoCounter = 0;
int ParameterCounter = 0;


//Load the configuration information and location of raw videos

While(MoreVideos){
        VideoLocation = "File path";
}

NumberOfVideos = VideoLocation.length;

//Load the parameters' configuration information

While(MoreParameterSets){
        CompressionParamters = [Parameter 1, Parameter 2, Parameter 3];
}

NumberOfParameterSets = CompressionParamters.length;

//Reconfigure the config file and call the encoder to process all of the raw videos sequentially

While(VideoCounter < NumberOfVideos){


        While(ParameterCounter < NumberOfParameterSets){

                Open ConfigurationFile.cfg;
                Write ConfigurationFile.cfg VideoLocation[VideoCounter];
                Write ConfigurationFile.cfg CompressionParamters[ParameterCounter,0]
                [ParameterCounter,1]
                        [ParameterCounter, 2];
                Close ConfigurationFile.cfg;

                Execute lencoder.exe;

                ParameterCounter++;

        }

        ParameterCounter = 0;
        VideoCounter++;
}
```

Figure 7-4: A pseudo-code for the automation of the video compression process

7.2.2 Visual Quality Assessment Using the QVA Tool

In order to assess the visual quality for the compressed video sequences, each video had to be split into frames. Next, the QVA (see Section 5.3.2) was used to calculate the 11 viewability measures for each frame defined in Chapter 5, Table 5-4. Once all frames were processed, a MATLAB script calculated the mean, median, and standard deviation for each viewability measure, as described in Chapter 5, Section 5.3.2. Figure 7-5 presents a pseudo-code for the aforementioned process.

```
For (each video)

{
        Run ffmpeg.exe                          \\Splits the each video sequence into 250 frames

        Run QVA.bat                             \\A batch file that calculates the 11 viewability
                                                measures for each video frame and outputs a comma
                                                delimited txt file.

        Run a MATLAB script                     \\ Reads the comma delimited file and calculates the
                                                mean, median, and standard deviation for each
                                                viewability measure, and generates the Joint Rank.

        Write (Mean, Median, Standard Deviation, Joint Rank)
}
```

Figure 7-5: A pseudo-code representing data flow within the developed QVA tool

Extracting video frames was done using *ffmpeg*, which is a command line tool, used to convert multimedia files between formats. *ffmpeg* is a free software and is licensed under the GNU Lesser General Public License (LGPL) [83]. In order to automate the frame extraction process and the calculation of the objective viewability measures per video frame, *batch processing* was used. Figure 7-6 depicts an example of the batch file called "QVA.bat" used to process one of the *News* video sequences.

```
@echo off
set srcdir=.\video7
set bindir=.\QVA\Debug
set dstdir=.\QVA\results
set get_frames=.\ffmpeg
set jr_exe=.\Project ://Current Directory
set PATH=%get_frames%;%PATH%


%get_frames%\ffmpeg.exe -s 368*272 -i %get_frames%\input\video7.yuv
%get_frames%\output\video7\news7_%%03d.ppm


for %%v in (%srcdir%\news7_00*.ppm) do %bindir%\QVA.exe %srcdir%\%%~nxv -S && echo %%~nxv
for %%v in (%srcdir%\news7_05*.ppm) do %bindir%\QVA.exe %srcdir%\%%~nxv -S && echo %%~nxv
for %%v in (%srcdir%\news7_10*.ppm) do %bindir%\QVA.exe %srcdir%\%%~nxv -S && echo %%~nxv
for %%v in (%srcdir%\news7_15*.ppm) do %bindir%\QVA.exe %srcdir%\%%~nxv -S && echo %%~nxv
for %%v in (%srcdir%\news7_20*.ppm) do %bindir%\QVA.exe %srcdir%\%%~nxv -S && echo %%~nxv

%jr_exe%\joint_rank.exe

chdir %get_frames%
```

Figure 7-6: An example of a batch process used to extract video frames and calculate the objective viewability measures

The code depicted in Figure 7-6 calls *ffmpeg.exe* to extract video frames from a video sequence and stores them at the specified path. Video frames where

extracted as Portable PixMap (.ppm) files, this was found to be the most convenient method of saving image data. In the next step the script calls the quantitative viewability assessment tool (*QVA.exe*) to calculate the objective viewability measures. The output of the process depicted in Figure 7-6 is a comma delimited text file that consists of 250 rows, with each row representing the 11 objective viewability measures for each video frame. Figure 7-7 displays a snapshot of the video frames extraction process, followed by the calculation of the quantitative viewability measures.

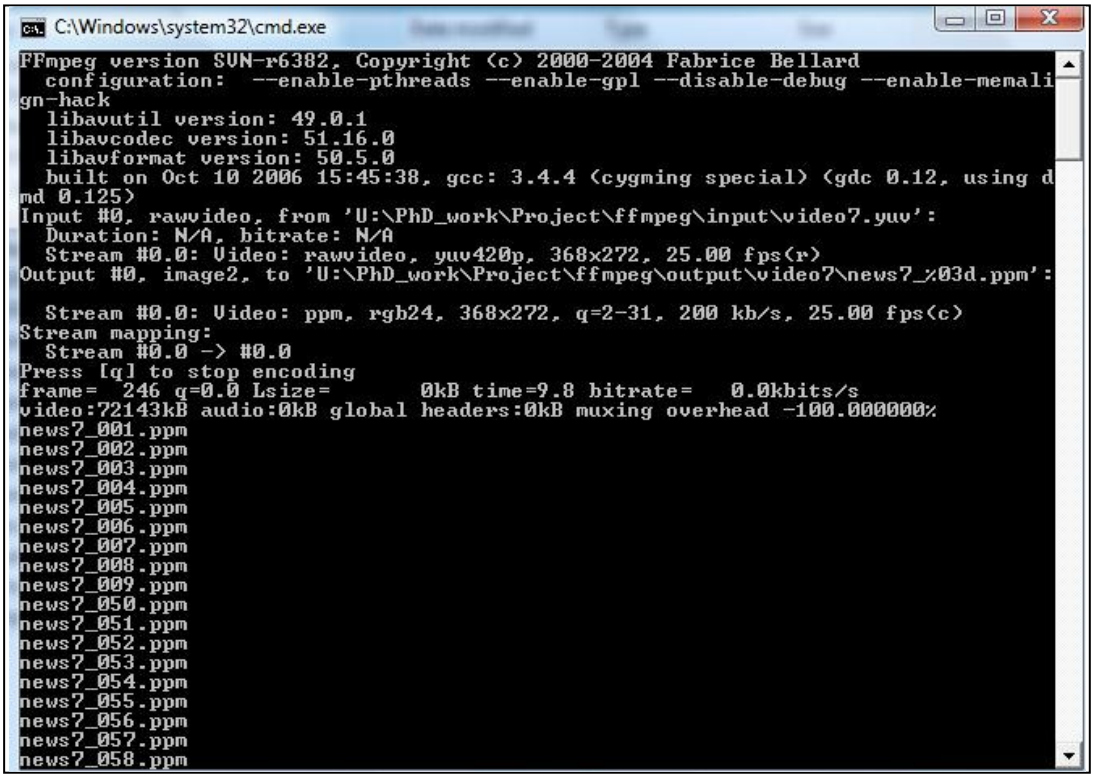


Figure 7-7: A snapshot for the process of extracting and assessing the objective quality of a sample *News* video sequence

Following the extraction of video frames and the calculation of the objective viewability measures, the mean, median, and standard deviation for each viewability measure were calculated across the entire set of frames per video sequence. From these measures, the Joint Rank was calculated. To achieve this, a MATLAB script was implemented (see Figure 7-8). The script first loads the comma delimited text file outputted from the script in Figure 7-7, then calculates the mean, median, and standard deviation, and stores them in an array "statistical_variable". The qualitatively adjusted viewability measure, Joint Rank, is calculated by multiplying

the quantitative "statistical_variable" array with the "weights" array generated from the training process of the qualitative regression model developed in Chapter 5, Section 5.4.2.

