University of Southampton
Faculty of Engineering
School of Electronics and Computer Science
Southampton S017 1BJ

Wireless Video Communications

by

Jin Yee Chung
B. Eng.

A doctoral thesis submitted in partial fulfilment of the
requirements for the award of Doctor of Philosophy
at the University of Southampton

May 2004

SUPERVISOR: *Professor Lajos Hanzo*
Dipl Ing, MSc, PhD, SMIEEE
Chair of Telecommunications
School of Electronics and Computer Science
University of Southampton
Southampton SO17 1BJ
United Kingdom

This Thesis is dedicated
to My family in Borneo, Malaysia
and
in memory of my beloved grandmother.

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

FACULTY OF ENGINEERING AND APPLIED SCIENCE
SCHOOL OF ELECTRONICS AND COMPUTER SCIENCE

Doctor of Philosophy

Wireless Video Communications

by Jin Yee Chung

This thesis explores the feasibility of providing video services for mobile users. The design trade-offs of the MPEG-4 video compression standard ratified by the Moving Picture Expert Group (MPEG), as well as those of ITU-T study group's H.263 codec and the Video Coding Expert Group's (VCEG) H.264 video codec are investigated. More specifically, these two video codecs are capable of efficiently compressing video and yet achieving a remarkably high reconstructed video quality. The MPEG-4 video bitstream is subjected to a rigourous error sensitivity investigation, in order to assist in contriving various error protection schemes for wireless videophony systems.

For the sake of achieving error resilience, the source codec has to make provisions for error detection, resynchronisation and error concealment. Thus a packetisation technique invoking adaptive bit-rate control was used in conjunction with the various modulation scheme employed. A Burst-by-Burst (BbB) Adaptive Coded Modulation-Aided Joint Detection-Based CDMA (ACM-JD-CDMA) scheme has been proposed for wireless video telephony and its performance was characterised, when communicating over the third-generation UTRA system's wide-band vehicular fading channels. The coded modulation schemes invoked in our fixed modulation mode based systems are Low Density Parity Check code based Block Coded Modulation (LDPC-BCM) and Turbo Trellis Coded Modulation (TTCM). The performance of LDPC-BCM was evaluated and compared to that of TTCM in the context of the ACM-JD-CDMA system using a practical modem mode switching regime. Both schemes exhibited a similar transmission integrity, although the LDPC arrangement was capable of achieving this at a lower complexity.

We concluded our investigations with a comparative study of three sophisticated video transceivers. The MPEG-4 video codec was amalgamated with various channel coding and modulation schemes, such as an iterative parallel interference cancellation aided CDMA scheme as well as a more sophisticated space-time spreading assisted multi-carrier DS-CDMA video transceiver. Finally, a turbo-detected unequal error protection MPEG-4 video telephone system using a serially concatenated convolutional outer code, trellis coded modulation based inner code and space-time coding aided transmit diversity scheme was proposed. Our conclusion was that the latter scheme performed best, facilitating wireless video communications at a signal-to-noise ratio as low as about 4.5 dB.

Acknowledgements

I am indebted to many individuals for the care and support they gave me during my doctoral studies. First of all, I would like to express my deep gratitude to my supervisor, Prof. Lajos Hanzo, for his constant encouragement, guidance, kindness and patience. This project would not have progressed far without his involvement; especially when things were not going to plan. All his help, advice, ideas and encouragement are greatly appreciated and I am very thankful for his wonderful friendship.

I would also like to take this opportunity to thank all my previous and current colleagues in the Communications Group. They created such a great research environment and were always ready to lend a helping hand. Special thanks to Peter Cherriman, Chee Siong Lee, Jin Wang and Robert Mounder for sharing their knowledge in video coding. I am grateful to Soon Xin Ng, Mohamad Yusof Alias, Guo Feng and Hua Wei with whom I enjoyed working in our fruitful collaborations. Warmest thanks also to Lie Liang Yang, Bee Leong Yeap, Tong Hooi Liew, Mong Suan Yee, Hee Tong How, Byoung Jo Choi, Nurul Nadia Ahmad, Noor Shamsiah Othman, Wei Liu and Song Ni all other colleagues and staff. The assistance provided by Denise Harvey and Rebecca Earl, the Communications Group secretaries, is gratefully acknowledged.

In addition, I offer my heartfelt thanks to all my friends in Southampton and elsewhere, for their constant kind wishes and the good times that we shared together. Several friends have had a special impact on me during these few years. I would like to thank Helen Pik Shan Tam for her nursing when my hand was injured. She was also an excellent lunch partner who gladly discussed anything from signal processing to current affairs. Also thanks go to Yi Fui Liang, Joanna Lee, Jin San Yong, Kenneth Tso, John Smythe and Martin E. Cairns for their constant encouragement. I also owe sincere thanks to my family especially my parents in Borneo, Malaysia and brother Jin Thau who encourage and support me in all my decisions. Without you all, this thesis would never have been completed.

List of Publications

1. **J. Y. Chung, S. X. Ng, E. L. Kuan and L. Hanzo**, "Burst-by-burst adaptive coded modulation-aided joint detection-based CDMA for wireless video telephony", Proceedings of IEEE Vehicular Technology Conference Spring, Birmingham, USA, May 2002, Volume 3, pp. 1317-1321.

2. **J. Y. Chung, M. Y. Alias, F. Guo and L. Hanzo**, "LDPC and Turbo Coding Assted Space-Time Block Coded OFDM for H.26L", Proceedings of PIMRC'2003, Beijing, China, September 2003, pp. 2702-2706.

3. **J.Y. Chung, F. Guo, S.X. Ng and L.Hanzo**, "Multi-Mode Joint-Detection CDMA/H.26L Based Wireless Video Telephony", Proceedings of IEEE Vehicular Technology Conference Fall, Orlando, Florida, USA, October 2003, Volume 3 , pp. 6-9.

4. **S. X. Ng, J.Y. Chung, P. Cherriman and L. Hanzo** , "Burst-by-Burst Adaptive Decision Feedback Equalised TCM, TTCM and BICM for H.263-Assisted Wireless Video Telephony ", *submitted to* IEEE Transactions on Vehicular Technology.

5. **S. X. Ng, J. Y. Chung, F. Guo and L. Hanzo**, "Turbo-detection Aided Serially Concatenated MPEG-4/TCM Videophone Transceiver", *submitted to* VTC Fall 2004.

6. **J. Y. Chung, H. Wei and L. Hanzo**, "Iterative Parallel Interference Cancellation Aided CDMA Based MPEG-4 Video", *submitted to* 3G 2004.

Contents

Chapter 1

Introduction

1.1 A Brief Introduction to Compression Theory

The ultimate aim of data compression is the removal of redundancy from the source information. This, therefore, reduces the number of binary bits required for representing the information contained within the source. Achieving the best possible compression ratio requires not only an understanding of the nature of the source signal in its binary representation, but also how we as humans interpret the information that the data represents.

We live in a world of rapidly improving computing and communications capabilities, and owing to an unprecedented increase in computer awareness, the demand for computer systems and their applications is also drastically increased. Since the transmission or storage of every single bit incurs a cost, the advancement of cost-efficient source-signal compression techniques is of high significance. When considering the transmission of source information that may contain a substantial amount of redundancy, achieving a high compression ratio is of paramount importance.

In a simple system, the same number of bits might be used for representing the symbols *'e'* and *'q'*. Statistically speaking, it can be shown however that the character *'e'* appears in English text more frequently than the character *'q'*. Hence, upon representing the more-frequent symbols with fewer bits than the less-frequent symbols we stand to reduce the total number of bits necessary for encoding the entire information transmitted or stored.

Indeed a number of source-signal encoding standards have been formulated based upon the removal of predictability or redundancy from the source. The most widely used principle dates back to the 1940s and is referred to as Shannon-Fano coding [1,2], while the well-known Huffman encoding scheme was contrived in 1952 [3]. These approaches, however, have been further enhanced many times since then and have been invoked in various applications. Further research will undoubtedly endeavour

to continue improving upon those techniques, asymptotically approaching the information theoretic limits.

Digital video compression techniques [4] have played an important role in the world of wireless telecommunication and multimedia systems, where bandwidth is a valuable commodity. Hence, the employment of video compression techniques is of prime importance for the sake of reducing the amount of information that has to be transmitted for adequately representing a picture sequence without impairing its subjective quality, as judged by human viewers. Modern compression techniques involve complex algorithms which have to be standardised for the sake of global compatibility and interworking.

1.2 Evolution of Video Compression Standards

Digital video signals may be compressed by numerous different proprietary or standardised algorithms. The most important families of compression algorithms are published by recognised standardisation bodies, such as the International Standards Organisation (ISO), the International Telecommunication Union (ITU) or the Motion Picture Expert Group (MPEG). By contrast, proprietary compression algorithms developed and owned by a smaller interest group are of lesser significance owing to their lack of global compatibility and interworking capability. The evolution of video compression standards over the past half-a-century may be followed in Figure 1.1

As seen in the figure, the history of video compression commences in the 1950s. An analogue videophone system had been designed, constructed and trialled in the 1960s, but it required a high bandwidth and it was deemed that using the postcard-size black-and-white pictures produced did not substantially augment the impression of telepresence in comparison to conventional voice communication. In the 1970s, it was realised that visual identification of the communicating parties may be expected to substantially improve the value of multi-party discussions and hence the introduction of videoconference services was considered. The users' interest increased in parallel to improvements in picture quality.

Video coding standardisation activities started in the early 1980s. These activities were initiated by the *International Telegraph and Telephone Consultative Committee* (CCITT) [24], which is currently known as the *International Telecommunications Union - Telecommunication Standardisation Sector* (ITU-T) [12]. These standardisation bodies were later followed by the formation of the Consultative Committee for International Radio (CCIR)(currently ITU-R) [26], the *International Organisation for Standardisation* (ISO), and the *International Electrotechnical Commission* (IEC). These bodies coordinated the formation of various standards, some of which are listed in Table 1.1 and will be discussed further in the following sections.

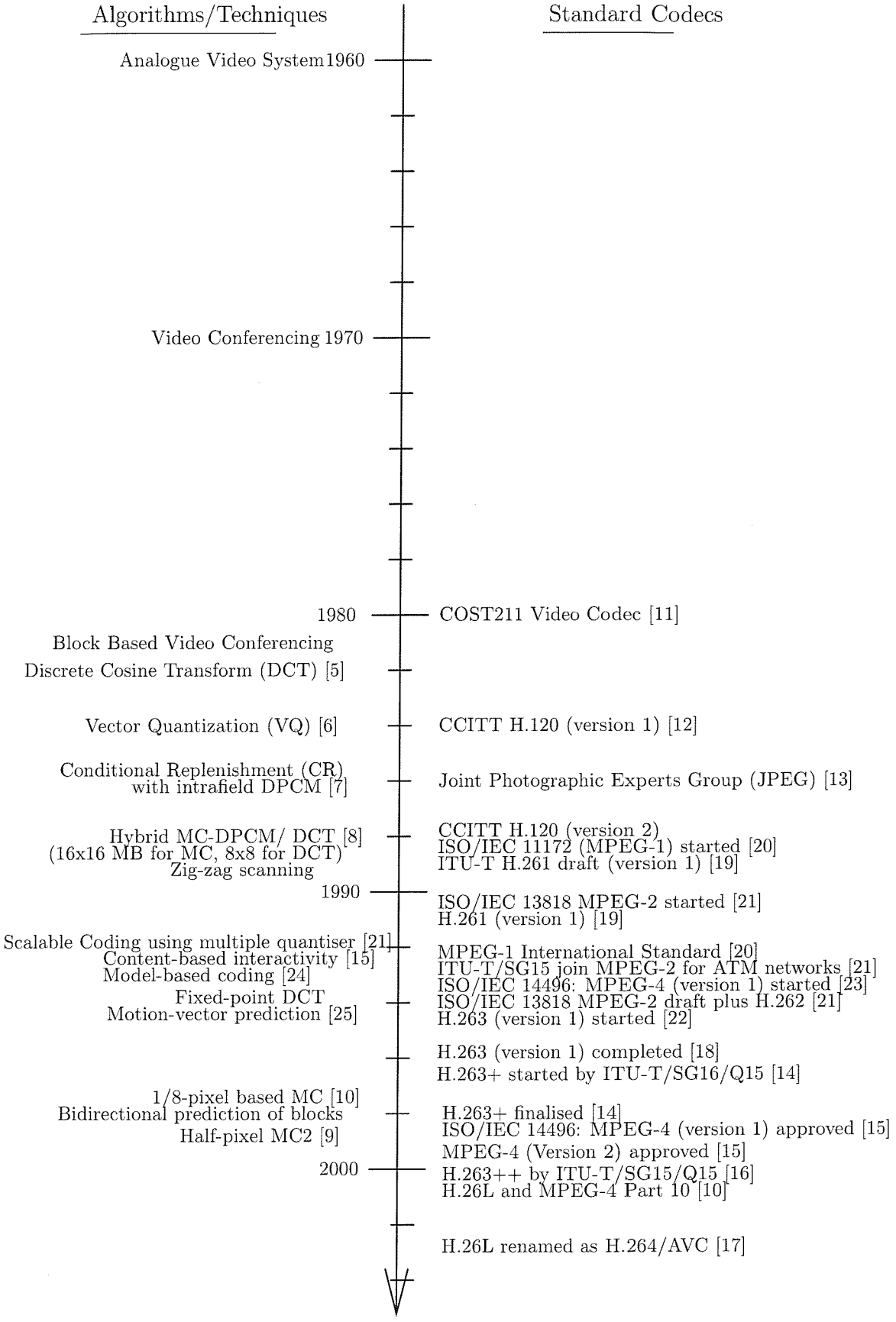


Figure 1.1: A brief history of video compression

Date	Standard
1956	AT&T designs and construct the first Picturephone test system [27]
1964	AT&T introduces Picturephone at the World's Fair, New York [27]
1970	AT&T offers Picturephone for $160 per month [27]
1971	Ericsson demonstrates the first Trans-Atlantic video telephone (LME) call
1973 Dec.	ARPAnet packet voice experiments
1976 March	Network Voice Protocol (NVP), by Danny Cohen, USC/ISI [28]
1981 July	Packet Video Protocol (PVP), by Randy Cole, USC/ISI [29]
1982	CCITT (predecessor of the ITU-T) standard H.120 (2 Mbit/s) video coding, by European COST 211 project [12]
1982	Compression Labs begin selling $250,000 video conference (VC) system, $1,000 per hour lines
1986	PictureTel's $80,000 VC system, $100 per hour lines
1987	Mitsubishi sells $1,500 still-picture phone
1989	Mitsubishi drops still-picture phone
1990	TWBnet packet audio/video experiments, portable video players (pvp) (video) from Information Science Institute (ISI)/Bolt, Beranek and Newman, Inc.(BBN) [30]
1990	CCITT standard H.261 (p x 64) video coding [19]
1990 Dec.	CCITT standard H.320 for ISDN conferencing [31]
1991	PictureTel unveils $20,000 black-and-white VC system, $30 per hour lines
1991	IBM and PictureTel demonstrate videophone on PC
1991 Feb.	DARTnet voice experiments, Voice Terminal (VT) program from USC/ISI [32]
1991 June	DARTnet research's packet video test between ISI and BBN. [32]
1991 Aug.	University of California, Berkeley (UCB)/Lawrence Berkeley National Laboratories (LBNL)'s audio tool vat releases for DARTnet use [32]
1991 Sept.	First audio/video conference (H.261 hardware codec) at DARTnet [32]
1991 Dec	dvc (receive-only) program, by Paul Milazzo from BBN, Internet Engineering Task Force (IETF) meeting, Santa Fe [33]
1992	AT&T's $1,500 videophone for home market [27]
1992 March	World's first Multicasr BackBONE (MBone) audio cast (vat), 23rd IETF, San Diego
1992 July	MBone audio/video casts (vat/dvc), 24th IETF, Boston
⋮	⋮

⋮	⋮
1992 July	Institute National de Recherche en Informatique et Automatique (INRIA) Videoconferencing System (ivs), by Thierry Turletti from INRIA [34]
1992 Sept.	CU-SeeMe v0.19 for Macintosh (without audio), by Tim Dorcey from Cornell University [35]
1992 Nov.	Network Video (nv) v1.0, by Ron Frederick from Xerox's Palo Alto Research Center (Xerox PARC), 25th IETF, Washington DC
1992 Dec.	Real-time Transport Protocol (RTP) v1, by Henning Schulzrinne [36]
1993 April	CU-SeeMe v0.40 for Macintosh (with multipoint conferencing) [37]
1993 May	Network Video (NV) v3.2 (with color video)
1993 Oct.	VIC Initial Alpha, by Steven McCanne and Van Jacobson from UCB/LBNL
1993 Nov.	VocalChat v1.0, an audio conferencing software for Novell IPX networks
1994 Feb.	CU-SeeMe v0.70b1 for Macintosh (with audio) , audio code by Charley Kline's Maven [37]
1994 April	CU-SeeMe v0.33b1 for Windows (without audio), by Steve Edgar from Cornell [37]
1995 Feb.	VocalTec Internet Phone v1.0 for Windows (without video) [38]
1995 Aug.	CU-SeeMe v0.66b1 for Windows (with audio) [37]
1996 Jan.	Real-time Transport Protocol (RTP) v2, by IETF avt-wg
1996 March	ITU-T standard H.263 (p x 8) video coding for low bit-rate communication [18]
1996 March	VocalTec Telephony Gateway [39]
1996 May	ITU-T standard H.324 for Plain Old Telephone System (POTS) conferencing [40]
1996 July	ITU-T standard T.120 for data conferencing [41]
1996 Aug.	Microsoft NetMeeting v1.0 (without video)
1996 Oct.	ITU-T standard H.323 v1, by ITU-T SG 16 [42]
1996 Nov.	VocalTec Surf&Call, the first Web to phone plugin
1996 Dec.	Microsoft NetMeeting v2.0b2 (with video)
1996 Dec.	VocalTec Internet Phone v4.0 for Windows (with video) [38]
1997 July	Virtual Room Videoconferencing System (VRVS), Caltech-CERN project [43]
1997 Sept.	Resource ReSerVation Protocol (RSVP) v1 [44]
1998 Jan.	ITU-T standard H.323 v2 [45]
1998 Jan.	ITU-T standard H.263 v2 (H.263+) video coding [14]
⋮	⋮

⋮	⋮
1998 April	CU-SeeMe v1.0 for Windows and Macintosh (using colour video), from Cornell University, USA [37]
1998 May	Cornell's CU-SeeMe development team has completed their work [37]
1998 Oct.	ISO/IEC standard MPEG-4 v1, by ISO/IEC JTC1/SC29/WG11 (MPEG) [15]
1999 Feb.	Session Initiation Protocol (SIP) makes proposed standard, by IETF music-work group [46]
1999 April	Microsoft NetMeeting v3.0b
1999 Aug.	ITU-T H.26L Test Model Long-term (TML) project, by ITU-T SG16/Q.6 (VCEG) [10]
1999 Sept.	ITU-T standard H.323 v3 [47]
1999 Oct.	Network Address Translation (NAT) compatible version of iVisit, v2.3b5 for Windows and Macintosh
1999 Oct.	Media Gateway Control Protocol (MGCP) v1, IETF
1999 Dec.	Microsoft NetMeeting v3.01 service pack 1 (4.4.3388)
1999 Dec.	ISO/IEC standard MPEG-4 v2
2000 May	Columbia SIP user agent sipc v1.30 2000 Oct: Samsung releases the first MPEG-4 streaming 3G (CDMA2000-1x) video cell phone
2000 Nov.	ITU-T standard H.323 v4 [48]
2000 Nov.	MEGACO/H.248 Protocol v1, by IETF megaco-wg and ITU-T SG 16
2000 Dec.	Microsoft NetMeeting v3.01 service pack 2 (4.4.3396))
2000 Dec.	ISO/IEC Motion JPEG 2000 (JPEG 2000, Part 3) project, by ISO/IEC JTC1/SC29/WG1 (JPEG) [13]
2001 June	Windows XP Messenger supports the SIP
2001 Sept.	World's first Trans-Atlantic gallbladder surgery using a video phone (by surgeon Lindbergh)
2001 Oct.	NTT DoCoMo sells $570 3G (WCDMA) mobile videophone
2001 Oct.	TV reporters use $7,950 portable satellite videophone to broadcast live from Afghanistan
2001 Oct.	Microsoft NetMeeting v3.01 (4.4.3400) on XP
2001 Dec.	Joint Video Team (JVT) video coding (H.26L and MPEG-4 Part 10) project, by ITU-T SG16/Q.6 (VCEG) and ISO/IEC JTC1/SC29/WG 11 (MPEG) [10]
2002 June	World's first 3G video cell phone roaming
2002 Dec.	JVT completes the technical work leading to ITU-T H.264 [17]
2003	Wireless video telephony commercialized

Table 1.1: Evolution of video communications

1.2.1 The International Telecommunications Union's H.120 Standard

Using state-of-the-art technology in the 1980s, a video encoder/decoder (codec) was designed by the Pan-European Cooperation in Science and Technology (COST) project 211, which was based on Differential Pulse Code Modulation (DPCM) and was ratified by the CCITT as the H.120 standard [49]. This codec's target bitrate was 2 Mbit/s for the sake of compatibility with the European Pulse Code Modulatated (PCM) bitrate hierarchy in Europe and 1.544 Mbit/s for North America [49], which was suitable for convenient mapping to their respective first levels of digital transmission hierarchy. Although the H.120 standard had a good spatial resolution, since Differential PCM (DPCM) operates on a pixel-by-pixel basis, it had a poor temporal resolution. It was soon realised that in order to improve the image quality without exceeding the above-mentioned 2Mbit/s target bitrate, less than one bit should be used for encoding each pixel. This was only possible, if a group of pixels for example a "block" of 8 × 8-pixel , were encoded together, such that the number of bits per pixel used may become non-integer. This led to the design of so-called block-based codecs. More explicitly, at 2Mbit/s and at a framerate of 30 frames/second the maximum number of bits per frame was approximately 66.67 *kbits*. Using black and white pictures at 176 × 144 pixel resolution, the maximum number of bit per pixel was 2 bit .