```
function joint_rank

load output.txt

statistical_variable = [mean(output(:,:)) median(output(:,:)) std(output(:,:))];

weights = [112.5;-275.38;47.959;106.09;362.39;-318.98;212.95;0.081656;-0.016475;0.03239;-0.082475;-
0.3977;-17.498;22.766;58.526;0.085309;0.24385;-0.33261;0.82243;-1.1922;-7.7167;100.08;33.11;-116.72];

Joint_Rank = β₀ + (statistical_variable * weights)
```

Figure 7-8: MATLAB script showing the calculation of the Joint Rank

7.2.3 Multi-Objective Evolutionary Algorithm (MOEA)

The MOEA consists of a MATLAB implementation of the NSGA-II algorithm, introduced in Chapter 6, Section 6.2.3. The original NSGA-II algorithm was obtained from [84]. This algorithm was modified as part of -this research. The modified elements of the code are depicted in Figure 7-9 and are detailed in this section.

The illustrated algorithm is based on evolutionary processes for finding the optimal set of solutions for the identified objective functions. The algorithm is first initialised by defining the population size, the total number of generations, and the number of decision variables. The decision variables space is limited to three decision variables as detailed in Chapter 6. A minimum and maximum value for each decision variable is defined, the population size is set to 100. During the experimental runs of the algorithm, it was found that the solution space converges to the optimal set of solutions, i.e. the Pareto front, after 20 generations. Therefore, the maximum number of generations is set to 20, and this is regarded as the stopping criteria after which the algorithm will terminate.

```
no_of_chromosomes = n                          //defines the population
no_of_generations = g                          //defines the total number of generations
no_of_decision_variables = d                   //defines the number of decision variables

initialise_variables()                         //Initialises population; generates "n" chromosomes each
                                               consisting of "d" decision variables

for i = 1 : g

        genetic_operator()                     // apply genetic operators: crossover and mutation;
                                               intermediate population size of 2n
        non_domination_sort_mod()              //calculates the Rank and the Crowding Distance for
                                               each chromosome. Then selects the best n solutions
        for  j = 1 : n
                select a chromosome            //a random chromosome is selected and passed to
                                               codec
                encode video sequence          //prepares the configuration file and runs the codec
                evaluate_objective(1)          //calculates fitness value 1 and concatenates it to the
                                               selected chromosome
                evaluate_objective(2)          // calculates fitness value 1 and concatenates it to the
                                               selected chromosome
                j = j + 1;                     //loops until the entire population is done
        end

tournament_selection()                         //selects chromosomes at random and compares
                                               their fitness then performs genetic operators
non_domination_sort_mod()                      //sorts the current intermediate population based on non
                                               domination
replace_chromosome()                           //replace the unfit individuals with the fit individuals to
                                               maintain a constant population size

i = i + 1;

end
```

Figure 7-9: Pseudo-code for the evolution process

The function *initialise_variables()* creates the chromosomes by initialising the decision variables (compression parameters) with random values based on their defined ranges. A random number is picked between the minimum and maximum possible values for each decision variable. In addition to the decision variables, the chromosome vector has the fitness value of each objective function, the rank, and the crowding distance, all concatenated at the end. However, when the function *genetic_operator()* is called, only the "decision variables" part of the chromosome vector is used to perform the genetic operations like crossover and mutation.

Once the population is initialised, function *non_domination_sort_mod()* sorts the population into fronts based on non-domination. Individuals of the first front are given a rank 1; individuals of the second front are given a rank 2, and so on, until individuals residing in all fronts are ranked. The diversity of the solutions is introduced by using the crowding distance operator [60], which is a measure of

density of solutions in a front. In order to choose between solutions residing in different fronts, i.e. different nondomination ranks, solutions with lower rank are preferred. However, if the choice is between solutions located in the same front, then solutions located in a lesser crowded region, i.e. with higher crowding distance, are preferred.

After the execution of the *genetic_operator()* function an intermediate population that consists of parents and offsprings is formed. Therefore, the population size at this stage is two times the initial population. The *non_domination_sort_mod()* function sorts the intermediate population based on non-domination. Only the best n solutions are taken forward to the next stage, where each chromosome is passed to each of the objective functions to be evaluated. The function *evaluate_objective()* takes one chromosome at a time, calculates the fitness values, and concatenate them to the selected chromosome (see Figure 7-10).

```
f = [];
% Objective function one
rate_function = 0;
for i = 1 : 3
      for j = 1: 33
            rate_function = rate_function + ([β(j)] [X(i)]^[n])
      end
end

%Rate objective function
f(1) = rate_function;

% Objective function two
dist_function = 0;
for i = 1 : 33
      dist_function = dist_function  + (abs (10 - (β(i)V(i)));
end