1.2.2 Joint Photographic Expert Group

During the late 1980s, 15 different block-based video conferencing proposals were submitted to the ITU-T standard body (formerly the CCITT), and 14 of these were based on using the Discrete Cosine Transform (DCT) [5] for still-image compression, while one used Vector Quantisation (VQ) [6]. The subjective quality of video sequences presented to the panel judges showed hardly any perceivable difference between the two types of coding techniques. In parallel to the ITU-T's investigations conducted during the period of 1984-88 [13], the Joint Photographic Experts Group (JPEG) was also coordinating the compression of static images. Again, they opted for the DCT as the favoured compression technique, mainly due to their interest in progressive image transmission. JPEG's decision undoubtedly influenced the ITU-T in favouring the employment of DCT over VQ. By this time there was a worldwide activity in implementing the DCT in chips and on Digital Signal Processors (DSPs).

1.2.3 The ITU H.261 Standard

During the late 1980s it became clear that the recommended ITU-T video conferencing codec would use a combination of motion-compensated inter-frame coding and the DCT. The codec exhibited a substantially improved video quality in comparison to the DPCM-based H.120 standard. In fact, the image quality was found sufficiently high for video conferencing applications at 384 kbits/s and good

quality was attained using 352 × 288-pixel Common Intermediate Format (CIF) or 176 × 144-pixel Quarter CIF (QCIF) images at bitrates of around 1 Mbit/s. The H.261 codec [19] was capable of using 31 different quantisers and various other adjustable coding options, hence its bitrate spanned a wide range. Naturally, the bitrate depended on the motion-activity and the video format, hence was not perfectly controllable. Nonetheless, the H.261 scheme was termed as a $p \times 64$ bits/s codec, $p = 1 \ldots\ldots 30$ to comply with the bitrates provided by the ITU's PCM hierarchy. The standard was ratified in late 1989.

1.2.4 The Motion Picture Expoer Group (MPEG-1)

In the early 1990s, the Motion Picture Experts Group (MPEG) was created as Sub-Committee (SC) 2 of ISO (ISO/SC2). The MPEG started investigating the conception of coding techniques specifically designed for the storage of video, in media such as CD-ROMs. The aim was to develop a video codec capable of compressing highly motion-active video scenes, such as those seen in movies for storage on hard disks, while maintaining a performance comparable to that of Video Home System (VHS) video-recorder quality. In fact, the basic MPEG-1 standard [20], which was reminiscent of the H.261 ITU codec [19], was capable of accomplishing this task at a bitrate of 1.5 Mbit/s. Since for the storage of distributive, rather than interactive of video, the encoding and decoding delays do not constitute a major constraint, one can trade delay for compression efficiency. Hence, in contrast to the H.261 interactive codec, which had a single-frame video delay, the MPEG-1 codec introduced the bidirectionally predicted frames in its motion-compensation scheme.

At the time of writing, MPEG decoders/players are becoming commonly used place for the storage of multimedia information on computers. MPEG-1 decoder plug-in hardware boads (e.g. MPEG magic cards) have been around for a while, and software-based MPEG-1 decoders are available with the release of operating systems or multimedia extensions for Personal Computer (PC) and Macintosh platforms.

MPEG-1 was originally optimised for typical applications using non-interlaced video sequences scanned at 25 frames per second (fps) in European format and at 29.9 fps in North American format. The bitrate of 1.2 to 1.5 Mbits/s typically results in an image quality comparable to home Video Cassette Recorders (VCRs) [20] using CIF images, which can be further improved at higher bitrates. Early versions of the MPEG-1 codec used for encoding interlaced video, such as those employed in broadcast applications were referred to as MPEG1+.

1.2.5 The MPEG-2 Standard

A new generation of MPEG coding schemes referred to as MPEG-2 [21] was also adopted by broadcasters who were initially reluctant to use any compression of video sequences. The MPEG-2 scheme encodes CIF-resolution codes for interlaced video at bit rates of 4-9 Mbits/s, and is now well on its way to making a significant impact in a range of applications, such as digital terrestrial broadcasting, digital satellite TV [4], digital cable TV, digital versatile disc (DVD) and many others. Television broadcasters started using MPEG-2 encoded digital video sequences during the late 1990s [21].

A slightly improved version of MPEG-2, termed as MPEG-3, was to be used for the encoding of High Definition TV (HDTV), but since MPEG-2 itself was capable of achieving this, the MPEG-3 standards were folded into MPEG-2. It is foreseen that by the year 2014, the existing transmission of NTSC format TV programmes will cease in North America and instead HDTV employing MPEG-2 compression will be used in terrestrial broadcasting.

1.2.6 The ITU H.263 Standard

The H.263 video codec was designed by the ITU-T standardisation body for low-bitrate encoding of video sequences in videoconferencing [18]. It was first designed to be utilised in H.323 based systems [45], but it has also been adopted for Internet-based video conferencing .

The encoding algorithms of the H.263 codec are similar to those used by its predecessor, namely the H.261 codec, although both its coding efficiency and error resilience have been improved at the cost of a higher implementational complexity [4]. Some of the main differences between the H.261 and H.263 coding algorithms are listed below. In the H.263 codec, half-pixel resolution is used for motion compensation, whereas H.261 used full-pixel precision in conjunction with a smoothing filter invoked for removing the high-frequency spatial changes in the video frame, which improved the achievable motion-compensation efficiency. Some parts of the hierarchical structure of the data-stream are now optional in the H.263 scheme, hence the codec may be configured for attaining a lower data-rate or better error resilience. There are four negotiable options included in the standard for the sake of potentially improving the attainable performance provided that both the encoder and decoder are capable of activating them [4]. These allow the employment of unrestricted motion vectors, syntax-based arithmetic coding, advanced prediction modes as well as both forward- and backward-frame prediction. The latter two options are similar to the MPEG codec's Predicted (P) and Bidirectional (B) modes.

1.2.7 The ITU H.263+/H.263++ Standards

The H.263+ scheme constitutes version 2 of the H.263 standard [14]. This version was developed by the ITU-T/SG16/Q15 Advanced Video Experts Group, which previously operated under ITU-T/SG15. The technical work was completed in 1997 and was approved in 1998. The H.263+ standard incorporated 12 new optional features in the H.263 codec. These new features support the employment of customised picture sizes and clock frequencies, improved compression efficiency, allow for quality, bitrate and complexity scalability. Furthermore, it has the ability to enhance the attainable error resilience, when communicating over wireless and packet-based networks, while supporting backward compatibility with the H.263 codec. The H.263++ scheme is version 3 of the H.263 standard [16], which was developed by ITU-T/SG16/Q15. Its technical content was completed and approved late in the year 2000.

1.2.8 The MPEG-4 Standard

The MPEG-4 standard is constituted by a family of audio and video coding standards that are capable of covering an extremely wide bitrate range, spanning from 4800 bit/s to approximately 4 Mbit/s [15]. The primary applications of the MPEG-4 standard are found in Internet-based multimedia streaming and CD distribution, conversational videophone as well as broadcast television.

The MPEG-4 standard family absorbs many of the MPEG-1 and MPEG-2 features, adding new features such as Virtual Reality Markup Lanuage (VRML) support for 3D rendering, object-oriented composite file handling including audio, video and VRML objects, the support of digital rights management and various other interactive applications.

Most of the optional features included in the MPEG-4 codec may be expected to be exploited in innovative future applications yet to be developed. At the time of writing, there are very few complete implementations of the MPEG-4 standard. Anticipating this, the developers of the standard added the concept of 'Profiles', allowing various capabilities to be grouped together.

As mentioned above, the MPEG-4 codec family consists of several standards, which are termed "Layers" and are listed below [15]:

- Layer 1: Describes the synchronisation and multiplexing of video and audio;
- Layer 2: Compression algorithms for video signals;
- Layer 3: Compression algorithms for perceptual coding of audio signals;
- Layer 4: Describes the procedures derived for compliance testing;
- Layer 5: Describes systems for software simulation of the MPEG-4 framework;

- Layer 6: Describes the Delivery Multimedia Integration Framework (DMIF).

1.2.9 The H.26L/H.264 Standard

Following the finalisation of the original H.263 standard designed for videotelephony, which was completed in 1995, the ITU-T Video Coding Experts Group (VCEG) commenced work on two further developments, specifically, a "short-term" effort included adding extra features to the H.263 codec, resulting in Version 2 of the standard and a "long-term" effort aiming for developing a new standard specifically designed for low bitrate visual communications. The long-term effort led to the draft "H.26L" standard, offering a significantly better video compression efficiency than the previous ITU-T standards. In 2001, the ISO MPEG recognised the potential benefits of H.26L and the Joint Video Team (JVT) was formed, including experts from both the MPEG and the VCEG. The JVT's main task was to develop the draft H.26L model1 into a full international standard. In fact, the outcome of these efforts turned out to be two identical standards, namely the ISO MPEG4 Part 10 scheme of MPEG4 and the ITU-T H.264 codec. The official terminology for the new standard is Advanced Video Coding (AVC), although, it is widely known by its old working title of H.26L and by its ITU document number H.264 [50].

In common with earlier standards, such as the MPEG1, MPEG2 and MPEG4 schemes, the H.264 draft standard does not explicitly define an unambiguous coding standard. Rather, it defines the syntax of an encoded video bitstream and the decoding algorithm for this bitstream. The basic functional elements, such as motion prediction, transformation of the motion-compensated error residual and the quantisation of the resultant DCT coefficients as well as their entropy encoding are not unlike those of the previous standards, such as MPEG1, MPEG2, MPEG4, H.261, H.263, etc. The important advances found in the H.264 codec occur in the specific implementation of each functional element. The H.264 codec is described in more detail in Section 3.2.

This thesis reports on advances attained during the most recent three years of the half-a-century history of video communications, focussing on the design aspects of wireless video telephony, dedicating particular attention to the contradictory design aspects portrayed in Figure 1.2 [4].

1.3 Video Communications

Video communication over rate-limited and error-prone channels, such as packet networks and wireless links, requires both a high error resilience and high compression. In the past, considerable efforts have been invested in the design and development of the most efficient video compression schemes and standards. For the sake of achieving high compression, most modern codecs employ motion-compensated prediction between video frames, in order to reduce the temporal redundancy, followed by a spatial

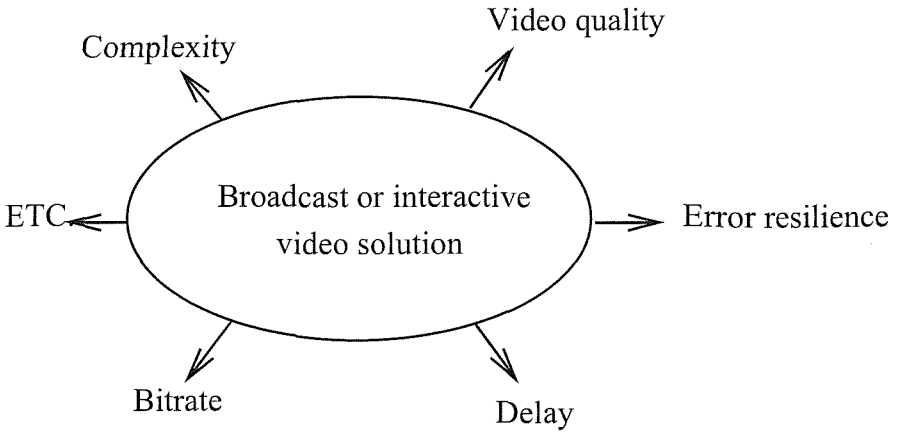


Figure 1.2: Contradictory system design requirements of various video communications systems

transform invoked for reducing the spatial redundancy. The resultant parameters are entropy-coded, in order to produce the compressed bitstream. These algorithms provide a high compression, however, the compressed signal becomes highly vulnerable to error-induced data losses, which is particularly detrimental when communicating over best-effort networks. In particular, video transmission is different from audio transmission since the dependency across successive video frames is much stronger owing to the employment of inter-frame motion-compensated coding. In this treatise, a network-adaptive source coding scheme is proposed for dynamically managing the dependency across packets, so that an attractive trade-off between compression efficiency and error resilience may be achieved. Open standards such as the ITU H.263 [51], H.264 [17] and the ISO/IEC MEPG-4 video codecs [15] were invoked in our proposed schemes.

To address the challenges involved in the design of wireless video transmissions and video streaming, in recent year the research efforts of the community have particularly been directed towards communications efficiency, error-resilience, low latency and scalability [4, 52, 53]. In video communications, postprocessing is also applied at the receiver side for the sake of error concealment and loss recovery. The achievable subjective quality was also improved by an adaptive deblocking filter in the context of the H.264/MPEG-4 video codec [54]. A range of techniques used for recovering the damaged video-frame areas based on the characteristics of image and video signals have been reviewed in [55]. More specifically, spatial-domain interpolation was used in [56] for recovering an impaired macroblock; transform-domain schemes were used for recovering the damage inflicted by partially received DCT coefficients, as presented in [57–60]. Temporal-domain schemes interpolate the missing information by exploiting the inherent temporal correlation in adjacent frames. Application examples include for example interpolated motion-compensation [61, 62], and state recovery [63]. More specifically, the conventional video compression standards employ an architecture, which we refer to as a single-state architecture, since they have for example a prediction loop assisted with a single state constituted

by previous decoded frame, which may lead to severe degradation of all subsequent frames, until the corresponding state is reinitialised in case of loss or corruption. In the state recovery system proposed by Apostolopoulos [63], the problem of having an incorrect state or that of encountering error propagation at the decoder is mitigated by encoding the video into multiple independently decodable streams, each having its own prediction process and state, such that if one stream is lost, the other streams can still be used for producing usable video sequence. Other schemes, such as the temporal smoothness method [64], the coding mode recovery scheme of [60, 64], and the Displaced Frame Difference (DFD) as well as Motion Vector (MV) recovery management of [65–68] have also resulted in substantial performance. These schemes can also be combined with layered coding, as suggested in [69, 70].

The development of a flexible and adaptive scheduling scheme that maintains a perceptually attractive video quality regardless of the channel quality encountered is one of the major contributions of this thesis. Recently, significant research interests have also been devoted to Burst-by-burst Adaptive Quadrature Amplitude Modulation (BbB-AQAM) transceivers [71, 72], where the transceiver reconfigures itself on a burst-by-burst basis, depending on the instantaneous perceived wireless channel quality. Modulation schemes of different robustness and different data throughput have also been investigated [73, 74]. The BbB-AQAM principles have also been applied to Joint Detection Code Division Multiple Access (JD-CDMA) [71, 75] and OFDM [76]. A range of other adaptive in video transmission schemes have been proposed for the sake of reducing the transmission delay and the effective of packet loss by Girod *et al.* [77, 78].

Video communication typically requires higher data transmission rates than other sources, such as for example audio or text. A variety of video communications schemes have been proposed for increasing the robustness and efficiency of communication [79–83]. Many of the recent proposals employ Rate Distortion (R-D) optimisation techniques for improving the achievable compression efficiency [84–86], as well as for increasing the error-resilience, when communicating over lossy networks [87, 88]. The goal of these optimisation algorithms is to jointly minimise the total video distortion imposed by both compression and channel-effects, subject to a given total bit-rate constraint. A specific example of recent work in this area is related to Intra/Inter-mode switching [89, 90], where Intra-frame coded macroblocks are transmitted according to the prevalent network conditions for mitigating the effects of error propagation across consecutive video frames. More specifically, an algorithm has been proposed in [90–92] for optimal Intra/Inter-mode switching, which relies on estimating the overall distortion imposed by quantisation, error propagation and error concealment.

A sophisticated channel coding module invoked in a robust video communication system may incorporate both Forward Error Correction (FEC) and automatic re-transmission on request (ARQ), provided that the ARQ-delay does not affect 'lip-synchronisation'. Missing or corrupted packets may be recovered at the receiver, as long as a sufficiently high fraction of packets is received without er-

rors [4, 93, 94]. In particular, Reed-Solomon (RS) codes are suitable for this application as a benefit of their convenient features [95, 96]. FEC is also widely used for providing unequal error protection (UEP), where the more vulnerable bits are protected by stronger FEC codes. Recent work has addressed the problem of how much redundancy should be added and distributed across different by prioritised data partitions [96–100]. In addition to FEC codes, data randomisation and interleaving have also been employed for providing enhanced protection [63,101,102]. ARQ techniques incorporate channel feedback and employ the retransmission of erroneous data [4,103–106]. More explicitly, ARQ systems use packet acknowledgments and time-outs for controlling, which particular packets should be retransmitted. Unlike FEC schemes, ARQ intrinsically adapts to the varying channel conditions and hence in many applications tends to be more efficient. However, in the context of real-time communication and low-latency streaming the latency introduced by ARQ is a major concern. Layered or scalable coding, combined with transmission prioritisation, is another effective approach devised for providing error resilience [61, 97, 107–109]. In a layered scheme, the source signal is encoded such that it generates more than one different significance groups or layers, with the base layer containing the most essential information required for media reconstruction at an acceptable quality, while the enhancement layer(s) contain information that may be invoked for reconstruction at an enhanced quality. At high packet loss rates, the more important, more strongly protected layers can still be recovered, while the less important layers might not. Commonly used layered techniques may be categorised into temporal scalability [110], spatial scalability [111, 112], Signal-to-Noise Ratio (SNR) scalability [15], data partitioning [17], or any combinations of these. Layered scalable coding has been widely employed for video streaming over best-effort networks, including the Internet and wireless networks [109, 113–116]. Different layers can be transmitted under the control of a built-in prioritisation mechanism without network support, such as the UEP scheme mentioned above, or using network architectures capable of providing a variety of different Quality of Service (QoS) [117–120]. A scheme designed for optimal Intra/Inter-mode selection has recently been proposed for scalable coding, in order to limit the inter-frame error propagation inflicted by packet losses [121]. Another scheme devised for adaptive bit rate allocation in the context of scalable coding was presented in [122]. Layered scalable coding has become part of various established video coding standards, such as for example the members of the MPEG [15, 20, 21] and H.263+ codec family [14].

Kondoz *et al.* [123] reported promising results on adopting the MPEG-4 codec for wireless applications by exploiting the rate control features of video transcoders and combined them with error resilient General Packet Radio Service (GPRS) type mobile access networks.

The employment of bidirectionally predicted pictures during the encoding process is capable of substantially improving the compression efficiency, since they are encoded using both past and future pictures as references. The efficiency of both forward and backward prediction was studied as early as 1985 by Musmann *et al.* [124]. Recent developments have been applied to the H.264/MPEG codecs,

amongst others by Girod [125] and Ghanbari [126]. In order to achieve even higher compression in video coding, Canagarajah and Bull in [127] proposed a fast Block Matching Motion Estimation (BMME) algorithm referred to as the simplex minimisation search (SMS), which was incorporated into various video coding standards such as the H.261, H.263, MPEG-1 and MPEG-2 for the sake of both single or multiple reference aided motion estimation.

A plethora of video coding techniques designed for the H.264 standard have been proposed also in the excellent special issues edited by Luthra, Sullivan and Wiegand [128]. At the time of writing there are numerous ongoing research initiatives with the objective of improving the attainable video transmission in wireless environments. Wenger in [129] discussed the transmission of H.264 encoded video over IP networks' while Stockhammer *et al.* in [130] studied the transmission of H.264 bitstreams over wireless environments. Specifically, the design of the H.264 codec specifies a video coding layer and a network adaptation layer, which facilitate the transmission of the bitstream in a network-friendly fashion. As for the wieless networking area, a recent publication of Nix *et al.* in [131] investigated the coexistence of 802.11g WLANs and high data rate Bluetooth enabled consumer electronic devices in indoor home and office environments.

Further important contributions in the area of joint source and channel coding entail the development of a scheme that offers the end to end joint optimisation of source coding and channel coding/modulation over wireless links, for example those by Kliewer and Thobaben [132], Fuja *et al.* [133].

1.4 Organisation of Thesis

The thesis is organised as follows:

- Chapter 2 provides a short introduction and gives an overview of the MPEG-4 video codec. This chapter will assist the reader in following our further elaborations in the forthcoming chapters.
- Chapter 3 has been divided into two parts, the first half provides an overview of the H.264 video codec, while the second part is constituted by a comparative study of the MPEG-4 and H.264 video codecs.
- Video compression often invokes Variable Length Coding (VLC). All the standard video codecs have adopted these techniques, since they are capable of dramatically reducing the bitarate. However, VLC techniques render the bistream vulnerable to transmission errors. In Chapter 4 we show the effects of transmission errors on the MPEG-4 codec, quantifying the sensitivity of the various bits.

- In Chapter 5, an H.263 and H.264 based video telephony system was proposed and investigated, using a novel Burst-by-Burst (BbB) adaptive Coded Modulation-Aided Joint Detection-Based Code-Division Multiple Access (JD-CDMA) video transceiver, which was capable of dropping its source coding rate and video quality under transceiver control as a function of the near-instantaneous channel quality.

- In Chapter 6, Several MPEG-4 video telephony systems will be evaluated. First of all we investigate an Iterative Parallel Interference Cancellation (PIC) aided CDMA MPEG-4 video phone scheme, followed by the study of a sophisticated space-time spreading assisted multi-carrier DS-CDMA system. Finally, we proposed a novel turbo-detected unequal error protection MPEG-4 video telephony using a serially concatenated convolutional outer code, trellis coded modulation based inner code and space-time coding. we also propose a simple packetisation scheme, where we partitioned the MPEG-4 bitstream into two bit-sensitivity classes of the Unequal Protection Scheme (UEP).

- Finally, Chapter 7 provides an overall comparative study of the various videophone systems and offers suggestions for future research.

Having provided an overview of the thesis, let us now commence our discourse with a brief overview of the MPEG-4 video coding scheme.

Chapter 2

MPEG-4 Video

2.1 Introduction

The"Moving Picture Experts Group" (MPEG) was established in 1988 [134], within the International Standard Organisation's (ISO) Steering Group (SG) 29, which was responsible for the encoding of moving pictures and audio. The MPEG group commenced the development of the MPEG-1 standard in 1988, released the MPEG-1 standard in 1993 and embarked on the standardisation of the MPEG-2 scheme in 1990 [135]. The MPEG-1 standard was mainly targeted at CD-ROM applications dedicated to recording video at bit rates of up to 1.5Mb/s [135, 136]. By contrast, the MPEG-2 standard was designed for substantially higher quality, namely for audiovisual applications such as today's home entertainment systems and digital broadcasting systems requiring video bit rates between 2Mb/s and 30Mb/s [137, 138].

The MPEG-4 standardisation process was initiated in 1994 - with the mandate of standardising algorithms for audio-visual coding in multimedia applications, while allowing for interactivity, and supporting high compression as well as universal accessibility and portability of both the audio and video content [139].

The MPEG-4 Visual standard was developed by the ISO/IEC 14496-2 [1], and its Version 1 was released in 1998, which additional tools and profiles were added in two amendments of the standard, culminating in Version 2 during late 2001. The operating bit rates targeted by the MPEG-4 video standard are between 5 and 64kbit/s in the context of mobile or Public Switched Telephone Network (PSTN) based video applications, spanning up to 4 Mbit/s for digital TV broadcast applications, and even to rates in excess of 100Mbit/s in High Definition TV (HDTV) studio applications [140].

[1] This is the project profile name for the International Organisation for Standardization/International Electrotechnical Commission. For example, the profile index 15444 for JPEG, 11172 for MPEG-1, 13818 for MPEG-2, 14496 for MPEG-4, etc.

The MPEG-4 video coding standard is capable of supporting all functionalities already provided by MPEG-1 and MPEG-2. The MPEG-4 Visual standard improves the popular MPEG-2 standard both in terms of the achievable compression efficiency, at a given visual quality, as well as in terms of the attainable flexibility that facilitates its employment in a wide range of applications. It achieves these substantial advances by making use of more advanced compression algorithms and by providing an extensive set of 'tools' for coding and manipulating digital media. The MPEG-4 Visual standard consists of a 'core' video encoder/decoder model that invoke a number of additional coding tools. The core model is based on the well-established hybrid DPCM/DCT coding algorithm and the basic function of the core is extended by 'tools' supporting an enhanced compression efficiency and reliable transmission. Furthermore, MPEG-4 facilitates an efficient and novel coded representation of the audio and video data that can be "content based", which is a concept to be highlighted in Section 2.3.

To elaborate a little further, the MPEG-4 video standard compresses the video signal with the aid of a compression tool box constituted by a set of encoding tools supporting several classes of functionalities. In short, the most important features supported by the MPEG-4 standard are *a high compression efficiency, content-based interactivity, and universal access*, which are summarised below [135, 141]:

- *Achieving high compression efficiency* has been a core feature of both MPEG-1 and MPEG-2. The storage and transmission of audio visual data requires a high compression efficiency, while reproducing a high-quality video sequence, hence enabling applications such as High Definition Television (HDTV) and Digital Video Disc (DVD) storage.

- *Content-based interactivity* represents video on an 'object-basis', rather than on a video 'frame-basis', which is one of the novel features offered by MPEG-4. The concept of content-based functionality will be elaborated on in more depth in Section 2.3.

- *Universal access* allows audiovisual information to be transmitted and accessed in various network environments such as mobile networks as well as wire line-based systems.

This chapter provides a rudimentary overview of the MPEG-4 video standard. Following the overview of the standard – its approach and features, in Section 2.2, the philosophy of the object oriented coding scheme will be discussed. This is followed by a discussion on the so-called profiles defined for coding of arbitrary-shaped objects and rectangular video frames in Section 2.3. Then the profiles defined for scalable coding of video objects are highlighted in Section 2.4 and subjective video quality measurement methods as well as our experimental results are discussed in Sections 2.5 and 2.6.

2.2 Overview of MPEG-4

2.2.1 MPEG-4 Profiles

The MPEG-4 standard aims for satisfying the requirements of various visual communications applications using a toolkit-based approach for encoding and decoding of visual information [15, 141]. Below we will describe some of the key features of the MPEG-4 video standard, which are superior in comparison to the previous video coding standards:

- The core compression tools are based on those of the ITU-T H.263 standard, which are more efficient than those of the MPEG-1 [142] and MPEG-2 [143] video compression schemes. Efficient compression of progressive and interlaced video sequences as well as optional additional tools were introduced for the sake of further improving the attainable compression efficiency.
- Coding of video objects, having both rectangular shapes and irregular-shapes object. This is a new concept in the context of standard-based video coding and enables the independent encoding of both foreground and background objects in a video scene.
- Support for error-resilient transmission over hostile networks. A range of error resilience tools were included in the MPEG-4 codec for the sake of assisting the decoder in recovering from transmission errors and for maintaining a successful video connection in an error-prone network. Furthermore, the scalable coding tools are capable of supporting flexible transmission at a range of desired coded bitrates.
- Coding of still image within the same framework as full-motion video sequences.
- Coding of animated visual objects, such as 2D and 3D computer-generated polygonal meshes, animated objects, etc.
- Coding for specialist applications, such as very high 'studio' quality video. In this application maintaining a high visual quality is more important than attaining a high compression.

Table 2.1 lists the MPEG-4 visual profiles invoked for coding video scenes. These profiles range from the so-called Simple Profile derived for the encoding of rectangular video frames through profiles designed for arbitrary-shaped and scalable object coding to profiles contrived for the encoding of studio-quality video.

2.2.2 MPEG-4 Features

Similarly to MPEG-1 [142] and MPEG-2 [143], the MPEG-4 specifications cover both the presentation and transmission of digital audio and video. However, in this thesis we only consider the specifics of

MPEG-4 Visual profile	Main features
□ Simple	Low-complexity coding of rectangular video frames
□ Advanced Simple	Coding rectangular frames with improved efficiency and support for interlaces video
□ Advanced Real-Time Simple	Coding rectangular frames for real-time streaming
□ Core	Basic coding of arbitrary-shaped video objects
□ Main	Feature-rich coding of video objects
□ Advanced Coding Efficiency	Highly efficient coding of video objects
□ N-Bit	Coding of video objects with sample resolutions other than 8 bits
□ Simple Scalable	Scalable coding of rectangular video frames
□ Fine Granular Scalability	Advanced scalable coding of rectangular frames
□ Core Scalable	Scalable coding of video objects
□ Scalable Texture	Scalable still texture with improved efficiency and object-based features
□ Advanced Scalable Texture	Scalable still texture with improved efficiency and object-based featured
□ Advanced Core	Combines features of Simple, Core and Advanced Scalable Texture Profiles
□ Simple Studio	Object-based coding of high quality video sequences
□ Core Studio	Object-based coding of high quality video with improved compression efficiency

Table 2.1: MPEG-4 Visual profiles for coding natural video [15, 141]

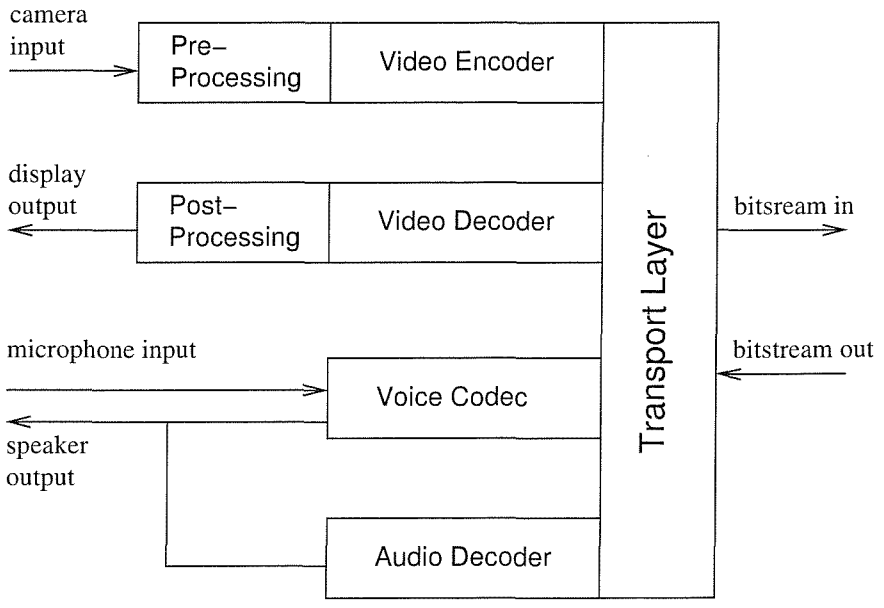


Figure 2.1: Simplified video telephone schematic

the video coding standard. The block diagram of a basic videophone scheme is shown in Figure 2.1. Let us now consider some of the associated operations in slightly more detail.

Pre-processing and video encoding

According to the MPEG ISO standard [144], the MPEG-4 video codec only supports YUV 4:2:0[2] Quarter Common Intermediate Format (QCIF) or Common Intermediate Format (CIF) video representations in the context of compression [23]. Here, the pre-processing block of Figure 2.1 performs all necessary processing of the camera input, in order to create the required 4:2:0 YUV QCIF or CIF based sequences. In order to encode the YUV video sequence into the MPEG-4 bitstream, the MPEG-4 encoder adopts the well-established motion compensation (MC) and Discrete Cosine Transform (DCT) based structure shown in Figure 2.2. The block by block discussion of the MPEG-4 encoder's components is postponed until Sections 2.3.1 and 2.3.2.

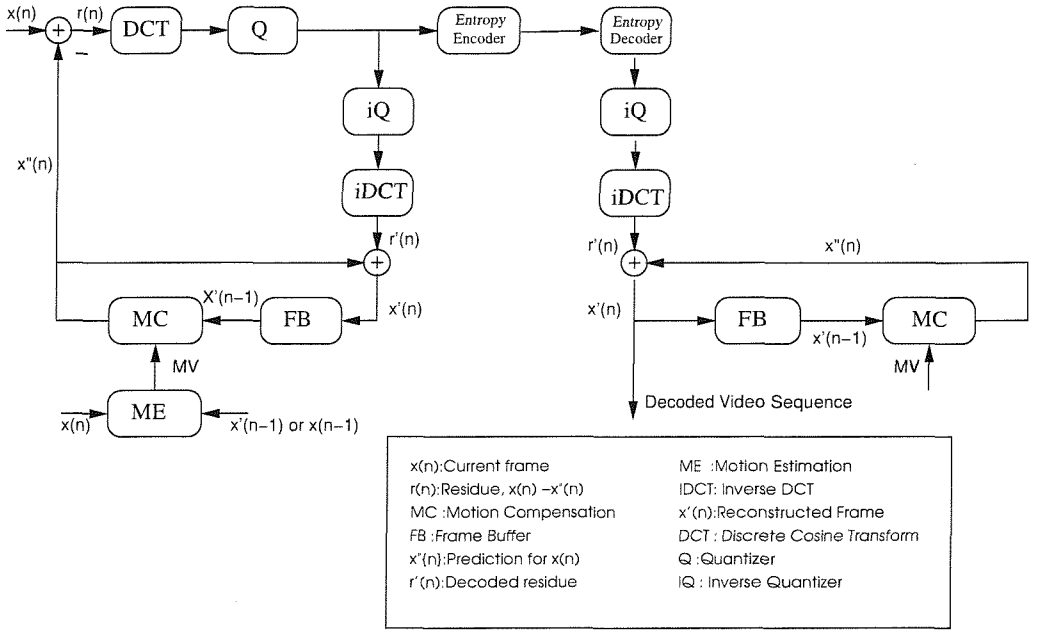


Figure 2.2: Block diagram of the MPEG-4 encoder and decoder.

Transport-layer

In MPEG-4, the transport of the video stream is divided into four layers, namely the Elementary Stream [146], the Synchronisation Layer, the 'Flexmuxed' [147] stream and the 'Transmux' [148] Stream. The MPEG-4 system's architecture and transport layer have been nicely documented in the literature, for example in [146, 149]. Below we briefly outline these transport layer characteristics:

[2]A colour encoding scheme [145] in which the luminance (Y) and the so-called color-difference signals U and V are represented separately. The human eye is less sensitive to colour variations than to intensity variations. Hence, the YUV format allows the encoding of the luminance (Y) information at full resolution and that of the colour-difference information at a lower resolution.

- The term *Elementary Streams* [146] refers to data that fully or partially contain the encoded representation of a single audio or video object, scene description information or control information.
- The *Synchronisation Layer (SL)* [150] adds the identification of the information sources, such as for example audio or video sources, as well as time stamps.
- *Flexmuxed Streams* [147] convey groups of elementary streams according to a specific set of common attributes, such as for example quality requirements.
- *Transmux Streams* [148] are constituted by streams transmitted over the network using transport protocols, such as the Real-Time Protocol (RTP) [151] used for transmission over the Internet.

Video Decoder and Post Processing

Figure 2.2 portrays the simplified block diagram of the MPEG-4 video decoder. Observe that the structure of the decoding process is identical to that of the encoder's local decoder. Motion compensation, which has been comparatively studied for example in [4], is the most important process both in the video encoder and decoder in terms of achieving a high video compression. Motion Compensation (MC) generates the Motion Vectors (MV) on the basis of identifying the most likely position within the previous video frame, where the current 8x8-pixel video block has originated from, as it moved along a certain motion trajectory in the consecutive video frames. This motion-compensation process involves allocating a certain search area in the previous frame and then sliding the current block over this search area in an effort to find the position of highest correlation. Once this position has been identified, the motion compensated prediction residual (MCPR) [4] is formed by subtracting the two blocks from each other.

2.2.3 MPEG-4 Object Based Orientation

One of the functionalities defined by the MPEG-4 standard is the audio-visual 'object' based processing, which forms the 'object-based' representation of the audio or video signal [137]. A video object can be exemplified by a person walking against the backdrop of mountains. Both the object and the background can be substituted by an other object or by another backdrop and as expected certain encoding techniques perform better for certain objects. This representation supports *'content-based interactivity'*, which will be discussed in more detail in section 2.3.

Figure 2.3 shows an example of a video frame extracted from a video scene, which consists of several objects namely text, an antenna, a mobile phone and the background scene. Again, in MPEG-4 based coding [144] these objects are referred to as 'Video Objects' (VO) . The syntax of this representation may be written as (VO1) - text, VO2 - the antenna, VO3 - mobile phone and VO4 - background.

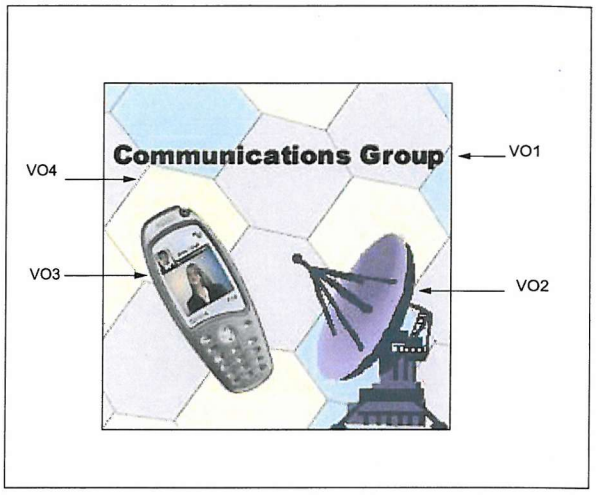


Figure 2.3: Original decoded video scene.

An MPEG-4 video scene may consist of one or more Video Objects (VO) . A video object (VO) is an area of the video scene that may occupy an arbitrarily-shaped region and may be present for an arbitrary length of time. An instance of a VO at a particular point in time is a video object plane (VOP). This definition encompasses the traditional approach of coding complete frames, in which each VOP is a single frame of video and a sequence of frames forms a VO. For example, Figure 2.4 shows a VO consisting of three rectangular VOPs, however in the MPEG-4 video standard, the introduction of the arbitrarily shaped VO concept allows for more flexibility. Figure 2.5 shows a VO that consists of three irregular-shaped VOPs, each one present within a frame and each encoded separately, hence leading to the concept of object-based coding, which will be discussed in more detail in Section 2.3. The VO can be in binary shapes as VO1, VO2, VO3 in Figure 2.6 or in rectangular shape as VO4 in Figure 2.6, which is equivalent to the dimension of the entire video frame's size. For example, if a QCIF video format is used, the dimension would be 176×144 pixels.

In Table 2.2 we summarised some of the important nomenclature, which will be often used, when referring to the MPEG4 video coding syntax. Let us now consider content-based interactivity in the context of MPEG-4 based coding in more detail.

2.3 MPEG-4 : Content-Based Interactivity

'Content-Based Interactivity' attempts to encode an image scene in a way that it will allow the separate decoding and reconstruction of the various objects as well as facilitating the manipulation of the original scene with the aid of simple operations carried out in the form of its bitstream representation [137, 152]. As mentioned in Table 2.2, the MPEG-4 video coding standard provides an

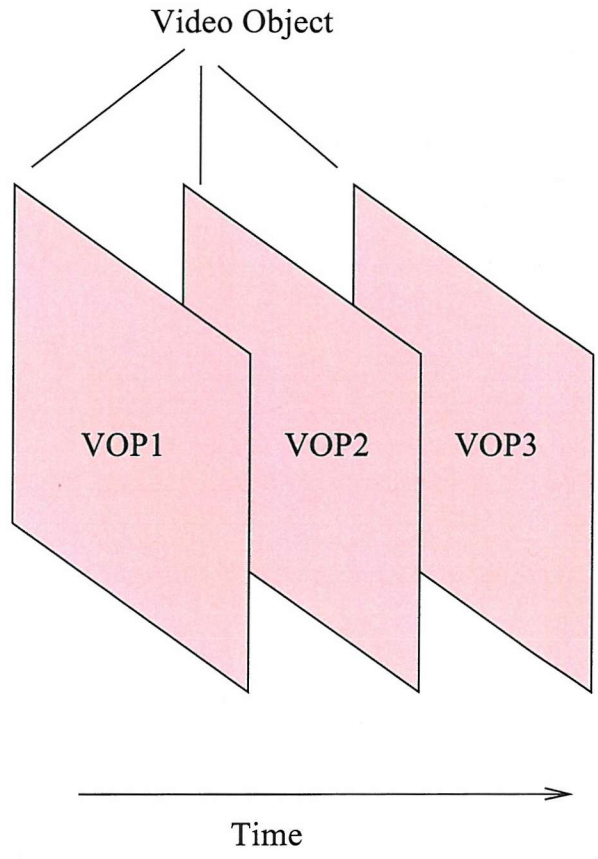


Figure 2.4: VOPs and VO (rectangular)

"object-layered" bitstream referred to as Video Object Layer (VOL) for supporting this function. Hence, at the encoder the bitstream will be object-layered and the shape, the grade of transparency of each object as well as the spatial coordinates and additional parameters describing object scaling, rotation, etc. are described by the bitstream of each video object layer (VOL) [137]. The received bitstream including all the information bits is decoded and reconstructed by displaying the objects in their original size and at the original location, as depicted in Figure 2.7. Alternatively, it is possible to manipulate the image sequence according to the user's preference, allowing the scaling shifting and other transformations of the objects, as seen in Figure 2.8.

As illustrated in Figure 2.8, the mobile phone video object was not decoded, the satellite ground station was decoded and displayed using scaling or rotation. Additionally, a new mobile videophone object defined by the user was included, which did not belong to the original scene. Since the bitstream of the sequence is organised in an "Object Layered" form, the object manipulation is performed at the bitstream level, by adding or deleting the appropriate object bitstreams [153, 154].