% Distortion objective function
f(2) = dist_function;
```

Figure 7-10: A code extract of the function "*evaluate_objective()*" to calculate the fitness of the two objective functions

The function *evaluate_objective()* takes an array of decision variables and returns the values of the objective functions. The returned values represent two different fitness measures for the selected decision variables. These values are concatenated at the end of the decision variables vector. For more details on the objective functions, refer to Chapter 6, section 6.4.1 and 6.4.2 respectively. The

overall algorithm, as depicted in Figure 7-9, minimises the two objective functions, i.e. minimises both the rate and the distortion functions.

7.3 Combined Evaluation

Following the unit testing performed in previous chapters; this section discusses the integration testing of the components of the optimisation framework, in particular, the effects of the individual components on the overall functionality of the framework. As part of this evaluation, a set of simulation based experiments were conducted. Table 7-1 summarises the default parameters adopted for all the simulations evaluated in this section. The simulation experiments were followed by the validation of the output of the optimisation framework.

Table 7-1: Parameter settings for the simulations

Simulation Parameter	Value
Population size	100
Number of decision variables	3
Number of generations	20
Number of objective functions	2
Number of constraints	2
Crossover probability	0.90
Mutation probability	0.01

7.3.1 Testing of the Optimisation Framework

This section describes a set of simulation experiments that were designed to test the correct functionality of the optimisation framework. As seen in Table 7-1, the optimisation algorithm was set to generate a population of 100 chromosomes, each consisting of three decision variables (see Table 7-2). The two objective functions and their associated constraints were discussed in detail in Chapter 6. Crossover and mutation probabilities were set to 0.90 and 0.01 respectively. In total, 480 video sequences belonging to four video categories were used in the simulation tests. Table 7-2 lists the ranges for the decision variables that were identified in Chapter 4 and used throughout the thesis.

Table 7-2: Value ranges for the decision variables

Decision Variables	Value range
Quantisation parameter	30-40
I-Frame Period	2-3
Number of B-frames	1-2

NSGA-II was used for the aforementioned simulations because of its ability to find an optimum set of solutions that is close to the Pareto-optimal set. It was observed that solutions converge more towards the Pareto-optimal front as the number of generations is increased. During the simulations, the maximum number of generations was varied between 5 and 20. It was noticed that the convergence towards the Pareto-optimal front did not significantly improve after 20 generations. Figure 7-11 depicts the evolution process starting from an initial population of solutions, the convergence after 6 generations, and the convergence after 20 generations.

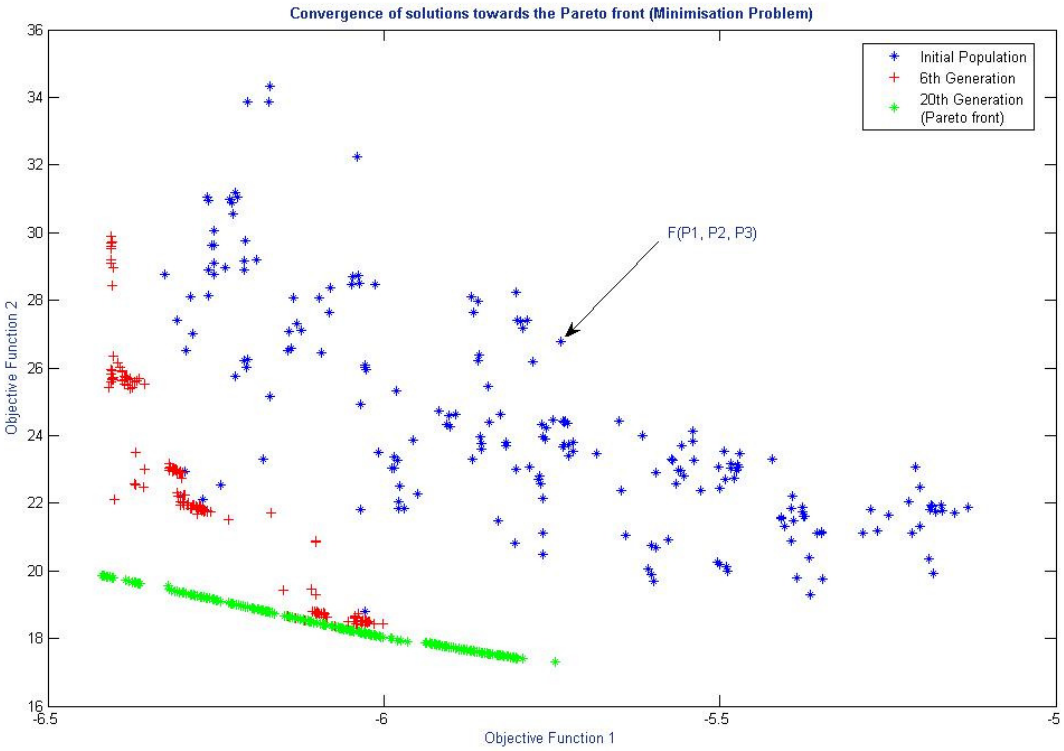
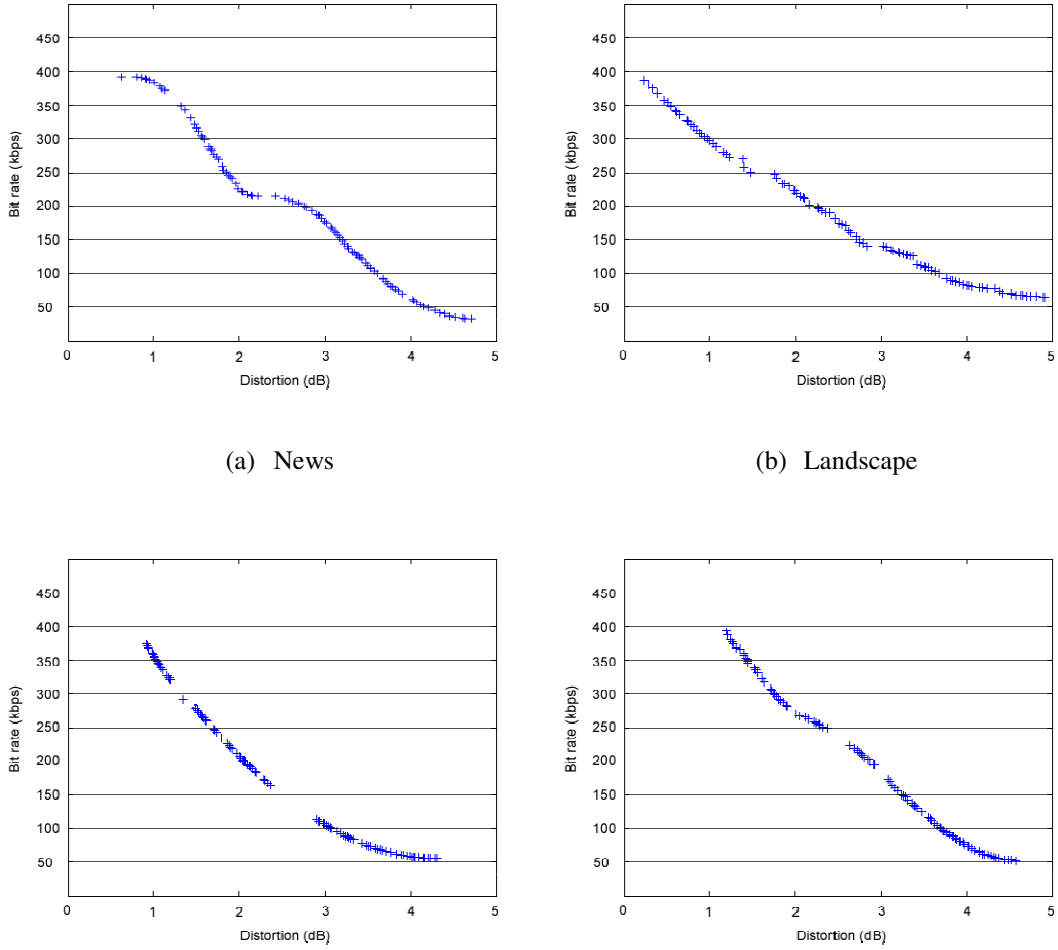


Figure 7-11: Convergence of solutions towards the Pareto front – a minimisation problem

The goal of these simulations was to obtain a diverse set of Pareto-optimal solutions for each of the four video categories. For example, in the case of the *News* category, the optimisation framework was set according to the aforementioned

default parameters. The optimisation algorithm was then executed on a population of a 100 chromosomes and run until the completion of 20 generations. The output from the simulation of the four video categories is depicted in Figure 7-12 (a-d).



(a) News (b) Landscape

(c) Traffic (d) Sports

Figure 7-12: Pareto-optimal solution obtained from simulation experiments

In Figure 7-12, each coordinate corresponds to a unique combination of three coding parameters used to compress the source video to achieve the respective bit rate and distortion represented by the y- and x- axis respectively. These corresponding bit rates, distortion levels, and coding parameters are stored in look tables for each of the four video categories. Table 7-3 provides a summary of the *News* video sequences in a form of a lookup table. In the table, a tick represents a

number of unique coding parameters that are capable of meeting the desired system requirements.

Table 7-3: A sample lookup table for the *News* video category

Bit rate levels	Distortion levels				
	0-1	**1-2**	**2-3**	**3-4**	**4-5**
0-50	X	X	X	X	✓
50-100	X	X	X	✓	✓
100-150	X	X	X	✓	X
150-200	X	X	✓	✓	X
200-250	X	✓	✓	X	X
250-300	X	✓	X	X	X
300-350	X	✓	X	X	X
350-400	✓	✓	X	X	X

The lookup table depicted in Table 7-3 allows the user to select the optimum coding parameters based on three possible scenarios: (i) the system is limited in bandwidth resources, (ii) supports a maximum tolerable distortion, and (iii) is both limited in bandwidth resources and supports a maximum tolerable distortion. For example, Figure 7-13 illustrates a set of possible optimal solutions for a video encoder constrained by a bandwidth of 300 kbps and a maximum tolerable distortion of 3dB.

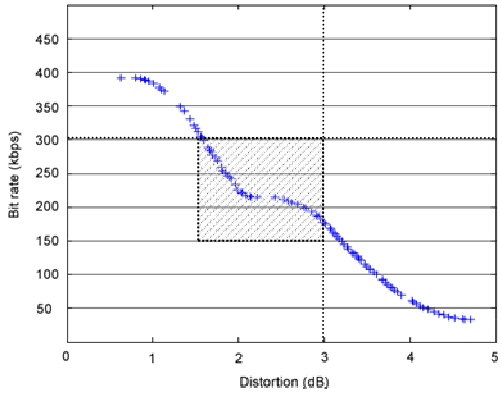


Figure 7-13: An example set of possible optimal solutions for a video encoder constrained by a bandwidth of 300 kbps and a maximum tolerable distortion of 3dB

7.3.2 Validation of the Optimisation Framework

This section describes the validation of the performance of the optimisation framework. To validate the output of the aforementioned simulation experiments, a random subset consisting of 10 samples of the optimal solutions for the *Traffic* video category was experimentally tested. The validation test bed consisted of a PC (Intel P4-2800MHz processor, and 2GB of memory). The source video was first encoded using the selected sample sets of compression parameters, and then the QVA tool, developed in Chapter 5, was used to assess their visual quality. Table 7-4 represents the findings of this validation process.

Table 7-4: The findings from the optimisation framework validation process

Sample No.	Constraints		Optimised coding parameters			Results	
	Max bit rate (kbps)	Max distortion (dB)	QP	I-Frame	B-Frame	Bit rate	Joint Rank
1	300	3	33	2	1	279	7.2
2	300	3	35	3	1	248	7.8
3	300	3	31	2	2	291	8.5
4	300	3	35	3	2	261	7.9
5	300	3	34	3	1	256	7.7
6	300	3	34	2	2	245	7.5
7	300	3	36	2	2	241	7.1
8	300	3	35	2	1	234	7.2
9	300	3	37	3	1	235	7.0
10	300	3	39	3	2	242	7.3

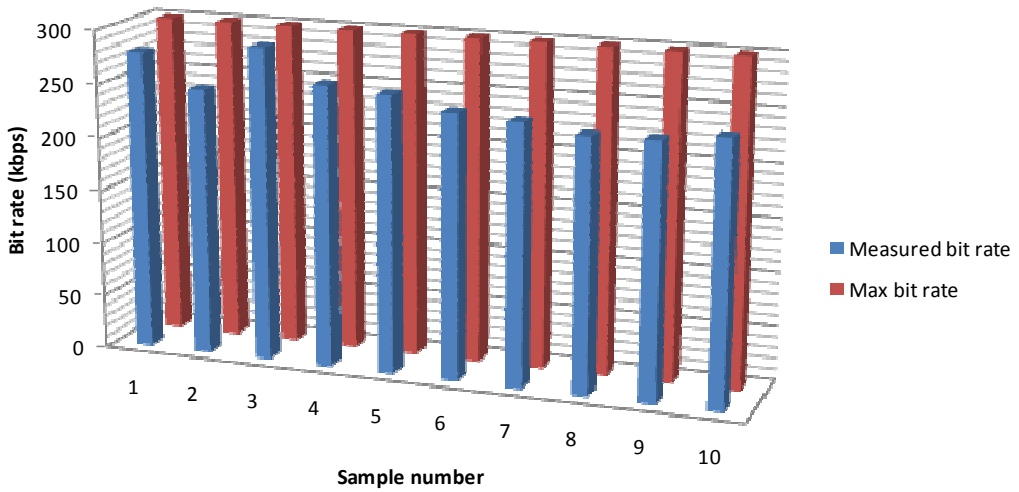