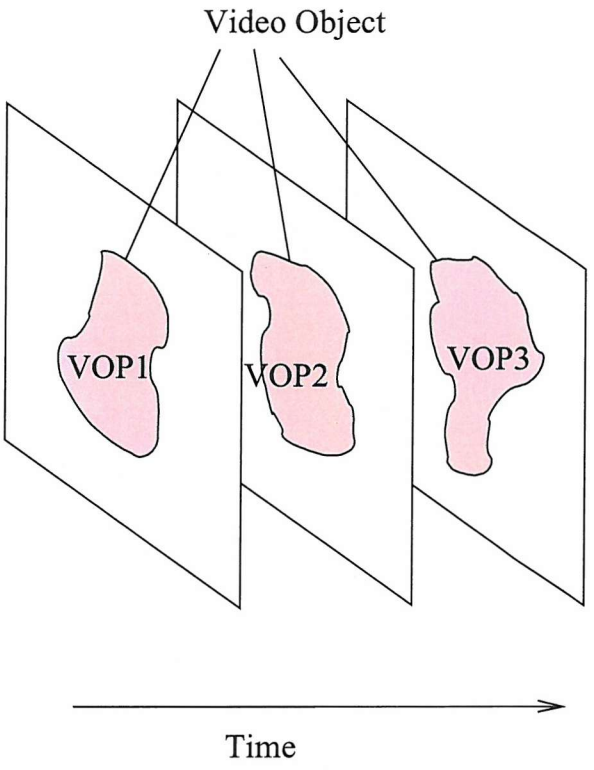


Figure 2.5: VOPs and VO (arbitrary shape)

2.3.1 Video Object Plane Based Encoding

Before we may commence the encoding of an object, it must be sampled. Most objects are sampled at regular time intervals corresponding to the frame-scanning rate, and each sample corresponding to the object's spatial representation at an instant in time is known as a *video object plane* (VOP). Hence each object in the scene is represented by a series of VOPs. In more familiar terms, a camera views a scene and captures the information by sampling the scene (by either canning, or shuttering and scanning). The camera provides its output as a sequence of frames or, in MPEG-4 terminology, the texture part of a sequence of VOPs. A VOP contains texture data and either rectangular shape information or more complex shape data associated with the object. VOPs, like frames in earlier versions of the MPEG codec family [20, 21], may be encoded using intra-frame coding or by using motion compensation.

The MPEG-4 standard introduces the concept of Video Object Planes (VOPs) for supporting the employment of content-based interactive functionalities [135]. The associated concept is illustrated in Figure 2.9. The content of each video input frame is segmented into a number of arbitrarily shaped image regions - i.e. into Video Objects (VO) and each VO is sampled in the time domain by extracting the corresponding area of the consecutive video frames. Each time domain sample of a VO

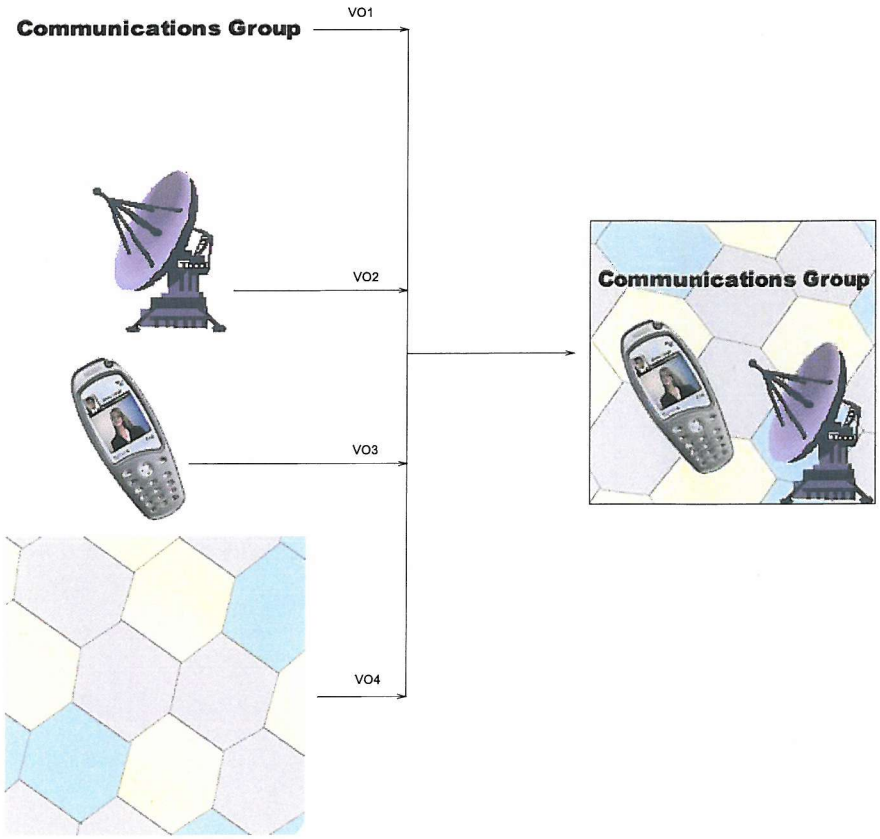


Figure 2.6: Object oriented coding functionality in MPEG-4.

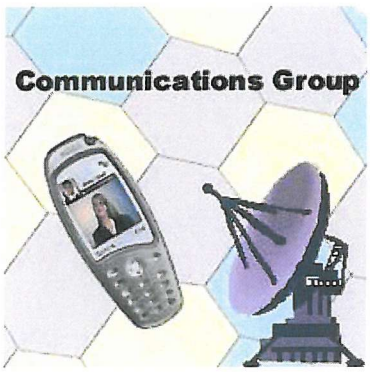


Figure 2.7: Original decoded video scene.

	Name	Description
□	Visual Object Sequence (VS)	The complete MPEG-4 scene, which may contain 2-D or 3-D natural as well as computer generated objects.
□	Video Object (VO)	A 2D video object. In the simplest case this may be a rectangular frame, or an arbitrarily shaped element of the video frame corresponding to an object or background of the scene.
□	Video Object Layer (VOL)	Every video object can be encoded in a scalable fashion i.e. at different bitrates, using a multi-layer representation constituted by the so-called base-layer and enhancement layer. Alternatively, it may be encoded in a non-scalable, i.e. fixed-bitrate form using a base-layer, but no enhancement layer, depending on the application. These layers are referred to as Video Object Layers (VOL). The VOL facilitates scalable coding, where the video object can be encoded using both spatial and/or temporal scalability.
□	Video Object Plane (VOP)	A VOP is a time-domain sample of a video object. The VOPs can be encoded independently of each other, i.e. using intra-frame coding or inter-frame as well as bidirectional coding techniques employing motion compensation.

Table 2.2: Different MPEG-4 object-oriented representations of various video scenes

Figure 2.8: Decoded video scene according to the user's preference. The content-based approach adopted by the MPEG-4 video coding standard allows flexible decoding, representation and manipulation of the video objects in a scene, where for example different-resolution video decoding is facilitated.

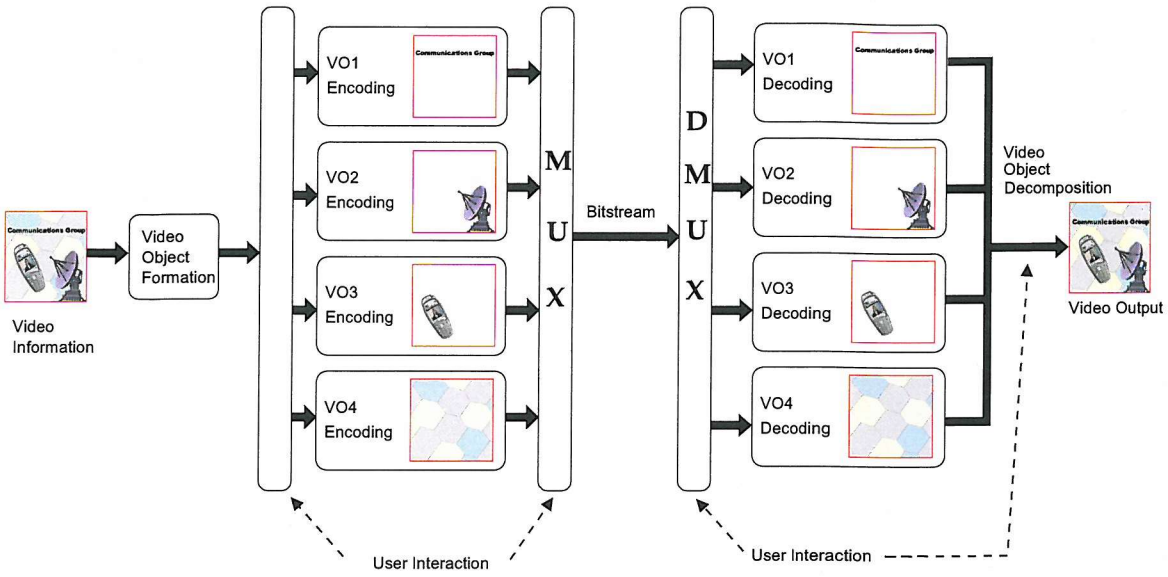


Figure 2.9: The content-based approach adopted by the MPEG-4 video coding standard allows flexible decoding, representation and manipulation of video objects in a scene.

which corresponds to its image in consecutive video frame constitutes a VOP. The shape and location of each VO may vary from frame to frame, which can be visualised by considering the example shown in Figure 2.10. More explicitly, Figure 2.10 shows five consecutive frames of a particular video object, namely that of the paraboloid antenna, which is rotating from the left to the right during the time interval specified by the five consecutive frames spanning the interval of Frame 1 to Frame 5. Hence, in this case the paraboloid antenna constitutes a VO, while the successive frames portraying the VO constitute a VOP.

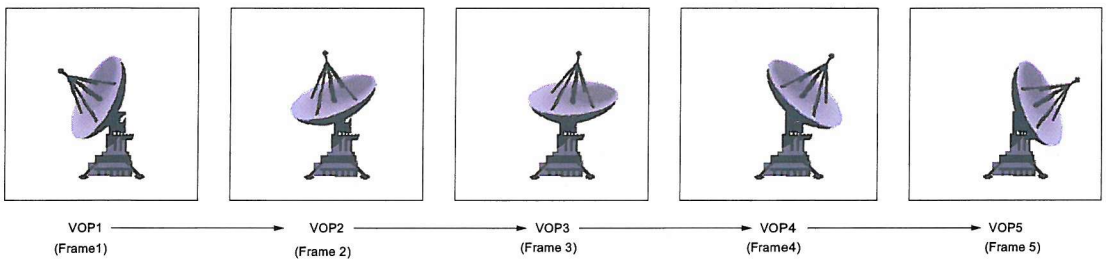


Figure 2.10: The encoding of image sequences using MEPG-4 VOPs enables the employment of content-based functionalities, where the VOP contains five consecutive time domain samples of a VO extracted from five consecutive frames representing a video scene.

In contrast to the MPEG-1 [142] and MPEG-2 [143] standards, the VOP used in MPEG-4 is thus no longer considered to be a rectangular region. Again, the VOP extracted from a video scene contains motion parameters, shape information and texture data. These are encoded using an arrangement

similar to a macroblock coding scheme that is reminiscent of the corresponding schemes used the MPEG-1 and MPEG-2 standards, as well as in the ITU H.263 coding standard [22].

2.3.2 Motion and Texture Encoding

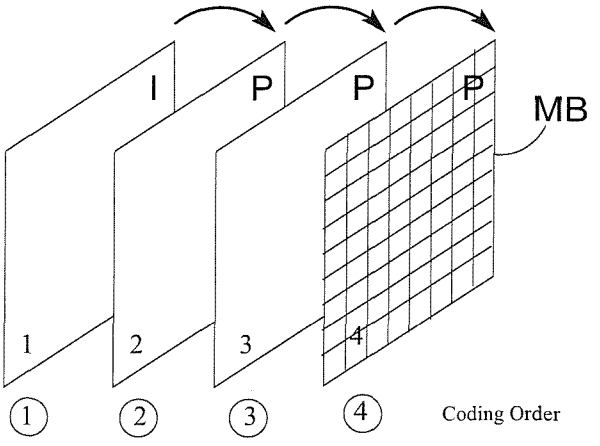


Figure 2.11: I-frames (I-VOP) and P-frame (P-VOP) in a video sequence. The P frames are encoded by using motion compensated prediction based on the nearest previous VOP.

As mentioned earlier, the MPEG-4 video encoding algorithm has a similar structure to that of the well-established MPEG-1/2 and H.263 coding algorithms.

Sikora and Schafer argued in [135] that the MPEG-4 coding algorithm encodes the first VOP in the intra-frame VOP coding mode (I-VOP), while each subsequent frame is encoded using inter-frame coded or predicted VOPs (P-VOP) and only data which accrues from the previous coded VOP frame is used for prediction. As can be seen from Figure 2.11, in this particular case the Video Object is treated in a rectangular form. The first frame is coded as an I-VOP frame, while the second frame-which is the first P-VOP frame-is encoded using the previous I-VOP frame as a reference for temporal coding. Each subsequent P-VOP frame uses the previous P-VOP frame as its reference. Let us now consider how motion and texture encoding is carried out in MPEG-4 [135]. Both the motion and texture encoding of a VOP sequence is block-based. A block in video coding is typically defined as a rectangular array of 8×8 pixels [139]. Since the chrominance of the video signal components is typically sampled at a spatial frequency, which is a factor of two lower, than the spatial sampling frequency of the luminance (Y), each chrominance (C) block carries the colour-difference related information corresponding to four luminance (Y) blocks. The set of these six 8×8-pixel blocks (4Y and 2C) is referred to as a macroblock (MB), as shown in Figures 2.12 and 2.13. A MB is treated as a single encoded unit during the encoding process [155]. Additionally, the MPEG-4 scheme uses Y,U,V coding and a 4:2:0 structure of colour information [145]. This means that the luminance is encoded for every

pixel, but the colour difference information is filtered and decimated to half the luminance resolution, both horizontally and vertically. Thus an image area represented by a block of 16 × 16 luminance pixels requires only 8 × 8 values for U and 8 × 8 values for V. Since the standard uses 8 × 8-pixel blocks for the DCT [144], the macroblock consists of four blocks of luminance samples (Y) and one block U as well as V samples. Figure 2.13 shows the macroblock encoding order for four luminance (Y) and two chronimance (U, V) blocks whereas Figure 2.12 shows the spatial relationship between the luminance and colour difference samples in YUV format.

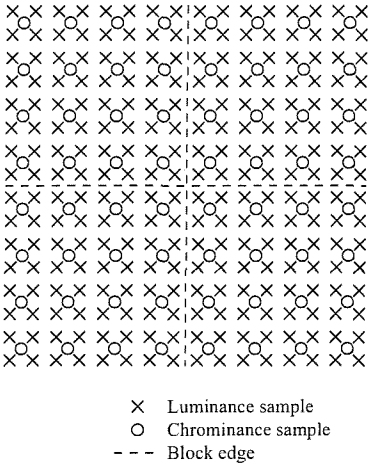


Figure 2.12: Positioning of luminance and chrominance samples in a macroblock containing four 8 × 8-pixel luminance blocks having a total area of 16 × 16 pixels in a video frame. Both colour-difference signals separating the chrominance samples are processed at half the spatial resolution.

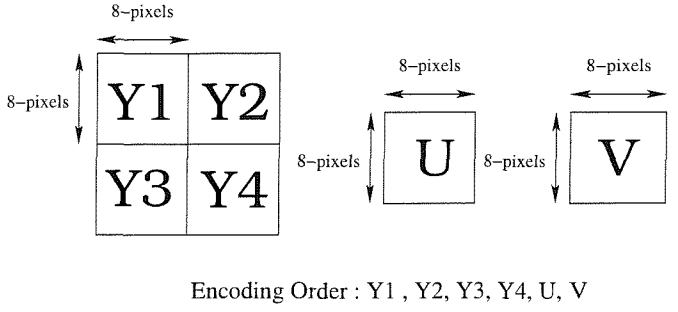


Figure 2.13: Encoding order of blocks in a macroblock.

The block diagram of an idealized video encoder and decoder has been shown earlier in Section 2.2 in the context of Figure 2.2. In this section, we will discuss each individual block of the MPEG-4 video codec in more depth. The MPEG-4 video encoder and decoder are shown in Figure 2.14 and 2.15, respectively. The first frame in a video sequence (I-VOP) is encoded in the intra-frame coded mode without reference to any past or future frames. As seen in Figure 2.2, at the encoder the DCT

is applied to each 8×8-pixel luminance and chrominance block. Then each of the 64 DCT coefficients is quantized (Q) in the block. After quantization, the lowest-frequency DCT coefficient, namely the DC coefficient is treated differently from the remaining coefficients, which are also often referred to as the 'alternating current' (AC) coefficients. The DC coefficient corresponds to the average luminance intensity of the block considered and it is encoded by using a differential DC component encoding method, employing the DC value of the previous frame as reference, when predicting and encoding the current one. The nonzero quantized values of the remaining DCT coefficients and their locations are 'zig-zag'-scanned and run-length or entropy-coded by means of variable-length code (VLC) tables similarly to the techniques shown from the MPEG-1, MPEG-2 and H.263 codecs [144]. More DCT algorithm will be discussed in Section 3.4.3. Readers interested in the details zig-zag scanning are referred to [4], for example.

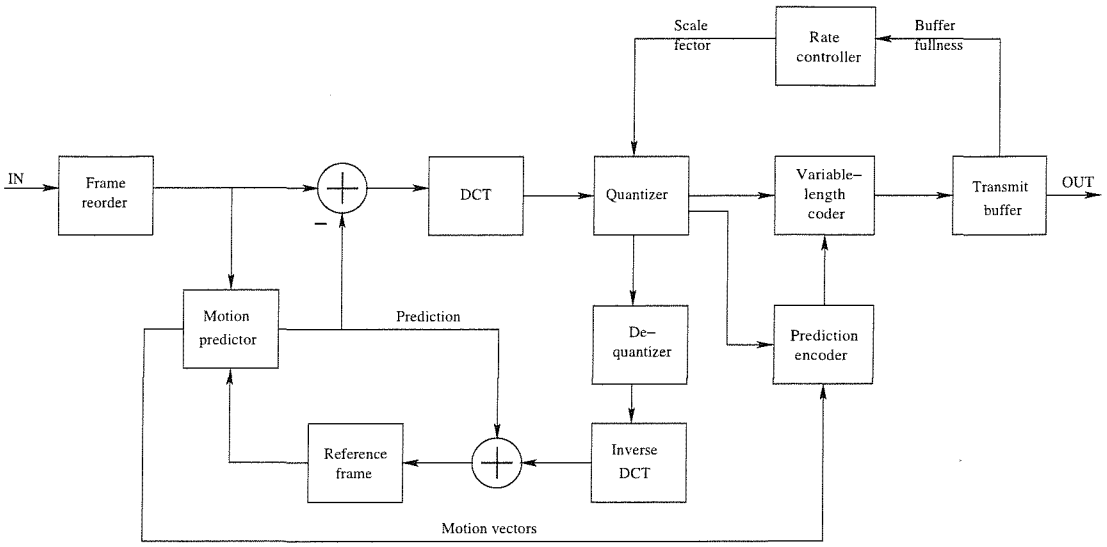


Figure 2.14: Block diagram of the MPEG-4 encoder.

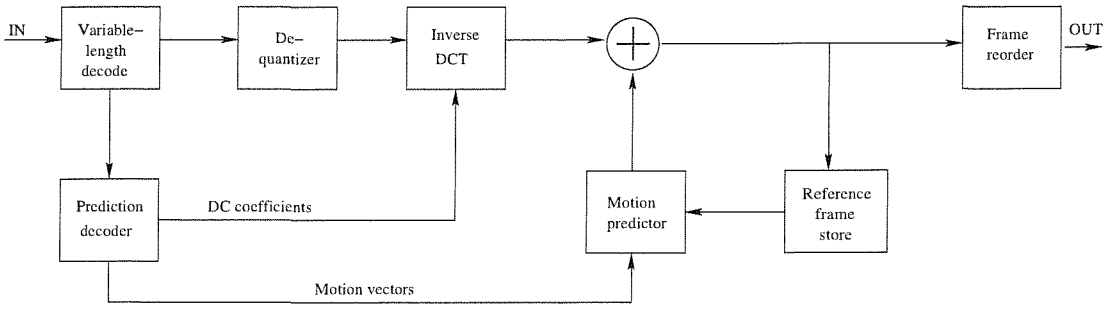


Figure 2.15: Block diagram of the MPEG-4 decoder.

When considering P-VOP coding, the previous I- or P-VOP frame, namely frame $n-1$ is stored in the reconstructed Frame Buffer (FB) of both the encoder and decoder for frame reference. Motion

Compensation (MC) is performed on a macroblock basis. Hence only one Motion Vector (MV) is estimated for the frame VOP-n for a particular macroblock to be encoded. These motion vectors are encoded and transmitted to the receiver. The motion-compensated prediction error $r(n)$ seen in Figure 2.2 is calculated by subtracting each pixel in a macroblock from its motion-shifted counterpart in the previous VOP frame, namely in VOP-$(n-1)$. Then an 8×8-dimensional DCT is applied to each of the 8×8 blocks contained in the macroblock, followed first by quantisation of the DCT coefficients and then by run-length coding and entropy coding, both of which constitute variable-length coding (VLC) techniques.

The decoder of Figure 2.15 uses the 'inverse' routine of the encoder for reproducing a macroblock of the Nth VOP frame at the receiver. After decoding the variable-length words contained in the video decoder's buffer, the pixel values of the motion prediction error $r(n)$ are reconstructed with the aid of the inverse quantizer (iQ) and inverse DCT blocks of Figure 2.2. The motion-compensated pixels of the previous VOP frame, namely those of VOP-$(n-1)$ are contained in the VOP frame buffer of the decoder, which are added to the motion prediction error $r(n)$ after appropriately positioning them according to the MVs, as seen in Figure 2.2 for the sake of recovering the particular macroblock of frame n.

2.3.3 Shape Coding

A VO can be rectangular or of arbitrary shape. For a rectangular VOP, the encoding process is similar to that of the MPEG-1 [142] and MPEG-2 [143] standard. However, if a VO is of arbitrary shape, a further coding step is necessitated prior to motion and texture coding, as illustrated in Figure 2.16 [156]. Specifically, in the MPEG-4 visual standard two types of shape information are considered as inherent characteristics of a VO, binary and gray scale shape information [157].

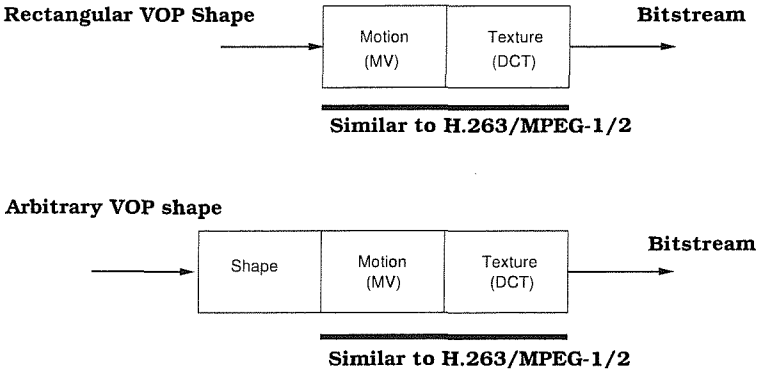


Figure 2.16: MPEG-4 shape coding structures.

2.3.3.1 VOP Shape Encoding

In the MPEG-4 video compression algorithm, the shape of every VOP is coded along with its other parameters, such as its texture and motion vectors. A binary alpha plane defines which pixels within the boundary box belong to the VO at a given instant of time. [158]. The VOP shape information or in this case it is called binary alpha plane is most commonly represented by a matrix having the same size of the VOP, where each element of the matrix may assume one of two possible values, namely 255 or 0, depending on whether the pixel is inside or outside the video object [157]. If the corresponding pixel belongs to the object, then the element is set to 255, otherwise it is set to 0. This matrix is referred to as a *binary mask* or as a *bitmap*. Figure 2.17 shows the binary alpha plane of the "Miss America" and "Akiyo" VOP.

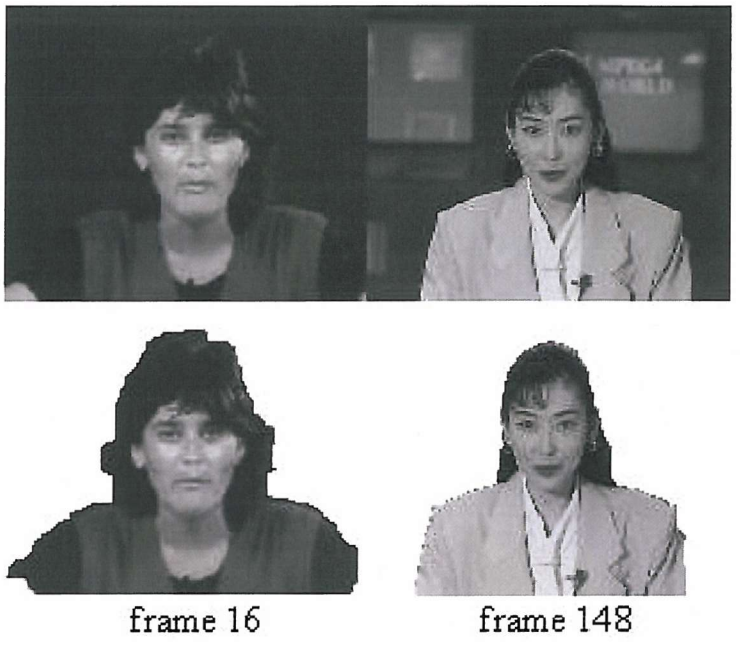


Figure 2.17: MPEG-4 binary plane of the "Miss America" and "Akiyo" frame.

Before encoding, the binary alpha plane is then partitioned into binary alpha blocks (BAB) of size 16 × 16-pixels. Each BAB is encoded separately. It is not surprising that a BAB may contain identical values, which are either all 0, in which case the BAB is referred to as a transparent block or 255, when the BAB is said to be an opaque block. The main MPEG-4 tools used for encoding BABs are the Context based Arithmetic Encoding (CAE) algorithm and the motion compensation scheme[3] [157]. Inter-frame CAE (InterCAE) and intra-frame CAE (Intra CAE) are the two variants

[3]Any video sequence can be viewed as a set of consecutive snapshots of still images of a scene. Therefore, the consecutive snapshots are correlated. It is this form of predictability or redundancy that the motion prediction mechanism is exploiting. In a basic form, we can simply use the previous frame for predicting the current frame.

of the CAE algorithm used in conjunction with P- or I-VOPs, respectively. The InterCAE scenario involves motion vectors, which are based on finding and encoding the best-matching position of the previous VOP frame, whereas the other one is used without motion compensation and is referred to as IntraCAE. A BAB of the current VOP may be encoded in one of the seven possible modes [159]:

1. If the entire BAB is flagged transparent, no shape encoding is necessary at all and hence texture information is not encoded for this BAB.
2. If the entire BAB is flagged opaque, no shape encoding is necessary at all, but the texture information is encoded for the VOP.
3. The BAB is encoded using IntraCAE without any reference to previous frames and motion compensation.
4. The block is not updated with respect to the same block of the previous frame, if we have zero Motion Vector Difference (MVD) for the block concerned between the previous and current frame.
5. Even if the MVD is zero, the content of the block may be updated. In this case, InterCAE is used for encoding the block update.
6. The MVD is non-zero and no update is necessary, thus neither the texture nor the shape of the block is encoded.
7. The MVD is non-zero and the block has to be updated. In this case, InterCAE is used for encoding both the texture and the shape of the block.

Modes 1 and 2 require no shape coding. For mode 3, shape is encoded using IntraCAE. For modes 4-7, motion estimation and compensation are employed. The motion vector difference (MVD) is the difference between the shape motion vector (MV) and its predicted value (MVP). This predicted value is estimated from either the neighbouring shape motion vectors or from the co-located texture motion vectors. When the mode indicates that no update is required, then the MV is simply used to copy an appropriately displaced 16×16-pixel block from the reference binary alpha plane to the current BAB. If, however, the mode indicates that an update is required, then the update is coded using InterCAE.

2.3.3.2 Gray Scale Shape Coding

Instead of having only 0 and 255 as possible values for the shape encoding matrix, the shape encoding may assume a range of values spanning from 0 to 255, which represent the degree of transparency for each pixel, where 0 corresponds to a transparent pixel and 255 represents an opaque pixel. As regards

to the values between 0 and 255, the smaller the value, the more transparent the pixel. Similarly, the larger the value, the more opaque the pixel [158]. This scenario is referred to as gray-scale shape encoding, rather than binary shape encoding. These values may also be stored in a matrix form for representing the shape of VOP. The Gray-scale shape information is also encoded using a block based DCT similar to the conventional approach used in texture coding.

2.4 Scalability of Video Objects

In terms of scalability of the text, images and video to be encoded, the MPEG-4 standard provides a procedure for the supporting complexity-based, spatial, temporal and quality scalability [159] which is the most frequently used parlance for indicating that the MPEG-4 codec may be configured in a plethora of different coding modes for the sake of striking different trade-offs in terms of the achievable implementation complexity, spatial and temporal resolution, video quality, etc. More specifically:

- Complexity-based scalability of the encoder and decoder facilitates the encoding of images or video at different levels of algorithmic complexity, where the complexity affects the quality of the reconstructed object.
- Spatial scalability makes it possible for the bitstream to be decoded in subsets so that the spatial resolution of the objects would be improved upon decoding each consecutive subset. A maximum of three specific scalability levels are supported for video objects and 11 different levels for still images as well as for text.
- The philosophy of Temporal scalability is similar to that of spatial scalability, except that the video is displayed at a reduced temporal resolution, rather than reduced spatial resolution, for example at lower frame-rate.
- Quality-motivated scalability implies that a bitstream could be separated into several layers, corresponding to different bitrates. When the layers are decoded, the quality is determined by the number of layers that was used.

Important considerations for video coding schemes to be used within future wireless networks are the achievable compression efficiency, the attainable robustness against packet loss, the ability to adapt, to different available bandwidths, different amounts of memory and computational power for different mobile clients, etc. Scalable video coding schemes have been proposed in the literature [113], which are capable of producing bit-streams decodable at different bit-rates, requiring different computational power and channel bit-rate. In this work we make use of a scalable video transmission scheme, which has been previously developed at Erlangen University in Germany, based on a spatio-temporal resolution pyramid [113].

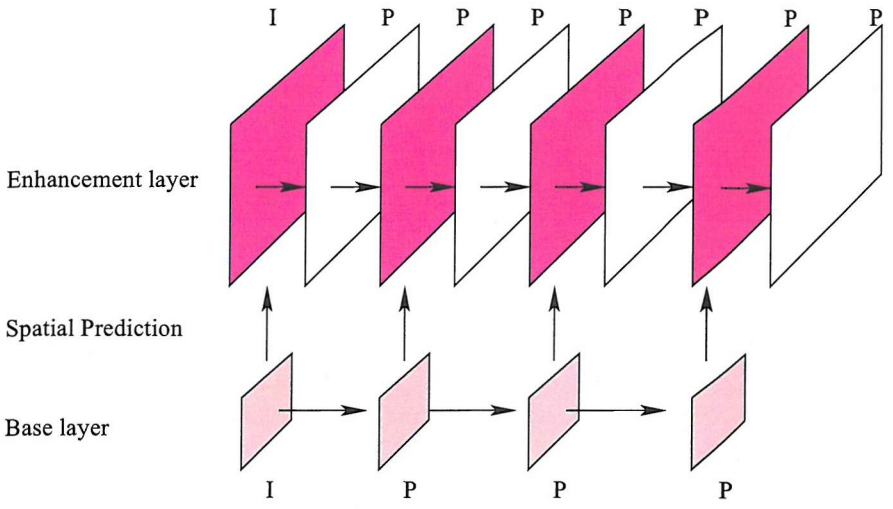


Figure 2.18: Example of a possible configuration of the spatio-temporal resolution pyramid of the scalable video codec. One base layer relying on an intra-frame and three interpolated inter coded frames are shown, supplemented by an enhancement layer having twice the spatial and temporal resolution. The horizontal and vertical arrows denote temporal and spatial prediction, respectively.

An example associated with two layers is shown in Figure 2.18. The enhancement layers are encoded by a motion-compensated hybrid codec, where the DCT has been replaced by lattice vector quantisation of the MCER. This approach leads to a coding efficiency attained by the layered coding scheme, which is comparable to that of single-layer codecs []. Since the ability to decode an enhancement layer depends on the reception of the base layer and lower enhancement layers, an efficient transmission scheme is expected to ensure that these layers are transmitted such that the associated packet loss is kept as low as possible even for high overall packet loss rates. In addition to the ability to adapt to different clients we can also ensure a sufficiently graceful degradation of the associated video quality in case of packet loss in this scenario.

As mentioned earlier, MPEG-4 provides both spatial and temporal scalability at the object level [15, 159]. In both of these scenarios this technique is invoked for the sake of generating a *base layer*, representing the lowest quality to be supported by the bitstream, and one or more *enhancement layers*. These layers may all be produced in a single encoding step. The scaling can be implemented in two different ways. When there are known bandwidth limitations, the different-rate versions of the bitstream may be used that include only the base layer, or the base layer plus lower-order enhancement layers. Alternatively, all layers may be transmitted and the scaling decision may be left for the decoder's discission. If the display device at the receiver side has a low resolution, or if the available computational resources are insufficient capability, the enhancement layers may be ignored.

Figure 2.19 shows the concept of an encoder exploiting spatial scalability; in this case at just two levels. The input VOP is down-converted to a lower resolution, resulting in the base layer. This layer is encoded and then a decoder reconstructs the base-layer VOP, as it will appear at the decoder's display. This VOP is then up-converted to the same resolution as the input, and a subtraction operation generates the difference in comparison to the original image. These are separately encoded in an enhancement-layer encoder. Note that each stream of the encoded VOPs forms a video object layer. The base-layer VOL uses both Intra- and Inter-frame coding, but the enhancement layer uses only predictive coding. The base-layer VOPs are used as references, as shown in Figure 2.18

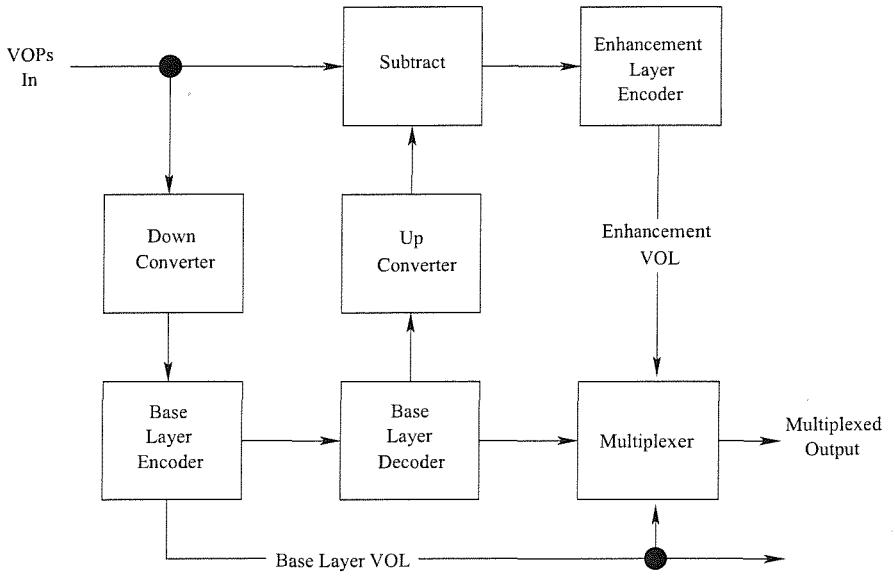


Figure 2.19: Spatially scalable encoder for a single enhancement layer.

2.5 Video Quality Measure

In this section, the objective video quality measure used during our investigations of the various wireless video transceivers is defined. Quantifying the video quality is a challenging task, because numerous factors may affect the results. Video quality is inherently *subjective* and our human perception is influenced by many factors.

2.5.1 Subjective Video Quality Evaluation

Several test procedures designed for subjective video quality evaluation were defined in the ITU-R Recommendation BT.500-11 [160]. A commonly-used procedure outlined in the standard is the so-called Double Stimulus Continuous Quality Scale (DSCQS) method, in which an assessor is presented

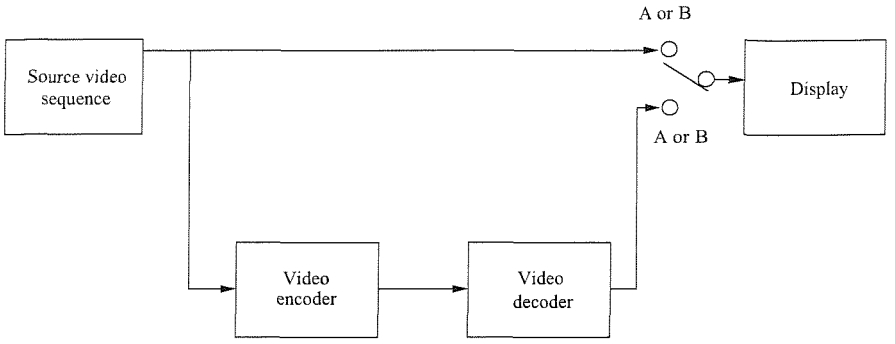


Figure 2.20: DSCQS testing system

with a pair of images or short video sequences A and B, one after the other, and is asked to assign both A and B a quality score on a continuous scale having five intervals ranging from 'Excellent' to 'Bad'. In a typical test session, the assessor is shown a series of pairs of sequences and is asked to grade each pair. Within each pair of sequences, one sequence is an unimpaired 'reference' sequence and the other is the same sequence, modified by a system or process under test. Figure 2.20 shows an experimental set up for the testing of a video codec, where the original sequence is compared to the same sequence after encoding and decoding. Additionally, the order in which the sequence 'A' and 'B' are presented randomly. The evaluation of subjective measures, such as the DSCQS measure is time consuming and expensive. Hence, contriving an objective video quality measure, which is capable of reliably predicting the subjective quality is desirable.

2.5.2 Objective Video Quality

The designers and developers of video compression and processing systems rely heavily on objective quality measures. The most widely used measure is the Peak Signal to Noise Ratio (PSNR) . The PSNR is measured on a logarithmic scale and depends on the normalized mean squared error (MSE) between the original and the reconstructed as well as potentially channel-impaired image or video frame, relative to $(2^n - 1)^2$, namely normalized by the square of the highest possible pixel value in the image, where n is the number of bits per image sample, yielding:

$$PSNR = 10log_{10}\frac{(2^n - 1)^2}{MSE}. \tag{2.1}$$

The PSNR may be conveniently calculated. However, its evaluation requires the availability of the unimpaired original image or video signal for comparison, witch may not be available.

Unfortunately the PSNR defined above does not always constitute a reliable image quality measure. For example, if the received image is shifted by one pixel compared to the original image, the human eye would hardly notice any difference, while the PSNR objective measures would indicate a more

substantial degradation in quality.

Nonetheless, owing to its appealing implicity, in this thesis, the PSNR will be used as the predominant image quality measure.

2.6 Effective of Coding Parameters

The previous sections described the MPEG-4 encoding and decoding process with the aid of block diagrams shown in Figures 2.14 and 2.15. In video compression, some of the encoding parameters seen in the context of Figures 2.14 and 2.15 may directly affect the resultant reconstructed video quality. Therefore in this section, we will briefly demonstrate how these parameters may affect the encoded/decoded video quality.

The MPEG-4 source code used in our simulations was a modified version of the software implementation provided by the Mobile Multimedia Systems (MoMuSys) Pan-European project. Simulations were carried out for the sake of characterising the achievable performance of the MEPG-4 codec. Most of our simulations used the popular "Akiyo", "Miss America" or "Foreman" (144 × 176)-pixel resolution QCIF video sequence at a transmission frame scanning rate of 10 or 30 frames/s. Additionally, the length of the video sequence used in our simulations was 100 frames and apart from the first frame, no Intra-Frame coded update was used, unless otherwise stated. Our results will be shown in graphical form in Figure 2.24 to Figure 2.28.

Our investigations commenced by demonstrating the effect of the quantiser index employed in the MPEG-4 encoder. In video compression, the quantisation operation is typically constituted by a 'forward quantiser' in the encoder and an 'inverse quantiser' in the decoder. A critical parameter is the quantiser step size. If the step size is large, typically a highly compressed bit stream is generated. However, the reconstructed values provide a crude approximation of the original signal. If the step size is small, the reconstructed values match the original signal more closely at the cost of a reduced compression efficiency. The effect of different quantiser step sizes is demonstrated in Figure 2.21 in terms of the video PSNR, while the associated subjective video quality is illustrated in Figure 2.22.

Let us now consider the effect of different bit-rates, although the effects of the video bit-rate and quantiser step size are highly correlated. In order to achieve a given target bit-rate, the encoded video frame buffer is used for adjusting the quantiser step size. More explicitly, the encoder produces a variable-rate stream and the buffer may be emptied at a near-constant transmission rate. If the encoder produces bits at a low rate, the buffer may become empty. By contrast, if the encoded bitrate is too high, the buffer may overflow and the data may become irretrievably lost. For the sake of avoiding these two problems, we typically feed a measure of buffer fullness to the rate controller, which reacts by appropriately adjusting the quantisation step size of the DCT coefficients of the video

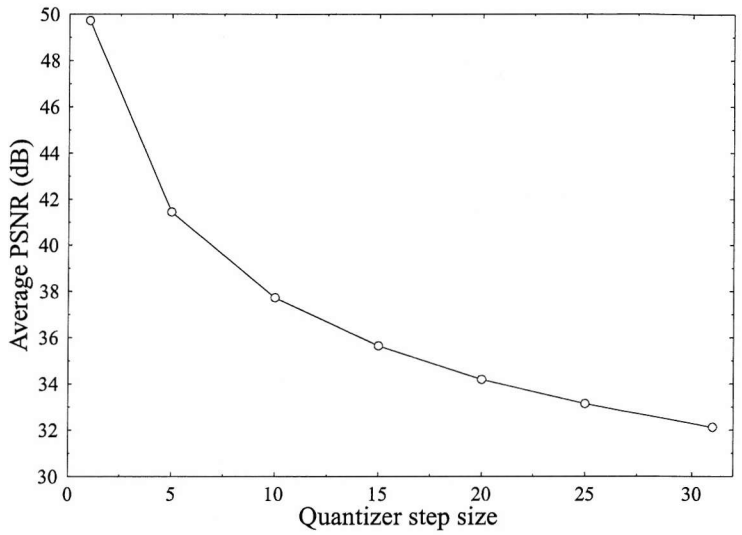


Figure 2.21: MPEG-4 average video quality (PSNR) versus quantiser step size ranging from 1 to 31 using the (144 × 176)-pixel QCIF "Miss America" sequence.