Figure 7-14: Measured values of bit rate compared to the maximum allowed values

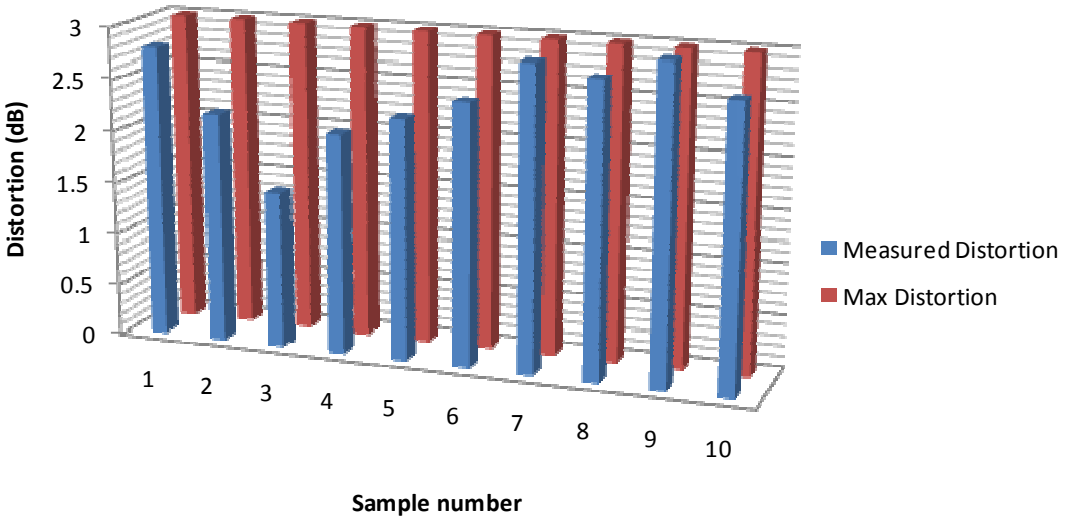


Figure 7-15: Measured values of distortion compared to the maximum allowed values

Figures 7-14 and 7-15 illustrate that the optimised coding parameters were successfully used to generate video that met the bandwidth constraints and maximum distortion requirements. The high accuracy of the validation trials proves the ability of the developed optimisation framework to enhance the performance of the H.264/AVC video encoder through the provision of optimised sets of compression parameters that fit specific application and resource constraints.

7.4 Conclusion

This chapter has discussed the implementation of the multi-objective optimisation framework. Firstly, the unit testing of the individual components of the framework was summarised. This was followed by the integration testing of the individual components of the framework. The chapter concluded with the evaluation and validation of the outcomes of the optimisation framework.

The evaluation process consisted of an extensive set of simulations and validation experiments for the optimisation framework. This process was both time consuming and computationally expensive. However, the output of this optimisation process, i.e. the lookup tables, provides a source of optimised coding parameters, which can be directly referred to for real-time video coding applications.

Finally, this chapter has illustrated the high accuracy of the outcome of the optimisation framework, and the potential of using such tools for optimising the H.264/AVC video encoder. Moreover, this process can be re-applied to optimise the performance of other video coding standards.

Chapter 8

Conclusions and Recommendations for Future Work

8.1 Summary

In video transmission over low-bandwidth channels, high-quality video and sufficient channel throughput should be guaranteed. However, as a result of the unprecedented growth of wireless communication technologies, competition for bandwidth resources has become fierce. This highlights a critical need for optimising the performance of video encoders. However, there is a dual optimisation problem, wherein, the objective is to reduce the buffer and memory requirements while maintaining the quality of the encoded video. Additionally, through the analysis of existing video compression techniques, it was found that the operation of video encoders requires the optimisation of numerous decision parameters to achieve the best trade-off between bit rate and visual quality; given the resource limitations arising from operational constraints such as memory and complexity.

Optimising the performance of the H.264/AVC video encoder has involved finding solutions for multiple conflicting objectives. This task of multi-objective optimisation has been shown to be a difficult process that is computationally expensive. This research has developed an automated tool for optimising video compression to achieve an optimal trade-off between bit rate and visual quality, given maximum allowed memory and computational complexity constraints, within a diverse range of scene environments. The evaluation of this optimisation framework has highlighted the effectiveness of the developed solution. Moreover, the research throughout this thesis has achieved all the proposed objectives described in Chapter 1.

8.2 List of Contributions

This thesis has contributed in optimising the performance of video encoders. The purpose of this thesis is to investigate the suitability of multi-objective optimisation frameworks for enhancing the performance of the H.264 video codec. The main contributions and findings from the research are listed below:

1. **A comprehensive analysis of the effect of varying a selected set of compression parameters on the efficiency of the H.264/AVC video encoder.** From the analysis, the encoding parameters that have a significant impact on the computational complexity, rate-distortion characteristics, and memory utilisation have been identified; these are QP, I-Frame Period, and the Number of B-Frames. It was demonstrated that incrementing the QP by 1 contributes to a 12.5% decrease in bit rate and a 0.4dB decrease in PSNR. The other two coding parameters control the GOP structure. It was found that although intra prediction contributes to around 2.5% of the total computation time, it results in considerable savings when spatial correlation is significant and the motion in the video sequence is minimal. Moreover, the effect of adding more I and B-Frames is evident on the PSNR, where the enhancement is at least 0.25dB for each added frame.

2. **A novel technique for quantitatively assessing the visual quality of image sequences based on human judgement on quality, without the need for a reference image.** This technique was designed to precisely mimic human visual perception of quality. The task involved the training of the developed model to predict the quality of compressed video sequences in a way that correlates very well to the human judgment on quality. A model was developed to find the correlation between the human judgment on quality and a set of objective viewability measures.

3. **The development of a regression model that correlates objective quality metrics to the subjective ones for 5 different scene categories**. The model was trained on a video dataset that involved 600 compressed videos of 5 different categories. Compression parameters were varied within a fixed range with successive levels of compression, as identified in Chapter 4. The visual quality of the compressed videos was then assessed based on qualitative metrics during a number of focus groups. Multiple regression analysis was used to correlate the qualitative and quantitative measures. The outcome of this correlation process was a vector of regression coefficients that was used to predict the qualitative viewability measures from the quantitative counterparts. This chapter was concluded with an evaluation of the differences between the observed and predicted values of visual quality. The evaluation has shown that the proposed model predicted the visual quality for the compressed video test sequences to a close degree, with a small average variance between the predicted and observed Joint Rank values of less than one. This high correlation suggests that there is significant potential for accurately mimicking human visual quality perception using an automated tool. The model developed in this chapter will be used in Chapter 6, where a multi-objective optimisation framework is proposed in order to optimise the quality metrics of compressed videos.

4. **The design of a multi-objective optimisation framework for optimising the identified codec parameters**. The aim of this framework was to improve the compression of images and video sequences acquired from image sensors without compromising visual quality. To deal with such conflicting objectives and to accommodate for the cost-performance trade-offs, a multi-objective optimisation framework was proposed.

5. **The development of a mathematical representation for objective and constraint functions.** Two objective functions relating to rate and distortion were formulated. These objective functions are to be minimised in a memory and CPU resource constrained environment. Therefore, two functions were formulated relating to memory and CPU constraints, which are determined by the system performance requirements. The decision space for this optimisation framework consists of three-dimensional decision variable vectors, encoded as chromosomes, each representing a unique set of decision variables (i.e. compression parameters).

6. **The findings of the evaluation of the multi-objective optimisation framework.** The evaluation process consisted of an extensive set of simulations and validation experiments for the optimisation framework. This process was both time consuming and computationally expensive. However, the output of this optimisation process, i.e. the lookup tables, provides a source of optimised coding parameters, which can be directly referred to for real-time video coding applications. This process has illustrated the high accuracy of the outcome of the optimisation framework, and the potential of using such tools for optimising the H.264/AVC video encoder.

8.3 Recommendations for Future Work

The aforementioned contributions have successfully contributed in addressing the gap in existing literature as described in the summary of this chapter. As with all research, there are areas that require further investigation. However, due to time and resource limitations such areas are recommended for future work.

One area of particular interest would be the development of an automated scene recognition extension, which would allow the video encoder to identify the scene type, therefore, selecting the appropriate coding parameters from the corresponding lookup tables, based on the available system resources. Another potential area for research, involves the incorporation of object-based compression to enable the application of different sets of compression parameters to different parts of the picture.

Moreover, future research could incorporate a different set of scene types where usual video compression techniques may not be appropriate especially when it is vital to preserve high detail in a scene. An example of this is high dynamic range (HDR) imaging from systems such as Infra-Red and thermal cameras. HDR guarantees larger dynamic range luminances between the darkest and lightest areas of an image than standard digital imaging techniques, therefore allowing accurate representation of a wide range of intensity levels found in real scenes.

References

[1] G. J. Sullivan and T. Wiegand, "Video Compression - From Concepts to the H.264/AVC Standard," *Proceeding of the IEEE,* vol. 93, pp. 18, 2005.

[2] G. J. Sullivan, P. Topiwala and A. Luthra, "The H.264/AVC advanced video coding standard: Overview and introduction to the fidelity range extensions," in *Applications of Digital Image Processing XXVII,* 2004, pp. 454-474.

[3] D. A. V. Veldhuizen and G. B. Lamont, "Multiobjective evolutionary algorithms: Analyzing the state-of-the-art," *Evol. Comput.,* vol. 8, pp. 125-147, 2000.

[4] B. L. Deekshatulu, A. D. Kulkarni and G. Kashipati Rao, "Quantitative evaluation of enhancement techniques," *Signal Process,* vol. 8, pp. 369-375, 1985.

[5] D. C. C. Wang, A. H. Vagnucci and C. C. Li, "Digital image enhancement: A survey," *Computer Vision, Graphics, and Image Processing,* vol. 24, pp. 363-381, 1983.

[6] I. E. G. Richardson, *H.264 and MPEG-4 Video Compression : Video Coding for Next-Generation Multimedia.* Chichester: Wiley, 2003.

[7] A. Ford and A. Roberts, "Colour space conversions," Westminster University, London, August. 1998.

[8] ITU-R Recommendation BT.601-4, "Encoding parameters of digital television for studios," .

[9] ITU-T, "Video CODEC for audiovisual services at px64 kbits," Tech. Rep. Tech. Rep. ITU-T Recommendation H.261, March 1993.

[10] ITU-T, "Video coding for low bit rate communication," Tech. Rep. Tech. Rep. ITU-T Recommendation H.263, 1998.

[11] ITU-T and MPEG, "Advanced video coding for generic audiovisual services," Tech. Rep. Tech. Rep. ITU-T Recommendation H.264 and ISO/IEC 14496-10, May 2003.

[12] I. E. Abdou and W. K. Pratt, "Qualitative Design and Evaluation of Enhancement/ Thresholding Edge Detector," *Proceedings of IEEE,* vol. 67, pp. 753-763, 1979.

[13] T. N. Pappas, R. J. Safranek and J. Chen, "Perceptual criteria for image quality evaluation," *Handbook of Image and Video Processing,* pp. 669–684, 2000.

[14] J. Lubin, "The use of psychophysical data and models in the analysis of display system performance," *Digital Images and Human Vision,* pp. 163-178, 1993.

[15] P. C. Teo and D. J. Heeger, "Perceptual image distortion," in *SID INTERNATIONAL SYMPOSIUM DIGEST OF TECHNICAL PAPERS,* 1994, pp. 209-209.

[16] S. Daly, "The visible differences predictor: an algorithm for the assessment of image fidelity, Digital images and human vision," *Cambridge, MA: MIT Press,* pp. 179-206, 1993.

[17] ITU-R, "Recommendation ITU-R BT.500-11: "Methodology for the subjective assessment of the quality of television pictures"," *International Telecommuncation Union, Geneva, Switzerland,* 2002.

[18] Z. Wang, H. R. Sheikh and A. C. Bovik, "Objective video quality assessment," *The Handbook of Video Databases: Design and Applications,* pp. 1041–1078, 2003.

[19] Z. Wang, "Objective Image/Video Quality Measurement—A Literature Survey," *Literature Survey for EE381K: Multidimensional Digital Signal Processing Course Project,* 1998.

[20] ISO/IEC, "Information Technology—Coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbit/s-part 2: Video," Tech. Rep. Tech. Rep. ISO/IEC 11172-2 (MPEG-1 Video), 1993.

[21] ITU-T, "Generic coding of moving pictures and associated audio: Part 2 video," Tech. Rep. Tech. Rep. ITU-T Recommendation H.262 ISO/IEC 13818-2 MPEG-2 Video, 1994.

[22] ISO/IEC, "Coding of audio-visual objects– part 2: Visual," Tech. Rep. Tech. Rep. ISO/IEC 14496-2 (MPEG-4 Part 2: Visual), 2001.