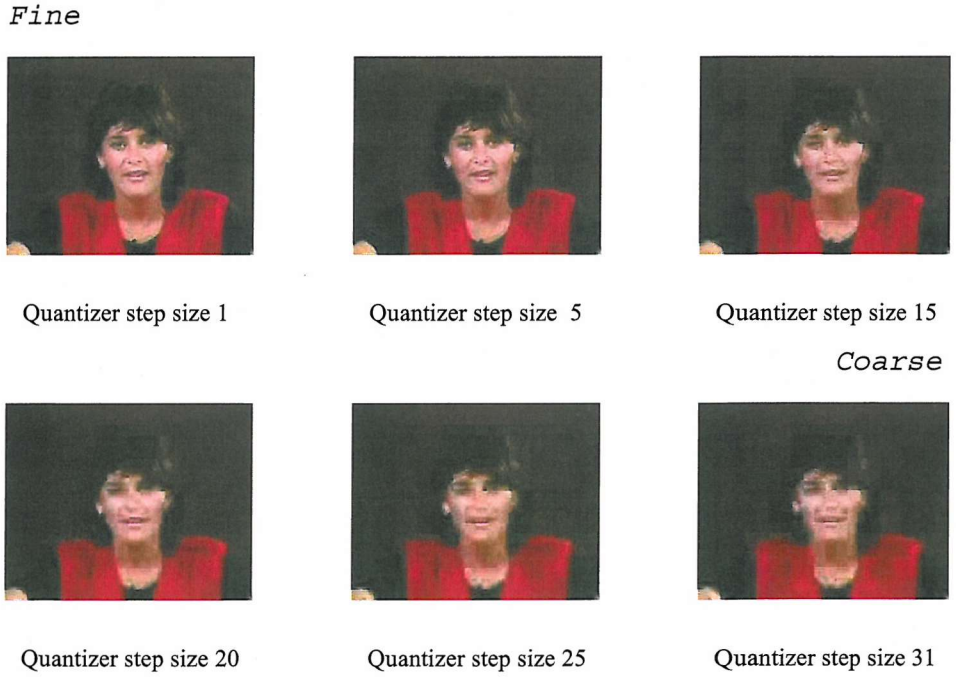


Figure 2.22: The effect of quantisation step size on the MPEG-4 encoder in the 2nd frame of the 144x176 pixel QCIF "Miss America" sequence.

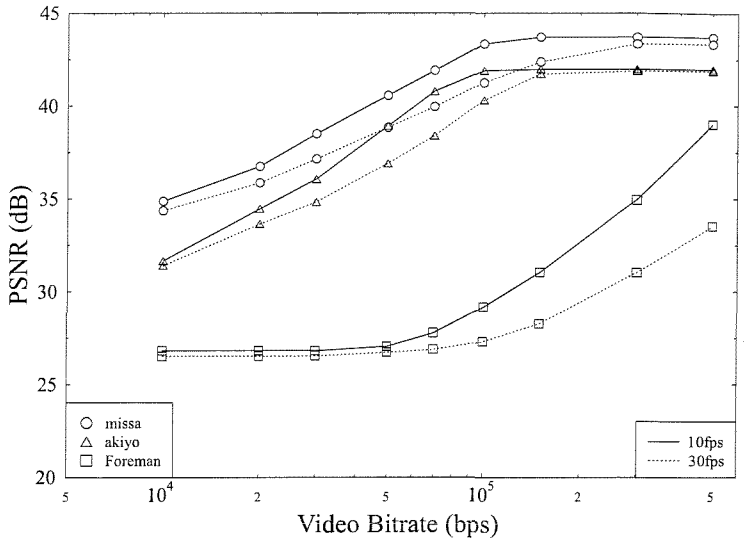


Figure 2.23: MPEG-4 video quality (PSNR) versus coded bit rate for the (144x176)-pixel QCIF "Miss America","Akiyo" and "Foreman" sequences.

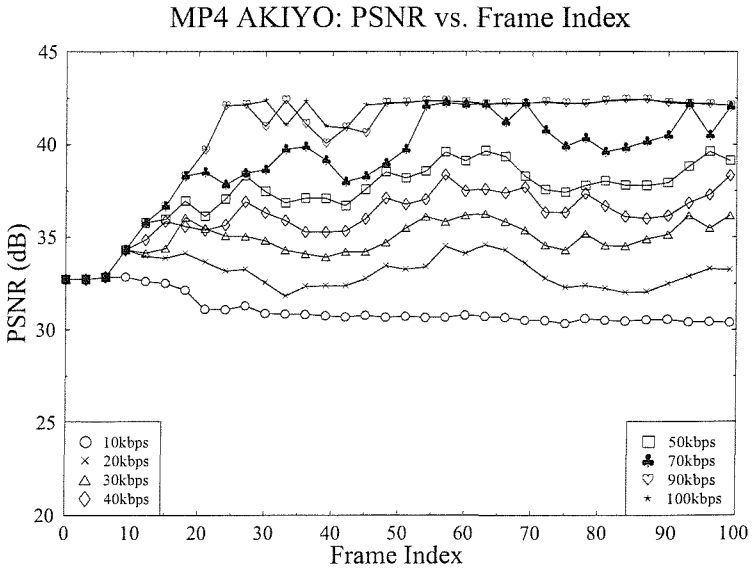


Figure 2.24: MPEG-4 video quality (PSNR) versus frame index performance for the bit rate range spanning from 10kbps to 100kbps using the (144x176)-pixel QCIF "Akiyo" sequence.

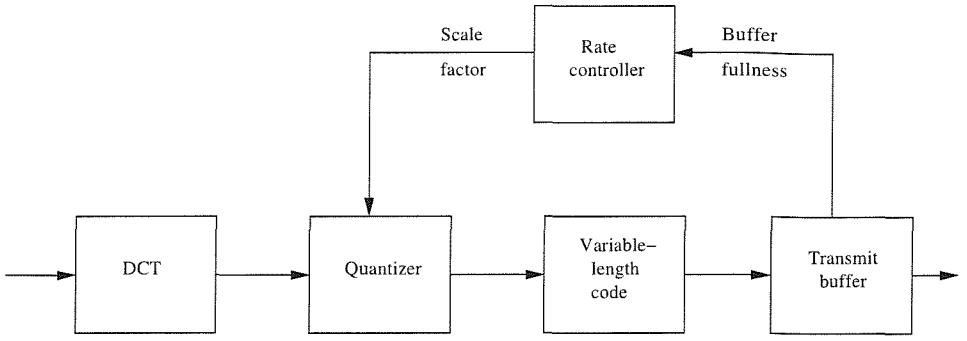


Figure 2.25: Rate control in MPEG-4.

blocks that have not yet been encoded. Figure 2.25 shows the rate control mechanism of the basic MPEG-4 encoder. The actions of the rate-controller are characterised in Figures 2.23 and 2.24. More specifically, Figure 2.23 characterises the average PSNR versus bit-rate performance, while Figure 2.24 shows the PSNR versus the video frame index. The bit rate range used in these simulations spanned between 10kbit/s (kbps) and 100kbps. As expected, it can be seen from the graphs that the video quality increases upon increasing the video bit-rate. Observe in Figure 2.23 that the PSNR versus video bitrate curves reach a saturation point above 100 kbps for both the "Miss America" and the "Akiyo" video sequences, especially at 10 frames per second (fps) scenario. This is because they are relatively slow-motion 'head and shoulder' video sequences compared to the "Foreman" video sequence. Therefore a lower bit-rate was required for both the "Miss America" and "Akiyo" sequences. Figure 2.24 plots the PSNR versus frame index performance for the "Akiyo" sequence scanning at the video frame rate of 10 fps in bit-rate range spanning from 10 kbps to 100 kbps.

So far, our experiments were carried out in an error-free environment for various encoding parameters. The next experiments were carried by transmitting the MPEG-4 bitstream over an Additive White Gaussian Noise (AWGN) channels using a simple BPSK modem. Note that the system used in this experiment employed no channel coding. Figure 2.26 portrays the PSNR versus video frame index performance for transmission over the AWGN channel for the channel SNR range spanning from 9dB to 14dB. Figure 2.27 shows the average luminance (Y) PSNR versus Bit Error Ratio (BER). Observe in Figures 2.26 and 2.27 that as expected, the decoded picture quality increases as the channel SNR increases. At low channel SNRs, the channel inflicts a high BER, hence the decoded video frame sequence will be highly degraded. This is demonstrated in Figure 2.29. Viewing the codec's performance from a different perspective, Figure 2.28 shows the average PSNR degradation versus BER.

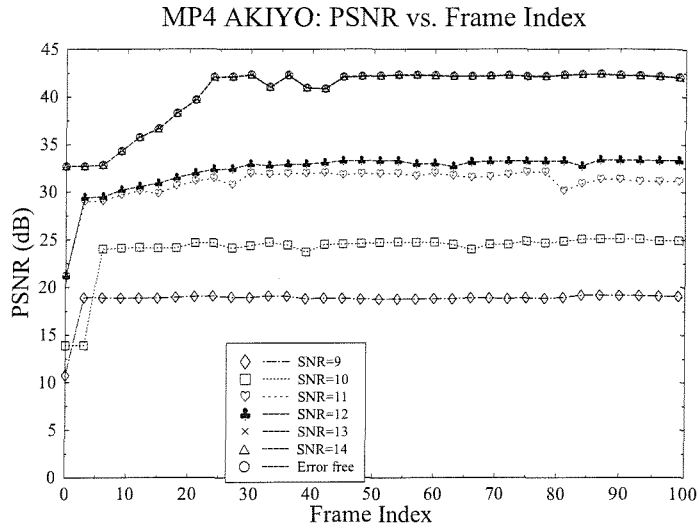


Figure 2.26: Decoded video quality (PSNR) versus frame index when transmitting over an AWGN channel using a BPSK modem. These results were recorded for the 'Akiyo' video sequence at a resolution of (144x176)-pixels and 100kbps video bitrate. The 81st decoded video frame was shown in Figure 2.29

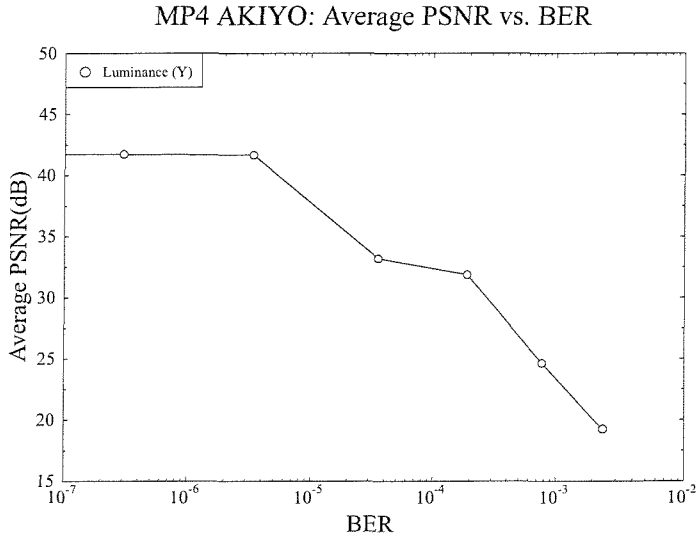


Figure 2.27: Average video quality (PSNR) versus BER for transmission over an AWGN channel using BPSK modulation and the (144x176)-pixel QCIF 'Akiyo' sequence.

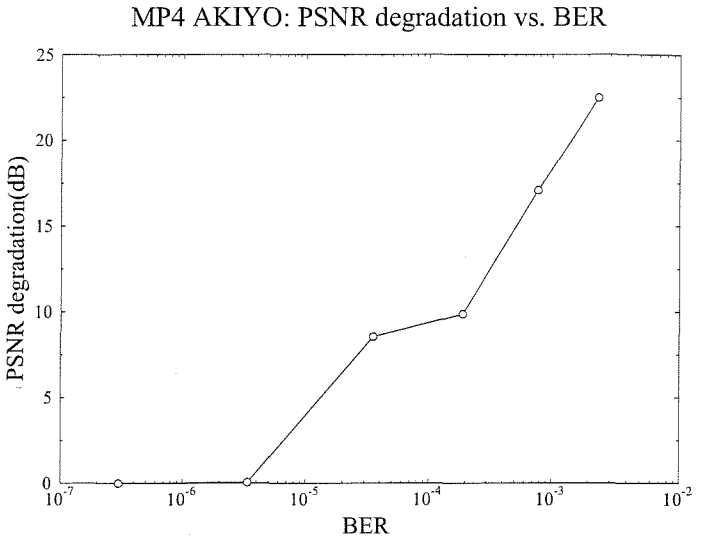


Figure 2.28: Decoded video quality degradation in terms of PSNR versus Bit Error Rate (BER) for transmission over an AWGN channel using BPSK modulation due to corruption of the MPEG-4 bit stream during transmission. The results were recorded for the QCIF resolution "Akiyo" video sequence.

2.7 Summary and Conclusion

In Section 2.1, we commenced our discourse by a brief historical perspective on the development of the MPEG-4 visual standard, the MPEG-4 visual profiles and features. These include the diverse set of coding tools described in the standard that are capable of supporting a wide range of applications, such as efficient coding of video frames, video coding for error-prone transmission networks, object-based coding and manipulation, as well as the interactive visual applications.

In Section 2.3, we introduced the MPEG-4 visual encoding standard that supports an object-based representation of the video sequence. This allows convenient access to and manipulation of arbitrarily shaped regions in the frames of the video sequence.

Scalable coding is another of the feature supported by the MPEG-4 codec. Scalability is supported in terms of generating several layers of information. In addition to the base layer, enhancement layers may be decoded, which will improve the resultant image quality either in terms of the achievable temporal or spatial resolution.

In Section 2.5 the PSNR objective measure was introduced for quantifying the video quality. Finally, the effect of certain important video encoding parameters were discussed in Section 2.6,

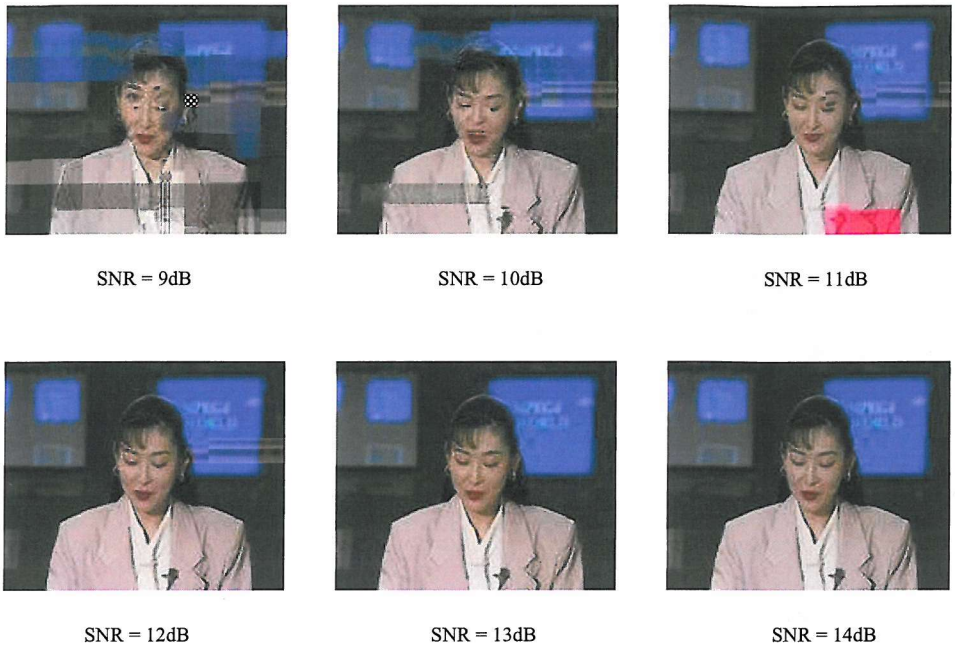


Figure 2.29: Image degradation due to corruption of the MPEG-4 bitstream over AWGN channels using a BPSK modem in the 81st frame of the (144x176)-pixel QCIF "Akiyo" sequence. The BER within this frame at the above SNRs were between 10^{-7} to 10^{-3} respectively, while the PSNR ranged from 41.74dB to 19.27dB.

commencing with an emphasis on the *quantiser step size* parameter, which also affects the resultant video *target bitrate*.

The specifications of MPEG-4 standard continue to evolve with the addition of new tools, such as the recently introduced profile supporting video streaming [15]. However, amongst developers and manufacturers, the most popular elements of the MPEG-4 Visual standard to date have been the simple and the advanced simple profile tools, which were summarised in Table 2.1. Having studied the MPEG-4 codec, let us know focus our attention on one it relative, namely on the H.264 [17] codec in the next Chapter.

Chapter 3

Comparative study of the MPEG-4 and H.264 codecs

3.1 Introduction

In this chapter, we provide a comparative study between the MPEG-4 and H.264 codecs and briefly outline the role of the ISO MPEG and ITU VCEG groups in developing these standards. Creating, maintaining and updating the ISO/IEC 14496 (MPEG-4) standards is the responsibility of the Moving Picture Experts Group (MPEG) under the auspices of the International Standards Organization (ISO). The H.264 Recommendation (also known as MPEG-4 Part 10, 'Advanced Video Coding' and formerly known as H.26L [10]) emerged as a joint effort of the MPEG and the Video Coding Experts Group (VCEG), another study group of the International Telecommunications Union (ITU).

Prior to these activities, MPEG developed the highly successful MPEG-1 and MPEG-2 standards for encoding of both video and audio, which are now widely used for the transmission and storage of digital video, as well as the MPEG-4, MPEG-7 and the MPEG-21 standards. By contrast, VCEG was responsible for the first widely-used video telephony standard (H.261) and its successors, the H.263, H.263+ and H.263++ and schemes as well as for the early development of the H.264 codec. The two groups set-up the collaborative Joint Video Team (JVT) for the sake of finalising the H.264 proposal and for creating an international standard (H.264/MPEG-4 Part 10) published by both ISO/IEC and ITU-T.

3.2 The ITU-T H.264 Project

The H.264 standard is the result of a recent joint research initiative of the International Telecommunications Union - Telecommunications Standardization Sector (ITU-T) Video Coding Experts Group (VCEG) and the ISO/IE MPEG standardization committee. The H.264 codec offers substantially improved coding efficiency at the same video quality as the MPEG-4 or H.263 schemes [10]. The main goal of this new ITU-T H.264 standardization effort was that of enhancing the achievable compression ratio, while providing a"network-friendly" packet-based video representation addressing both real-time"conversational" video telephony and "non-conversational", i.e. storage, broadcast or streaming type applications [10].

One of the fundamental concepts of H.264 is the separation of the design into two distinct layers, namely the video coding layer and the network adaptation layer. The video coding layer is responsible for efficiently representing the video content, while the network adaptation layer is responsible for packaging the coded data in an appropriate manner for transmission over the network [161]. In this chapter we focus our attention mainly on the video coding layer.

3.3 H.264 Video Coding Techniques

The coding techniques involved by the H.264 codec are to adopt similar to the schemes that have been successfully employed in earlier video coding standards. Hence, the same basic functional elements such as the prediction, transform, quantisation and entropy coding stages of the previous standards such as the MPEG-1, MPEG-2, MPEG-4, H.261 and H.263 codecs may be readily identified. Moreover, there are some important changes in each functional element of the H.264 scheme.

The schematic diagram of the H.264 encoder is depicted in Figure 3.1. The encoder consists of two data flow paths, a "forward" path oriented from left to right and a "reconstruction" path evolving from right to left. The H.264 corresponding decoder is shown in Figure 3.2, which mimics the structure of the H.264 encoder's local decoder.

3.3.1 H.264 Encoder

An input frame X_n is presented for encoding in Figure 3.1. The frame is processed in units of a macroblock corresponding to 16×16 pixels in the original image. As in the previous standards, each macroblock is encoded in either the intra- or inter-frame coding mode. In either case, a predicted macroblock **P** is formed based on a reconstructed frame. In intra-frame mode, **O** is formed from samples in the current frame, namely frame n that have been previously encoded, decoded and reconstructed, resulting in the reconstructed frame X'_n in Figure 3.1. By contrast, in the inter-frame coding mode, **P**

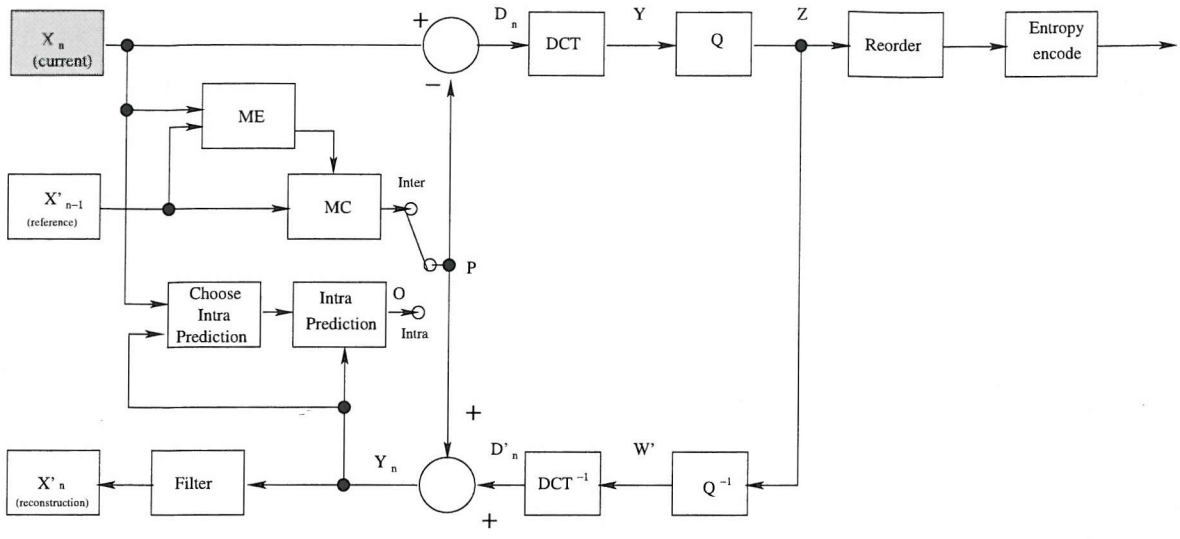


Figure 3.1: Schematic of the H.264 encoder

is formed by motion-compensation aided prediction from one or more reference frame(s), in Figure 3.1, the reference frame is shown as the previous encoded frame X'_{n-1}. Note that the prediction of each macroblock may be formed by a single or up to five past or future frames that have already been encoded, reconstructed and stored in the reconstruction frame buffer.