[23] N. Day and J. M. Martinez, "Introduction to MPEG-7 (v2.0)," *ISO/IEC JTC1/SC29/WG11,* vol. 3751, 2000.

[24] ISO/IEC, "MPEG-21 overview," Tech. Rep. Tech. Rep. ISO/IEC JTC1/SC29/WG11/N5231, Oct. 2002. 2002.

[25] ITU-T, "ITU-T H.262 (MPEG-2 video) information technology—Generic coding of moving pictures and associated audio information: Video," Tech. Rep. ISO/IEC, Tech. Rep. 13818-2, 1995.

[26] T. Wiegand, G. J. Sullivan, G. Bjontegaard and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Transactions on Circuits and Systems for Video Technology,* vol. 13, pp. 560-576, 2003.

[27] J. Ostermann, J. Bormans, P. List, D. Marpe, M. Narroschke, F. Pereira, T. Stockhammer and T. Wedi, "Video coding with H. 264/AVC: tools, performance, and complexity," *Circuits and Systems Magazine, IEEE,* vol. 4, pp. 7-28, 2004.

[28] H. Malvar, A. Hallapuro, M. Karczewicz and L. Kerofsky, "Low-complexity transform and quantization in H. 264/AVC," *IEEE Transactions on Circuits and Systems for Video Technology,* vol. 13, pp. 598-603, 2003.

[29] D. Marpe, H. Schwarz and T. Wiegand, "Context-based adaptive binary arithmetic coding in the H. 264/AVC video compression standard," *IEEE Transactions on Circuits and Systems for Video Technology,* vol. 13, pp. 620-636, 2003.

[30] H. Everett, "Generalized Lagrange multiplier method for solving problems of optimum allocation of resources," *Oper. Res.,* vol. 11, pp. 399-417, 1963.

[31] Y. Shoham and A. Gersho, "Efficient bit allocation for an arbitrary set of quantizers [speechcoding]," *IEEE Transactions on Acoustics, Speech and Signal Processing,* vol. 36, pp. 1445-1453, 1988.

[32] P. A. Chou, T. Lookabaugh and R. M. Gray, "Entropy-constrained vector quantization," *IEEE Transactions on Acoustics, Speech and Signal Processing,* vol. 37, pp. 31-42, 1989.

[33] G. J. Sullivan and R. L. Baker, "Rate-distortion optimized motion compensation for video compressionusing fixed or variable size blocks," in *IEEE Global Telecommunications Conf. (GLOBECOM),* 1991, pp. 85-90.

[34] T. Wiegand, M. Lightstone, D. Mukherjee, T. G. Campbell and S. K. Mitra, "Rate-distortion optimized mode selection for very low bit rate video coding and the emerging H. 263 standard," *IEEE Transactions on Circuits and Systems for Video Technology,* vol. 6, pp. 182-190, 1996.

[35] M. C. Chen and A. N. Willson Jr, "Design and optimization of a differentially coded variable blocksize motion compensation system," in *International Conference on Image Processing*, pp. 259, 1996.

[36] G. M. Schuster and A. K. Katsaggelos, "A video compression scheme with optimal bit allocation amongsegmentation, motion, and residual error," *IEEE Trans. Image Process.,* vol. 6, pp. 1487-1502, 1997.

[37] A. Ortega and K. Ramchandran, "Rate-distortion methods for image and video compression," *IEEE Signal Process. Mag.,* vol. 15, pp. 23-50, 1998.

[38] G. J. Sullivan and T. Wiegand, "Rate-distortion optimization for video compression," *IEEE Signal Process. Mag.,* vol. 15, pp. 74-90, 1998.

[39] X. Li, "Enhancements & optimizations to H.264/AVC video coding, PhD thesis," Loughborough University, 2007.

[40] S. H. Ji, J. W. Park and S. D. Kim, "Optimization of memory management for H. 264/AVC decoder," in *Advanced Communication Technology, 2006. ICACT 2006. the 8th International Conference,* pp. 65, 2006.

[41] K. Takagi, Y. Takishima and Y. Nakajima, "A study on rate distortion optimization scheme for JVT coder," in *Proc. of SPIE Vol,* pp. 915.

[42] S. Ma, W. Gao and Y. Lu, "Rate-distortion analysis for H. 264/AVC video coding and its application to rate control," *IEEE Transactions on Circuits and Systems for Video Technology,* vol. 15, pp. 1533-1544, 2005.

[43] J. Zhang, Y. He, S. Yang and Y. Zhong, "Performance and complexity joint optimization for H. 264 video coding," in *Circuits and Systems, 2003. ISCAS'03. Proceedings of the 2003 International Symposium on,* 2003, .

[44] J. Stottrup-Andersen, S. Forchhammer and S. Aghito, "Rate-distortion-complexity optimization of fast motion estimation in H. 264/MPEG-4 AVC," in *Image Processing, 2004. ICIP'04. 2004 International Conference on,* 2004, .

[45] Y. Hu, Q. Li, S. Ma and J. Kuo, "Joint rate-distortion-complexity optimization for H. 264 motion search," in *2006 IEEE International Conference on Multimedia and Expo,* 2006, pp. 1949-1952.

[46] C. Kannangara, I. Richardson, M. Bystrom, J. Solera, Y. Zhao, A. MacLennan and R. Cooney, "Complexity reduction of H. 264 using Lagrange optimization methods," *IEE VIE 2005,(Glasgow, UK,* 2005.

[47] W. Pu, Y. Lu and F. Wu, "Joint power-distortion optimization on devices with MPEG-4 AVC/H. 264 codec," in *IEEE International Conference on Communications, 2006. ICC'06,* 2006, .

[48] Z. He, Y. Liang, L. Chen, I. Ahmad and D. Wu, "Power-rate-distortion analysis for wireless video communication under energy constraints," *IEEE Transactions on Circuits and Systems for Video Technology,* vol. 15, pp. 645-658, 2005.

[49] D. N. Kwon, P. F. Driessen, A. Basso and P. Agathoklis, "Performance and computational complexity optimization in configurable hybrid video coding

system," *IEEE Transactions on Circuits and Systems for Video Technology,* vol. 16, pp. 31-42, 2006.

[50] K. Deb, *Multi-Objective Optimization using Evolutionary Algorithms.* John Wiley & Sons Ltd., 2001.

[51] D. A. Van Veldhuizen and G. B. Lamont, "Multiobjective evolutionary algorithms: Analyzing the state-of-the-art," *Evol. Comput.,* vol. 8, pp. 125-148, 2000.

[52] A. Osyczka, "Multicriteria optimization for engineering design," *Design Optimization,* pp. 193-227, 1985.

[53] A. Abraham and L. Jain, "Evolutionary multiobjective optimization," *Evolutionary Multiobjective Optimization,* pp. 1-6, 2005.

[54] C. A. Coello, "An updated survey of GA-based multiobjective optimization techniques," *ACM Computing Surveys (CSUR),* vol. 32, pp. 143, 2000.

[55] V. Pareto, "Cours d'Economie Politique, volume I and II," *F.Rouge, Lausanne,* vol. 250, 1896.

[56] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: a comparative case studyand the strength Pareto approach," *IEEE Transactions on Evolutionary Computation,* vol. 3, pp. 257-271, 1999.

[57] A. G. Hernández-Díaz, L. V. Santana-Quintero, C. A. Coello Coello and J. Molina, "Pareto-Adaptive ε-dominance," *Evol. Comput.,* vol. 15, pp. 493-517, 2007.

[58] J. D. Schaffer, "Some experiments in machine learning using vector evaluated genetic algorithms (artificial intelligence, optimization, adaptation, pattern recognition)," Vanderbilt University Nashville, TN, USA, 1984.

[59] N. Srinivas and K. Deb, "Muiltiobjective optimization using nondominated sorting in genetic algorithms," *Evol. Comput.,* vol. 2, pp. 221-248, 1994.

[60] K. Deb, A. Pratap, S. Agrawal and T. Meyrivan, "A fast and elitist multi-objective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation,* vol. 6, pp. 182-197, 2002.

[61] K. Deb, S. Agrawal, A. Pratap and T. Meyarivan, "A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II," *Lecture Notes in Computer Science,* vol. 1917/2000, pp. 849-858, 2000.

[62] W. Ren, P. Weal, M. Singh and S. Singh, "Minerva video retrieval benchmark," in *International Conference on Signal Processing, Pattern Recognition and Applications (SPPRA 2006), Innsbruck, Austria,* 2006, pp. 15-17.

[63] K. Sühring, A. M. Tourapis and G. Sullivan. (2005, H.264/AVC reference software by joint video team (JVT) of ISO/IEC MPEG & ITU-T VCEG.

[64] M. Horowitz, A. Joch, F. Kossentini and A. Hallapuro, "H. 264/AVC baseline profile decoder complexity analysis," *IEEE Transactions on Circuits and Systems for Video Technology,* vol. 13, pp. 704-716, 2003.

[65] X. Li, E. Q. Li and Y. K. Chen, "Fast multi-frame motion estimation algorithm with adaptive search strategies in H. 264," in *IEEE International Conference on Acoustics, Speech, and Signal Processing,* 2004, pp. 369–372.

[66] G. Fernández-Escribano, P. Cuenca, L. Orozco-Barbosa and A. Garrido, "Computational complexity reduction of intra-frame prediction in MPEG-2/H. 264 video transcoders," in *Proceeding of ICME,* 2005, .

[67] C. H. Kuo, M. Shen and C. C. J. Kuo, "Fast inter-prediction mode decision and motion search for H. 264," in *Proc. IEEE Int. Conf. Multimedia Expo,* 2004, pp. 663–666.

[68] D. T. Hoang, P. M. Long and J. S. Vitter, "Efficient cost measures for motion estimation at low bit rates," *IEEE Transactions on Circuits and Systems for Video Technology,* vol. 8, pp. 488-500, 1998.

[69] A. Pinar and B. Hendrickson, "Interprocessor communication with memory constraints," in *Proceedings of the Twelfth Annual ACM Symposium on Parallel Algorithms and Architectures,* 2000, pp. 39-45.

[70] T. M. Liu and C. Y. Lee, "Design of an H. 264/AVC Decoder with Memory Hierarchy and Line-Pixel-Lookahead," *Journal of Signal Processing Systems,* vol. 50, pp. 69-80, 2008.

[71] A. Al-Najdawi and R. S. Kalawsky, "Visual Quality Assessment of Video and Image Sequences—A Human-based Approach," *Journal of Signal Processing Systems,* vol. 59, pp. 223-231, 2008.

[72] A. Al-Najdawi and R. S. Kalawsky, "Quantitative quality assessment of video sequences A human-based approach," in *6th International Conference on Information, Communications & Signal Processing,* 2007, pp. 5.

[73] R. A. Krueger and M. A. Casey, *Focus Groups: A Practical Guide for Applied Research.* Pine Forge Pr, 2008.

[74] A. Bryman and E. Bell, *Business Research Methods.* USA: Oxford University Press, 2007.

[75] R. A. Krueger, *Focus Groups: A Practical Guide for Applied Research.* London: Sage Publications, 1994.

[76] D. L. Morgan, "Practical Strategies for Combining Qualitative and Quantitative Methods: Applications to Health Research," *Qualitative Health Research,* vol. 8, pp. 362-376, 1998.

[77] M. Singh, S. Singh and D. Partridge, "A knowledge-based framework for image enhancement in aviation security," *IEEE Transactions on Systems, Man, and Cybernetics, Part B,* vol. 34, pp. 2354-2365, 2004.

[78] M. Sharifi, M. Fathy and M. T. Mahmoudi, "A classified and comparative study of edge detection algorithms," in *Proceedings of the International Conference on Information Technology: Coding and Computing,* 2002, pp. 117-120.

[79] H. Coolican, *Research Methods and Statistics in Psychology.* London: Hodder & Stoughton, 2004.

[80] A. P. Field, *Discovering Statistics using SPSS for Windows: Advanced Techniques for the Beginner.* London: SAGE publications Ltd, 2000.

[81] C. M. Chin, C. M. Tan and M. L. Sim, "Future trends in radio resource management for wireless communications," *BT Technology Journal,* vol. 24, pp. 103-110, 2006.

[82] A. Al-Najdawi and R. S. Kalawsky, "A multi-objective optimization framework for video compression and transmission," in *6th International Symposium on Communication Systems, Networks and Digital Signal Processing, 2008.* 2008, pp. 336-339.

[83] FFmpeg Team, "A complete, cross-platform solution to record, convert and stream audio and video," vol. 2010, March 2, 2010, 2010.

[84] A. Seshadri, "NSGA - II: A multi-objective optimization algorithm," vol. 2008, 19/07/2009, 2009.

[85] P. Pancha; M. El Zarki. "Leaky bucket access control for VBR MPEG video", Proceedings for IEEE INFOCOM, Vol. 95, issue 2, 1995.

Appendix A: Visual Quality Assessment Questionnaire

Dear Respondent,

Please complete the questionnaire as honestly and descriptively as possible. The answers you provide will be vital in the successful development of video quality optimisation technology. Thank you in advance for your time and effort.

Please answer ***all*** questions by circling the appropriate value on the respective scales, unless otherwise stated. All scales range from 1 to 10, with 1 representing the lowest video quality and 10 the highest video quality.

Section 1: Personal Information

1. Gender

☐ Male ☐ Female

2. Age group

☐ Under 25 ☐ 35 – 44 ☐ 55 and over
☐ 25 – 34 ☐ 45 – 54

Section 2: Visual Quality Assessment of Video Sequences

3. Please rate the observed visual quality for each of the following ***News*** videos.

News1_ original	10	**News2_ original**	10	**News3_ original**	10	**News4_ original**	10	**News5_ original**	10
News1_1		News2_1		News3_1		News4_1		News5_1	
News1_2		News2_2		News3_2		News4_2		News5_2	
News1_3		News2_3		News3_3		News4_3		News5_3	
News1_4		News2_4		News3_4		News4_4		News5_4	
News1_5		News2_5		News3_5		News4_5		News5_5	
News1_6		News2_6		News3_6		News4_6		News5_6	
News1_7		News2_7		News3_7		News4_7		News5_7	
News1_8		News2_8		News3_8		News4_8		News5_8	
News1_9		News2_9		News3_9		News4_9		News5_9	
News1_10		News2_10		News3_10		News4_10		News5_10	
News1_11		News2_11		News3_11		News4_11		News5_11	
News1_12		News2_12		News3_12		News4_12		News5_12	
News6_ original	10	**News7_ original**	10	**News8_ original**	10	**News9_ original**	10	**News10_ original**	10
News6_1		News7_1		News8_1		News9_1		News10_1	
News6_2		News7_2		News8_2		News9_2		News10_2	
News6_3		News7_3		News8_3		News9_3		News10_3	
News6_4		News7_4		News8_4		News9_4		News10_4	
News6_5		News7_5		News8_5		News9_5		News10_5	
News6_6		News7_6		News8_6		News9_6		News10_6	
News6_7		News7_7		News8_7		News9_7		News10_7	
News6_8		News7_8		News8_8		News9_8		News10_8	
News6_9		News7_9		News8_9		News9_9		News10_9	
News6_10		News7_10		News8_10		News9_10		News10_10	