As seen in Figure 3.1, the predicted macroblock **P** is subtracted from the current macroblock for the sake of producing a Motion Compensated Error Residual (MCER) or difference macroblock D_n, which is then Discrete Cosine Transformed (DCT) and quantised to produce a set of quantised transform coefficients (Tcoeff) **Z**. These Tcoeffs are re-ordered using zig-zag scanning [4] and entropy encoded. The entropy encoded DCT coefficients, together with the accompanying side information such as the macroblock prediction mode, quantiser step size, motion vector information required for decoding the macroblock form the compressed bitstream. After concatenating the control headers the encoded information is then transmitted via the channel in a compressed bitstream format.

In the reconstruction path, the quantised macroblock coefficients **Z** are decoded in order to reconstruct a frame, which may then be used for the encoding of further macroblocks. The coefficients **Z** are inverse quantised in the block Q^{-1} of Figure 3.2 and inverse transformed in the block T^{-1} for producing the reconstructed MCER macroblock D'_n. Note that this is not identical to the original difference macroblock D_n; the quantisation process introduces granular effects and therefore D'_n is an approximate replica of D_n. The predicted macroblock **P** is then added to D'_n for creating an approximate replica of the original macroblock, namely the reconstructed macroblock X'_n. Finally, a smoothing filter is applied for a reducing blocking effects and the reconstructed reference frame is created from a series of macroblocks X'_n, depending on how many of the previous macroblocks were involved for creating the MCER.

3.3.2 H.264 Decoder

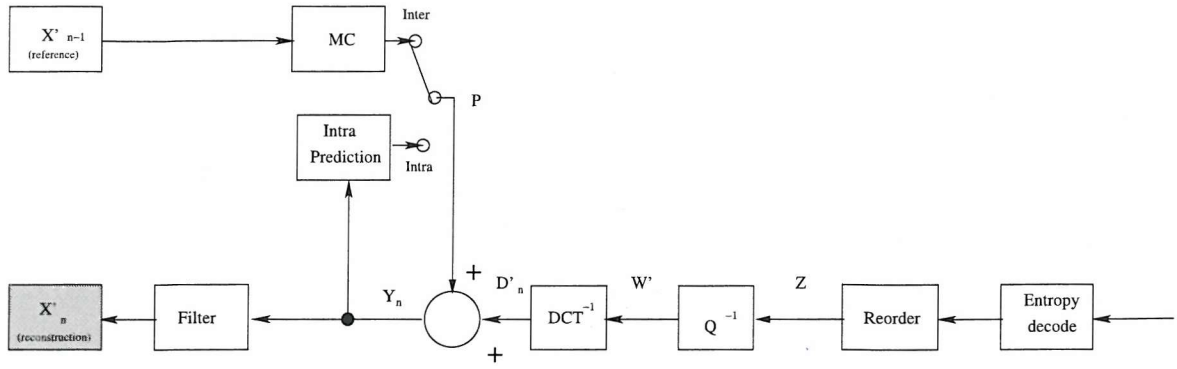


Figure 3.2: Schematic of the H.264 decoder

The decoder receives a compressed bitstream from the channel. The bitstreams are entropy decoded and appropriately ordered for the sake of producing a set of DCT coefficients **Z**. These are then inverse quantised and also inverse transformed for the sake of producing the decoded MCER D'_n. In the absence of transmission errors this quantity is identical to D'_n generated by the encoder of Figure 3.1. In the decoder, the first group of information to be decoded is the header information section of the received bitstream. The decoder then creates a predicted macroblock P, which is again, in the absence of transmission errors identical to the original predicted frame P formed in the encoder. This predicted frame P is then added to D'_n for the sake of producing X'_n, which is then smoothed by a filter for the sake of creating the decoded macroblock X'_n.

3.4 H.264 Specific Coding Algorithm

Having described the encoding and decoding process of the H.264 encoder and decoder, in this section we investigate some additional features of the H.264 standard. Some examples of the novel techniques first introduced into standard codecs, are constituted for example by spatial prediction [] in intra-frame coding [], motion compensation using an adaptive block size [], 4 × 4 pixel integer DCT [], Universal Variable Length Coding (UVLC) [162], Context-based Adaptive Binary Arithmetic Coding (CABAC) [163] and de-blocking filtering [], etc.

3.4.1 Intra-frame Prediction

Intra-frame coded pictures are typically encoded by directly applying the DCT to the different macroblocks in the frame. The intra-frame encoded pictures typically generate a high number of bits, since no temporal redundancy removal is used as part of the encoding process. In order to increase the

achievable efficiency of the intra-frame coding process in the context of the H.264 codec, the spatial correlation between adjacent macroblocks of a given frame is exploited. The associated philosophy is based on the observation that statistically speaking the adjacent macroblocks tend to have similar luminance and colour difference signals. Therefore, as a first step in the H.264 encoding process of a given macroblock, one may predict the macroblock of interest from the surrounding macroblocks of the intra-frame coded picture, which are the MBs located at the top and left of the macroblock, since those macroblocks would have already been encoded.

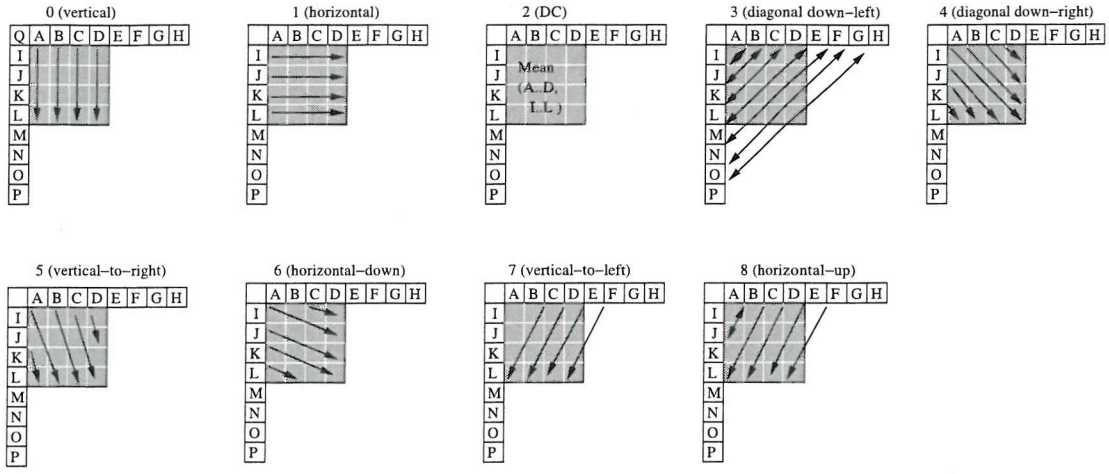


Figure 3.3: 4 × 4-pixel intra-frame prediction modes.

For the luminance of the corresponding MBs intra-frame prediction may be performed for each 4 × 4-pixel sub-block or for a 16 × 16-pixel macroblock. The H.264 codec offers a total of eight different modes for the prediction of the 4 × 4-pixel luminance blocks. The predicted block is calculated based on the samples labelled A-Q[1] in Figure 3.3, where the samples A-Q are at the edge of the neighbouring sub-blocks. Specifically, we may observe in Figure 3.3 that samples A-H are to the top and samples I-J are to the left of the 'current' MB which have previously been encoded and reconstructed. The Intra prediction process is illustrated in Table 3.1 and Figure 3.3. The arrows in Figure 3.3 indicate the direction of prediction in each mode.

For picture regions exhibiting less spatial detail, H.264 also supports intra-frame coding based on 16 × 16-pixel blocks. As an alternative to the 4 × 4-pixel intra-frame prediction modes described above, the entire 16 × 16-pixel luminance component of a macroblock can be predicted in one of four possible prediction modes, as shown in Table 3.2 and Figure 3.4.

[1]Note that each letter A to Q in Figure 3.3 represents a single pixel in the MB, where A to H are the horizontal neighbouring pixels at the top of the current MB. Furthermore, I to P are the vertical neighbouring pixels that are to the left of the current MB, while Q belongs to the bottom right pixel of another MB positioned at the top left corner of the current MB.

Modes	Description
Mode 0 (Vertical)	The pixels A,B,C,D are used for vertical prediction.
Mode 1 (Horizontal)	The left pixels I,J,K,L are used for horizontal prediction.
Mode 2 (DC)	All pixels are predicted by the mean of A-D and I-L.
Mode 3 (Diagonal Down-Left)	The pixels are diagonally predicted at a angle of 45° between the lower-left and upper-right corners of the block.
Mode 4 (Diagonal Down-Right)	The pixels are at an angle of 45° downwards with an orientation of left to the right.
Mode 5 (Vertical-Left-to-right)	Prediction at an angle of 22.5° with respect to the vertical axis from left to right.
Mode 6 (Horizontal-Down)	Prediction at an angle of 22.5° horizontal from left to right.
Mode 7 (Vertical-Right-to-left)	Prediction at an angle of 22.5° to the left of vertical.
Mode 8 (Horizontal-up)	Extrapolation at an angle of 22.5° above horizontal.

Table 3.1: 4x4 sub-block Intra Prediction modes

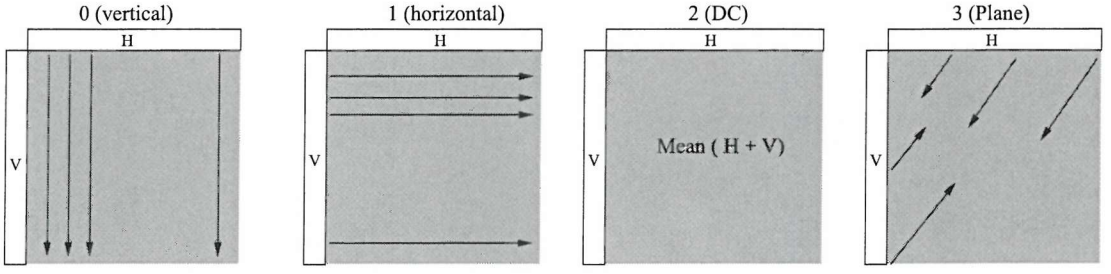


Figure 3.4: 16x16 Intra prediction modes.

Modes	Description
Mode 0 (vertical)	prediction from the upper pixels (H).
Mode 1 (Horizontal)	prediction from the left pixels (V).
Mode 2 (DC)	All pixels are predicted from the mean of the left and upper samples.
Mode 3 (Plane)	A linearly sloping 'plane' function is fitted to the upper and left-hand side pixels H and V. This prediction mode performs well in areas of smoothly-varying luminance.

Table 3.2: 16 × 16-pixel intra-frame prediction modes

Finally, the prediction mode of each block is encoded by assigning shorter prediction-mode signalling symbols to more likely modes, where the probability of each mode is determined based on the modes used for encoding the surrounding blocks.

3.4.2 Inter-frame Prediction

In inter-frame coding motion estimation and compensation is employed for the sake of exploiting the temporal redundancies that exist between successive frames, hence providing achieving an efficient encoding of video sequences. The motion estimation regime of the H.264 codec supports most of the features found in earlier video standards [161], however its efficiency was further improved. The following sub-sections describe in detail the four main motion estimation modes that are used in H.264, which are (1) the employment of various shapes and block sizes, (2) the use of high-precision sub-pixel motion vectors, (3) the utilisation of multiple reference frames, and (4) the employment of de-blocking or smoothing filters in the prediction loop.

3.4.2.1 Block sizes

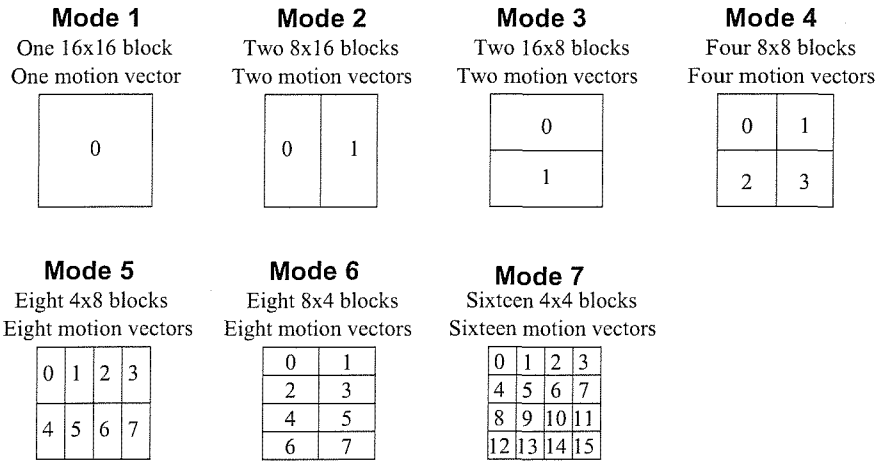


Figure 3.5: Different inter-frame prediction modes invoked for dividing a macroblock into sub-blocks for the sake of improving the accuracy of motion estimation in H.264

Motion compensation of each of the 16×16-pixel macroblock can be performed using a number of different block or sub-block sizes and shapes. These are depicted in Figure 3.5. As shown in the figure, seven different block sizes are supported in H.264, namely $16 \times 16, 8 \times 16, 16 \times 8, 8 \times 8, 4 \times 8, 8 \times 4, 4 \times 4$-pixel blocks [161]. Individual motion vectors can be transmitted for blocks as small as 4×4, therefore up to a total of 16 motion vectors may be transmitted for a single macroblock. The advantage of having smaller motion compensation blocks is that it has the potential of improving the quality of

prediction. In particular, the employment of small blocks allows the model to handle fine motion details and hence typically results in an improved subjective video quality, because it prevents the occurrence of blocking artifacts. Table 3.3 shows the luminance PSNR results of the various motion compensation search modes. Experiments were conducted for three different QCIF resolution video sequences, namely for the "Foreman", "Suzi" and "Miss America" sequences. Motion Compensation using a small 4 × 4-pixel block yields a better picture quality in terms of PSNR, than the larger 16 × 16-pixel block size.

	Y-PSNR (dB)		
Block Size	"Foreman"	"Suzi"	"Missa"
16 × 16	35.39	37.12	40.01
16 × 8	35.55	37.20	40.14
8 × 16	35.59	37.22	40.22
8 × 8	35.85	37.34	40.35
8 × 4	35.98	37.41	40.42
4 × 8	36.00	37.43	40.44
4 × 4	36.01	37.45	40.45

Table 3.3: Luminance PSNR results for the various motion compensation search modes.

3.4.2.2 Motion Estimation Accuracy

The prediction capability of the motion compensation algorithm used in the H.264 codec may be further improved by allowing motion vectors to be determined with a higher spatial accuracy than in the existing hybrid coding standards. Quarter-pixel accurate motion compensation is currently the lowest-accuracy form of motion compensation in H.264, in contrast with prior standards based primarily on half-pixel accuracy, with quarter-pixel accuracy only available elsewhere in the newest version of MPEG-4, while eight-pixel accuracy is being adopted as a feature that will likely be useful for increased coding efficiency at high bit rates and high video resolutions.

3.4.2.3 Multiple Reference Frame Selection for Motion Compensation

The H.264 standard also offers the advanced option of having multiple reference frames in inter-frame picture coding. Up to five different reference frames could be selected, resulting in an improved subjective video quality and more efficient encoding of the video frame under consideration. Moreover, using multiple reference frames might assist in rendering the H.264 coded bit stream more error resilient. However, from an implementation point of view, an additional processing delay, increased

implementational complexity and higher memory requirements are imposed at both the encoder and decoder. Table 3.4 shows the achievable PSNR results for different numbers of reference pictures. As can be seen in Table 3.4, the video quality of the codec relying on an increased number of reference frames for motion compensation is increased.

	Y-PSNR (dB)		
No. of reference frames	"Foreman"	"Suzi"	"Missa"
1	36.06	37.34	40.32
2	36.15	37.42	40.47
3	36.18	37.44	40.48
4	36.23	37.44	40.48
5	36.24	37.44	40.48

Table 3.4: Luminance PSNR for different number of frames invoked during motion compensation.

3.4.2.4 De-blocking Filter

The H.264 codec specifies the employment of a deblocking or smoothing filter that mitigates the visual affects of the horizontal and vertical block edges which may appear as consequence of truncating the two-dimensional video frame at the block-edges before the DCT is invoked. More explicitly, this truncation operation imposes two-dimensional gibbs-oscillation in the frequency domain. The filtering operation is generally based on 4x4-pixel block boundaries, in which two pixels on either side of the boundary may be updated using a 3-tap smoothing filter. The rules of applying the de-blocking filter are intricate and quite complex. Therefore, substantial research efforts are dedicated to reducing the complexity of the de-blocking filter, which is expected to further halve before the H.264 standard is finalised.

3.4.3 Integer Transform

Following the prediction process, each MCER macroblock is transformed, quantised and entropy coded. Earlier standards such as MPEG-1, MPEG-2, MPEG-4 and H.263 used the well known 8×8-pixel Discrete Cosine Transform (DCT) [5, 164] as their basic transform. The H26L codec is unique in that it employs a purely integer valued spatial transform, which is an approximation of the DCT. The transform process operates on 4×4-pixel blocks of MCER data. The H.264 codec uses three different transforms, depending on the type of MCER data as detailed below. (1) Transformation of 4×4-pixel luminance DC coefficients in intra macroblock. The order of the transformation steps

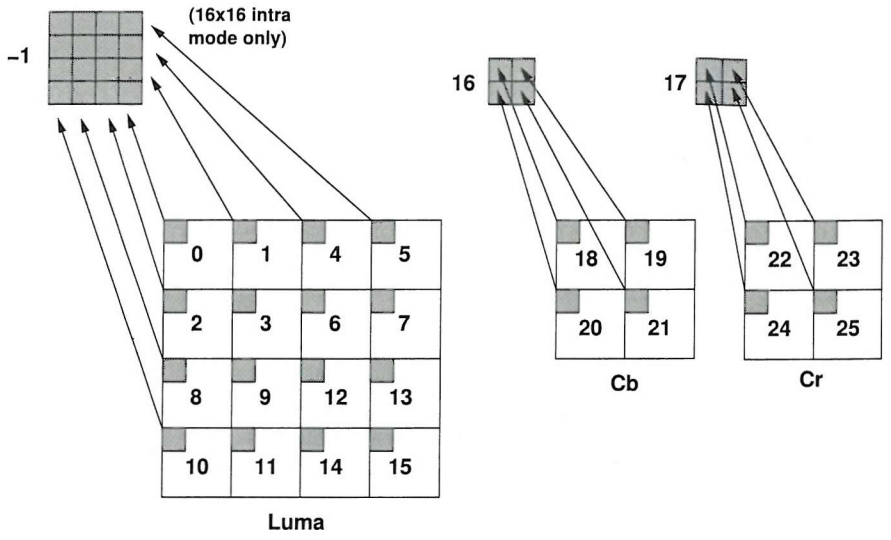


Figure 3.6: Scanning order of residual blocks within a macroblock

within a macroblock of the H.264 codec is shown in Figure 3.6. The numerical values indicated in the figure represent the order of encoding for each block in a macroblock. More explicitly, if the macroblock is coded in the 16 × 16-pixel Intra-frame coded mode, the block labelled as "-1" in Figure 3.6 is transformed first. This block contains the DC coefficient of each 4 × 4 luminance (luma) block or otherwise known as **Y**-block. Then the luma residual blocks labelled 0-15 are transformed with their DC coefficients set to zero in a 16 × 16 Intra-frame coded macroblock. Furthermore, (2) Transformation of 2 × 2 array of chroma DC coefficients. Blocks 16 and 17 consist of a 2 × 2 array of DC coefficients representative of the C_b and C_r chroma components, respectively, as seen in Figure 3.6. (3) transformation of all other 4 × 4 blocks (both luma and chroma) in the residual data. Finally, the chroma residual blocks 18-25 having zero DC coefficients are transformed.

The 4 × 4 block size assists in reducing blocking and ringing artifacts, while the employment integer-valued operations eliminates any mismatch between the encoder and decoder in the inverse transform. The 4×4-pixel integer transform is based on the DCT, although there are some fundamental differences:

1. It is an integer-valued transform where, all operations can be carried out using an integer-valued arithmetic, which results in no loss of accuracy.

2. The inverse transform is fully specified in the H.264 standard [161] and provided that this specification is followed, no mismatch occurs between the encoder and decoder.

The entire process of integer transformation and quantisation may be carried out using a 16-bit integer-valued arithmetic and only a single multiplication per coefficient is required without any loss

of accuracy.