News6_11		News7_11		News8_11		News9_11		News10_11	
News6_12		News7_12		News8_12		News9_12		News10_12	

4.	Please rate the observed visual quality for each of the following ***Traffic*** videos.

Traffic1_ original	10	**Traffic2_ original**	10	**Traffic3_ original**	10	**Traffic4_ original**	10	**Traffic5_ original**	10
Traffic1_1		Traffic2_1		Traffic3_1		Traffic4_1		Traffic5_1	
Traffic1_2		Traffic2_2		Traffic3_2		Traffic4_2		Traffic5_2	
Traffic1_3		Traffic2_3		Traffic3_3		Traffic4_3		Traffic5_3	
Traffic1_4		Traffic2_4		Traffic3_4		Traffic4_4		Traffic5_4	
Traffic1_5		Traffic2_5		Traffic3_5		Traffic4_5		Traffic5_5	
Traffic1_6		Traffic2_6		Traffic3_6		Traffic4_6		Traffic5_6	
Traffic1_7		Traffic2_7		Traffic3_7		Traffic4_7		Traffic5_7	
Traffic1_8		Traffic2_8		Traffic3_8		Traffic4_8		Traffic5_8	
Traffic1_9		Traffic2_9		Traffic3_9		Traffic4_9		Traffic5_9	
Traffic1_10		Traffic2_10		Traffic3_10		Traffic4_10		Traffic5_10	
Traffic1_11		Traffic2_11		Traffic3_11		Traffic4_11		Traffic5_11	
Traffic1_12		Traffic2_12		Traffic3_12		Traffic4_12		Traffic5_12	
Traffic6_ original	10	**Traffic7_ original**	10	**Traffic8_ original**	10	**Traffic9_ original**	10	**Traffic10_ original**	10
Traffic6_1		Traffic7_1		Traffic8_1		Traffic9_1		Traffic10_1	
Traffic6_2		Traffic7_2		Traffic8_2		Traffic9_2		Traffic10_2	
Traffic6_3		Traffic7_3		Traffic8_3		Traffic9_3		Traffic10_3	
Traffic6_4		Traffic7_4		Traffic8_4		Traffic9_4		Traffic10_4	
Traffic6_5		Traffic7_5		Traffic8_5		Traffic9_5		Traffic10_5	
Traffic6_6		Traffic7_6		Traffic8_6		Traffic9_6		Traffic10_6	
Traffic6_7		Traffic7_7		Traffic8_7		Traffic9_7		Traffic10_7	
Traffic6_8		Traffic7_8		Traffic8_8		Traffic9_8		Traffic10_8	
Traffic6_9		Traffic7_9		Traffic8_9		Traffic9_9		Traffic10_9	
Traffic6_10		Traffic7_10		Traffic8_10		Traffic9_10		Traffic10_10	
Traffic6_11		Traffic7_11		Traffic8_11		Traffic9_11		Traffic10_11	
Traffic6_12		Traffic7_12		Traffic8_12		Traffic9_12		Traffic10_12	

5.	Please rate the observed visual quality for each of the following ***Sports*** videos.

Sports1_ original	10	**Sports2_ original**	10	**Sports3_ original**	10	**Sports4_ original**	10	**Sports5_ original**	10
Sports1_1		Sports2_1		Sports3_1		Sports4_1		Sports5_1	
Sports1_2		Sports2_2		Sports3_2		Sports4_2		Sports5_2	
Sports1_3		Sports2_3		Sports3_3		Sports4_3		Sports5_3	
Sports1_4		Sports2_4		Sports3_4		Sports4_4		Sports5_4	
Sports1_5		Sports2_5		Sports3_5		Sports4_5		Sports5_5	
Sports1_6		Sports2_6		Sports3_6		Sports4_6		Sports5_6	
Sports1_7		Sports2_7		Sports3_7		Sports4_7		Sports5_7	
Sports1_8		Sports2_8		Sports3_8		Sports4_8		Sports5_8	
Sports1_9		Sports2_9		Sports3_9		Sports4_9		Sports5_9	
Sports1_10		Sports2_10		Sports3_10		Sports4_10		Sports5_10	
Sports1_11		Sports2_11		Sports3_11		Sports4_11		Sports5_11	
Sports1_12		Sports2_12		Sports3_12		Sports4_12		Sports5_12	
Sports6_ original	10	**Sports7_ original**	10	**Sports8_ original**	10	**Sports9_ original**	10	**Sports10_ original**	10
Sports6_1		Sports7_1		Sports8_1		Sports9_1		Sports10_1	
Sports6_2		Sports7_2		Sports8_2		Sports9_2		Sports10_2	
Sports6_3		Sports7_3		Sports8_3		Sports9_3		Sports10_3	
Sports6_4		Sports7_4		Sports8_4		Sports9_4		Sports10_4	
Sports6_5		Sports7_5		Sports8_5		Sports9_5		Sports10_5	
Sports6_6		Sports7_6		Sports8_6		Sports9_6		Sports10_6	
Sports6_7		Sports7_7		Sports8_7		Sports9_7		Sports10_7	
Sports6_8		Sports7_8		Sports8_8		Sports9_8		Sports10_8	
Sports6_9		Sports7_9		Sports8_9		Sports9_9		Sports10_9	
Sports6_10		Sports7_10		Sports8_10		Sports9_10		Sports10_10	
Sports6_11		Sports7_11		Sports8_11		Sports9_11		Sports10_11	
Sports6_12		Sports7_12		Sports8_12		Sports9_12		Sports10_12	

6.	Please rate the observed visual quality for each of the following ***Landscape*** videos.

Lscape1_ original	10	**Lscape2_ original**	10	**Lscape3_ original**	10	**Lscape4_ original**	10	**Lscape5_ original**	10
Lscape1_1		Lscape2_1		Lscape3_1		Lscape4_1		Lscape5_1	
Lscape1_2		Lscape2_2		Lscape3_2		Lscape4_2		Lscape5_2	
Lscape1_3		Lscape2_3		Lscape3_3		Lscape4_3		Lscape5_3	
Lscape1_4		Lscape2_4		Lscape3_4		Lscape4_4		Lscape5_4	
Lscape1_5		Lscape2_5		Lscape3_5		Lscape4_5		Lscape5_5	
Lscape1_6		Lscape2_6		Lscape3_6		Lscape4_6		Lscape5_6	
Lscape1_7		Lscape2_7		Lscape3_7		Lscape4_7		Lscape5_7	
Lscape1_8		Lscape2_8		Lscape3_8		Lscape4_8		Lscape5_8	
Lscape1_9		Lscape2_9		Lscape3_9		Lscape4_9		Lscape5_9	
Lscape1_10		Lscape2_10		Lscape3_10		Lscape4_10		Lscape5_10	
Lscape1_11		Lscape2_11		Lscape3_11		Lscape4_11		Lscape5_11	
Lscape1_12		Lscape2_12		Lscape3_12		Lscape4_12		Lscape5_12	
Lscape6_ original	10	**Lscape7_ original**	10	**Lscape8_ original**	10	**Lscape9_ original**	10	**Lscape10_ original**	10
Lscape6_1		Lscape7_1		Lscape8_1		Lscape9_1		Lscape10_1	
Lscape6_2		Lscape7_2		Lscape8_2		Lscape9_2		Lscape10_2	
Lscape6_3		Lscape7_3		Lscape8_3		Lscape9_3		Lscape10_3	
Lscape6_4		Lscape7_4		Lscape8_4		Lscape9_4		Lscape10_4	
Lscape6_5		Lscape7_5		Lscape8_5		Lscape9_5		Lscape10_5	
Lscape6_6		Lscape7_6		Lscape8_6		Lscape9_6		Lscape10_6	
Lscape6_7		Lscape7_7		Lscape8_7		Lscape9_7		Lscape10_7	
Lscape6_8		Lscape7_8		Lscape8_8		Lscape9_8		Lscape10_8	
Lscape6_9		Lscape7_9		Lscape8_9		Lscape9_9		Lscape10_9	
Lscape6_10		Lscape7_10		Lscape8_10		Lscape9_10		Lscape10_10	
Lscape6_11		Lscape7_11		Lscape8_11		Lscape9_11		Lscape10_11	
Lscape6_12		Lscape7_12		Lscape8_12		Lscape9_12		Lscape10_12	

7.	Please rate the observed visual quality for each of the following ***UMV*** videos.

UMV1_ original	10	**UMV2_ original**	10	**UMV3_ original**	10	**UMV4_ original**	10	**UMV5_ original**	10
UMV1_1		UMV2_1		UMV3_1		UMV4_1		UMV5_1	
UMV1_2		UMV2_2		UMV3_2		UMV4_2		UMV5_2	
UMV1_3		UMV2_3		UMV3_3		UMV4_3		UMV5_3	
UMV1_4		UMV2_4		UMV3_4		UMV4_4		UMV5_4	
UMV1_5		UMV2_5		UMV3_5		UMV4_5		UMV5_5	
UMV1_6		UMV2_6		UMV3_6		UMV4_6		UMV5_6	
UMV1_7		UMV2_7		UMV3_7		UMV4_7		UMV5_7	
UMV1_8		UMV2_8		UMV3_8		UMV4_8		UMV5_8	
UMV1_9		UMV2_9		UMV3_9		UMV4_9		UMV5_9	
UMV1_10		UMV2_10		UMV3_10		UMV4_10		UMV5_10	
UMV1_11		UMV2_11		UMV3_11		UMV4_11		UMV5_11	
UMV1_12		UMV2_12		UMV3_12		UMV4_12		UMV5_12	
UMV6_ original	10	**UMV7_ original**	10	**UMV8_ original**	10	**UMV9_ original**	10	**UMV10_ original**	10
UMV6_1		UMV7_1		UMV8_1		UMV9_1		UMV10_1	
UMV6_2		UMV7_2		UMV8_2		UMV9_2		UMV10_2	
UMV6_3		UMV7_3		UMV8_3		UMV9_3		UMV10_3	
UMV6_4		UMV7_4		UMV8_4		UMV9_4		UMV10_4	
UMV6_5		UMV7_5		UMV8_5		UMV9_5		UMV10_5	
UMV6_6		UMV7_6		UMV8_6		UMV9_6		UMV10_6	
UMV6_7		UMV7_7		UMV8_7		UMV9_7		UMV10_7	
UMV6_8		UMV7_8		UMV8_8		UMV9_8		UMV10_8	
UMV6_9		UMV7_9		UMV8_9		UMV9_9		UMV10_9	
UMV6_10		UMV7_10		UMV8_10		UMV9_10		UMV10_10	
UMV6_11		UMV7_11		UMV8_11		UMV9_11		UMV10_11	
UMV6_12		UMV7_12		UMV8_12		UMV9_12		UMV10_12	

Thank you for your time and effort