3.4.3.1 Development of the 4 × 4-pixel Integer DCT

The DCT was developed by Ahmed et al. in 1974 [165]. There are four slightly different versions of the DCT [145] and the one commonly used for video coding is referred to as DCT type II or simply as DCT-II. The two-dimensional (2D) DCT-II of an $N \times N$ block of pixels is given by [5, 145]:

$$F(U,V) = C(u)C(v)\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)\cos\left(\frac{(2x+1)u\pi}{2N}\right)\cos\left(\frac{(2y+1)v\pi}{2N}\right), \tag{3.1}$$

where $f(x,y)$ is the pixel value at location (x,y) within the block, $F(U,V)$ is the corresponding transform coefficient, where we have $0 \leq u,v,x,y \leq N-1$, and

$$C(u) = C(v) = \begin{cases} \sqrt{\frac{1}{N}} & u,v = 0 \\ \sqrt{\frac{2}{N}} & \text{otherwise.} \end{cases} \tag{3.2}$$

The DCT operation of Equation 3.1 can be expressed in terms of matrix multiplications as:

$$\mathbf{Y} = \mathbf{AXA}^T, \tag{3.3}$$

where $\mathbf{X}$ represents the original image block and $\mathbf{Y}$ represents the resultant DCT coefficients. The elements of $\mathbf{A}$ are defined for an $M \times N$-pixel image block as follows:

$$A_{mn} = k_n \cos\left[\frac{(2m+1)n\pi}{2N}\right], \quad \begin{matrix} m & = 0,1,\ldots,M-1 \\ n & = 0,1,\ldots,N-1. \end{matrix} \tag{3.4}$$

where

$$k_n = \begin{cases} \sqrt{\frac{1}{N}} & n = 0 \\ \sqrt{\frac{2}{N}} & n = 1,2,\ldots,N-1. \end{cases} \tag{3.5}$$

For example, for a 4 × 4-pixel image block size, Equation 3.4 becomes:

$$A_{mn} = k_n \cos\left[\frac{(2m+1)n\pi}{8}\right], \quad \begin{matrix} m & = 0,1,2,3 \\ n & = 0,1,2,3. \end{matrix} \tag{3.6}$$

where we have:

$$k_n = \begin{cases} \frac{1}{2\sqrt{2}} & n = 0 \\ \frac{1}{2} & n = 1,2,3. \end{cases} \tag{3.7}$$

Let us now describe the DCT of a 4×4-pixel array $\mathbf{X}$. From Equation 3.3 we arrive at:

$$\mathbf{Y} = \mathbf{AXA}^T = \begin{bmatrix} a & a & a & a \\ b & c & -c & -b \\ a & -a & -a & a \\ c & -b & b & -c \end{bmatrix} \begin{bmatrix} x_{00} & x_{01} & x_{02} & x_{03} \\ x_{10} & x_{11} & x_{12} & x_{13} \\ x_{20} & x_{21} & x_{22} & x_{23} \\ x_{30} & x_{31} & x_{32} & x_{33} \end{bmatrix} \begin{bmatrix} a & b & a & c \\ a & c & -a & -b \\ a & -c & -a & b \\ a & -b & a & -c \end{bmatrix}, \tag{3.8}$$

where according to Equation 3.6 we have:

$$\begin{aligned} a &= \frac{1}{2} \\ b &= \sqrt{\frac{1}{2}}\cos(\frac{\pi}{8}) \\ c &= \sqrt{\frac{1}{2}}\cos(\frac{3\pi}{8}). \end{aligned}$$

It can be readily shown that the matrix multiplication of Equation 3.8 can be factorised [161] according to the following equivalent form:

$$\begin{aligned} \mathbf{Y} &= (\mathbf{CXC}^T) \otimes \mathbf{E} \\ &= \left(\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & d & -d & -1 \\ 1 & -1 & -1 & 1 \\ d & -1 & 1 & -d \end{bmatrix} \begin{bmatrix} x_{00} & x_{01} & x_{02} & x_{03} \\ x_{10} & x_{11} & x_{12} & x_{13} \\ x_{20} & x_{21} & x_{22} & x_{23} \\ x_{30} & x_{31} & x_{32} & x_{33} \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & d \\ 1 & d & -1 & -1 \\ 1 & -d & -1 & 1 \\ 1 & -1 & 1 & -d \end{bmatrix} \right) \otimes \\ &\quad \begin{bmatrix} a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \\ a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \end{bmatrix}, \end{aligned} \tag{3.9}$$

where $\mathbf{CXC}^T$ is the "core" 2-D transform in the integer transform. Furthermore $\mathbf{E}$ is a matrix of scaling factors and the symbol $\otimes$ indicates that each element of $(\mathbf{CXC}^T)$ is multiplied by the corresponding scaling factor appearing in the same position in the matrix $\mathbf{E}$, which in a scalar multiplication, rather than matrix multiplication. The constants a and b are the same as before, while d is $\frac{c}{b} \approx 0.414$.

For the sake of simplifying the implementation of the transform, d is approximated by 0.5. Furthermore, in order to ensure that the transform remains orthogonal, b also has to be modified so that [161]:

$$\begin{aligned} a &= \frac{1}{2}, \\ b &= \sqrt{\frac{2}{5}}, \\ d &= \sqrt{\frac{1}{2}}. \end{aligned}$$

The 2^{nd} and 4^{th} rows of the matrix $\mathbf{C}$ and the 2^{nd} and 4^{th} columns of matrix $\mathbf{C}^T$ are scaled by a factor of 2 and the matrix $\mathbf{E}$ in Equation 3.9 is scaled down by the same factor in the appropriate positions in order to compensate. For these adjustments, this avoids multiplications by $\frac{1}{2}$ in the "core" transform $\mathbf{CXC}^T$, which would result in a loss of accuracy, when using an integer arithmetic. The final forward transform becomes:

$$\begin{aligned}\mathbf{Y} &= (\mathbf{C_f X C_f}^T) \otimes \mathbf{E_f} \\ &= \left(\begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ d & -2 & 2 & -1 \end{bmatrix} \begin{bmatrix} x_{00} & x_{01} & x_{02} & x_{03} \\ x_{10} & x_{11} & x_{12} & x_{13} \\ x_{20} & x_{21} & x_{22} & x_{23} \\ x_{30} & x_{31} & x_{32} & x_{33} \end{bmatrix} \begin{bmatrix} 1 & 2 & 1 & 1 \\ 1 & 1 & -1 & -2 \\ 1 & -1 & -1 & 2 \\ 1 & -2 & 1 & -1 \end{bmatrix} \right) \otimes \\ &\quad \begin{bmatrix} a^2 & ab/2 & a^2 & ab/2 \\ ab/2 & b^2/4 & ab/2 & b^2/4 \\ a^2 & ab/2 & a^2 & ab/2 \\ ab/2 & b^2/4 & ab/2 & b^2/4 \end{bmatrix}.\end{aligned} \tag{3.10}$$

This transform constitutes an approximation of the 4×4-pixel DCT. As a consequence of changing the factors d and b, the output of the new transform will not be identical to that of the 4×4 DCT.

The inverse transform defined in [166] is given by:

$$\begin{aligned}\mathbf{X}' &= \mathbf{C_i}^T(\mathbf{Y} \otimes \mathbf{E_i})\mathbf{C_i} \\ &= \left(\begin{bmatrix} 1 & 1 & 1 & 1/2 \\ 1 & 1/2 & -1 & -1 \\ 1 & -1/2 & -1 & 1 \\ 1 & -1 & 1 & -1/2 \end{bmatrix} \begin{bmatrix} y_{00} & y_{01} & y_{02} & y_{03} \\ y_{10} & y_{11} & y_{12} & y_{13} \\ y_{20} & y_{21} & y_{22} & y_{23} \\ y_{30} & y_{31} & y_{32} & y_{33} \end{bmatrix} \begin{bmatrix} a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \\ a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \end{bmatrix} \right) \otimes \\ &\quad \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1/2 & -1/2 & -1 \\ 1 & -1 & -1 & 1 \\ 1/2 & -1 & 1 & -1/2 \end{bmatrix}.\end{aligned} \tag{3.11}$$

This time, $\mathbf{Y}$ is pre-scaled upon multiplying each of its coefficients by the appropriate weighting factor found in the corresponding position of matrix $\mathbf{E_i}$. The factors +/- $\frac{1}{2}$ seen in the matrices $\mathbf{C}$ and $\mathbf{C}^T$, which can be implemented by a right-shift without a significant loss of accuracy, because the coefficients of $\mathbf{Y}$ are pre-scaled.

3.4.3.2 Quantisation

The H.264 codec uses a scalar DCT-coefficient quantiser. The quantiser's implementation is complicated by the requirements of (1) avoiding division and/or floating point arithmetic operations and (2) by incorporating the post- and pre-scaling matrices $\mathbf{E_f}$ and $\mathbf{E_i}$ of Equations 3.10 and 3.11, as described in Section 3.4.3.1.

The basic DCT-coefficient quantisation operation is as follows:

$$Z_{ij} = round\left(\frac{Y_{ij}}{Q_{step}}\right), \tag{3.12}$$

where Y_{ij} is a DCT coefficient generated by the transform described in Section 3.4.3.1, Q_{step} is a quantiser step size and Z_{ij} is a quantised DCT coefficient. A total of 31 step-size values of Q_{step} are supported by the standard [166] and these are indexed by the Quantisation Parameter, QP. The values of Q_{step} corresponding to each QP are shown in Table 3.5. Note that Q_{step} doubles in size for every increment of 6 in QP, Q_{step} increases by 12.5% for each increment of 1 in QP.

QP	0	1	2	3	4	5	6	7	...	10	...	20	...	31
Q_{step}	0.625	0.6875	0.8125	0.875	1	1.125	1.25	1.375	...	2	...	6.5	...	22

Table 3.5: Quantisation step sizes in the H.264 codec

The post-scaling factors a^2, $\frac{ab}{2}$ or $\frac{b^2}{4}$ of $\mathbf{E_f}$ are incorporated into the forward quantiser. First, the video input block $\mathbf{X}$ is transformed to give a block of unscaled coefficients $\mathbf{W} = \mathbf{CXC}^T$. Then, each DCT coefficient W_{ij} is quantized and scaled in a single operation according to:

$$Z_{ij} = round\left(W_{ij} \cdot \frac{PF}{Q_{step}}\right), \tag{3.13}$$

where PF is a^2, $\frac{ab}{2}$ or $\frac{b^2}{4}$ depending on their position (i, j). The scaling operation of (PF/Q_{step}) is implemented in the H.264 reference model software [166] as a multiplication by the multiplication factor of MF and a right-shift[2], thus avoiding any division operations:

$$Z_{ij} = round\left(W_{ij} \cdot \frac{MF}{2^{qbits}},\right) \tag{3.14}$$

[2]In integer arithmetic, Equation 3.14 can be implemented as follows:

$$\begin{aligned} |\,Zij\,| &= (|\,W_{ij} \cdot MF + f) \gg qbits \\ sign(Zij) &= sign(W_{ij}) \end{aligned}$$

where $\gg$ indicates a binary right-shift.

where

$$\frac{MF}{2^{qbits}} = \frac{PF}{Q_{step}} \tag{3.15}$$

$$qbits = 15 + floor\left(\frac{QP}{6}\right). \tag{3.16}$$

Let us now consider the inverse quantization process. The basic inverse quantizer or "rescale" operation is carried out as:

$$Y'_{ij} = Z_{ij} \cdot Q_{step}. \tag{3.17}$$

The pre-scaling factor used for the inverse transform of Equation 3.11 implemented with the aid of the matrix $\mathbf{E_i}$, containing values of a^2, ab and b^2 depending on the coefficient position is incorporated in this operation, together with a further constant scaling factor of 64, in order to avoid any rounding errors [166] according to:

$$W'_{ij} = Z_{ij} \cdot Q_{step} \cdot PF \cdot 64. \tag{3.18}$$

In Equation 3.18, W'_{ij} is a scaled DCT coefficient, which is then transformed by the "core" inverse transform, namely by $\mathbf{C_i}^T\mathbf{W}\mathbf{C_i}$. The decoded pixel values generated by the inverse transform are divided by 64 in order to remove the scaling factor of 64, which may be implemented by shift operations. Section 3.4.3.3 summarises the complete DCT transform, quantization, rescaling and inverse transform process, while Section 3.4.3.4 provides some examples for characterising the transformation process.

3.4.3.3 The Combined Transform, Quantisation, Rescaling and Inverse Transform Process

The entire process of generating the output residual block $\mathbf{X}'$ from input residual block $\mathbf{X}$ is described below in terms of a number of processing steps and illustrated in Figure 3.7 where the first three steps correspond to the encoding, while the remaining steps to decoding.

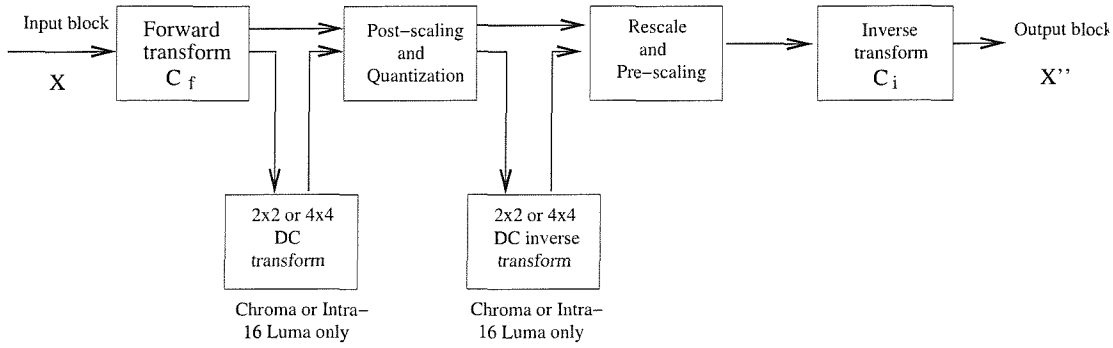


Figure 3.7: Transformation, quantization, rescale and inverse transform flow diagram

1. The input signal is constituted by the 4×4 motion compensated error residual samples contained in $\mathbf{X}$
2. Forward "core" transform: $\mathbf{W} = \mathbf{C_f X C_f}^T$
3. Post-scaling and quantization: $\mathbf{Z} = \mathbf{W} \cdot \frac{PF}{Q_{step} \cdot 2^{qbits}}$
4. Re-scaling, incorporating inverse transform's pre-scaling operation: $\mathbf{W'} = \mathbf{Z} \cdot Q_{step} \cdot PF \cdot 64$
5. Inverse "core" transform: $\mathbf{X'} = \mathbf{C_i}^T \mathbf{W'} \mathbf{C_i}$
6. Post-scaling: $\mathbf{X''} = round\left(\frac{\mathbf{X'}}{64}\right)$
7. Output: 4 × 4-pixel block of motion compensated error residual (MCER) samples: $\mathbf{X''}$

3.4.3.4 Integer Transform Example

Figure 3.8 shows a specific example of a 4×4-pixel block extracted from the "Foreman" video sequence having QCIF resolution.

Input block $\mathbf{X}$:

	j=0	1	2	3
i=0	46	49	55	64
1	38	39	48	49
2	44	38	39	42
3	63	58	56	54

Since the input MCER block appears to be fairly 'flat' the DCT output is expected to be concentrated in the top-left corner, corresponding to the DC component. Output $\mathbf{W}$ of "core" transform:

	j=0	1	2	3
i=0	776	-44	24	-2
1	-29	-152	3	-31
2	114	0	-6	0
3	-27	-6	9	7

Before Quantization and post-scaling process, we first need to determine the value of multiplication factor MF, Using the Equation 3.15, $MF = PF \times (\frac{2^{qbits}}{Q_{step}})$. In this example, $QP = 5$, hence $Q_{step} = 1.125$. Since MF is a multiplication factor, therefore some of the elements in a 4 × 4 matrix position are identical. Hence MF:

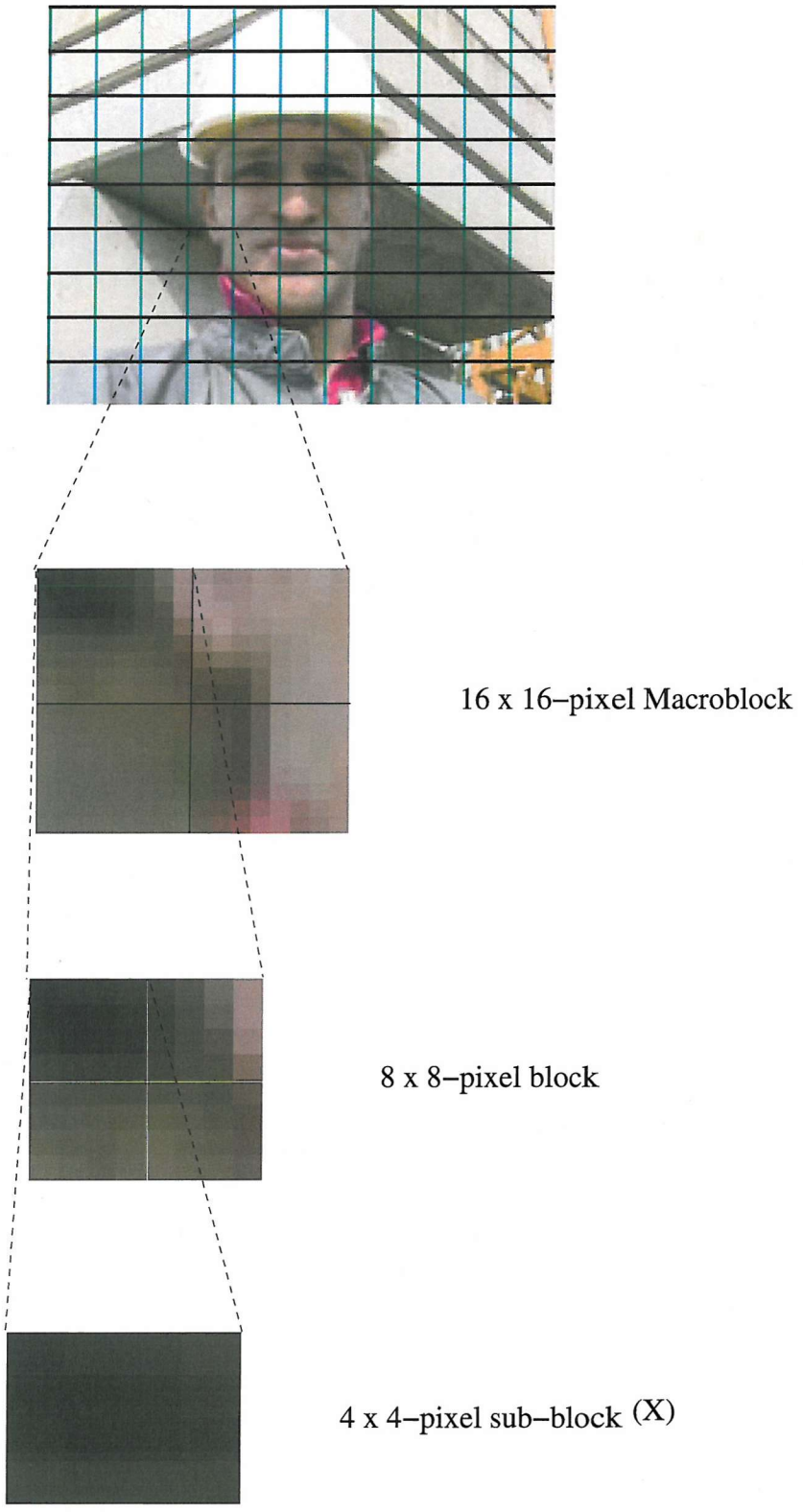


Figure 3.8: Figure shows the selected 4 × 4-pixel sub-block from a QCIF "Foreman" video sequence. The 4 × 4-pixel sub-block **X** was used as a demonstration example in this section.

	j=0	1	2	3
i=0	7281	4605	7281	4605
1	4605	2912	4605	2912
2	7281	4605	7281	4605
3	4605	2912	4605	2912

Next, from Equation 3.14, the 'core' transform **W** is multiplied to the MF producing the quantized transform coefficients **Z**. Again, **Z** is expected to be concentrated in the top-left corner, furthermore, increasing number of 'zero' element for the AC coeeficient is expected. Output of forward quantizer **Z**:

	j=0	1	2	3
i=0	172	-6	5	0
1	-4	-14	0	-3
2	25	0	-1	0
3	-4	-1	1	1

Next is the inverse quantization process, the receive quantized coefficients from the encoder is de-quantized or in other words called 're-scaled', the de-quantized coefficient before the main inverse transformation process is again has the same characteristic as the DCT output, where DC value is much more higher than AC values. Output of rescale $\mathbf{W}'$:

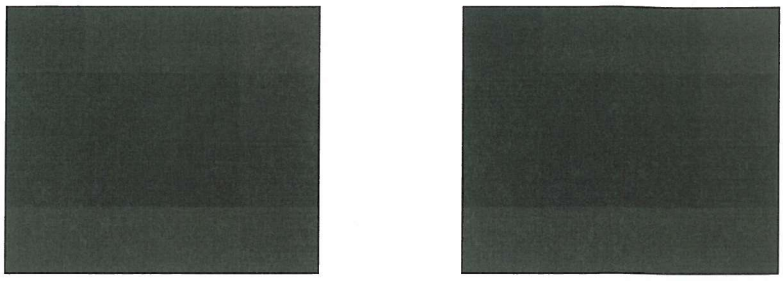
	j=0	1	2	3
i=0	3096	-136	90	0
1	-91	-403	0	-86
2	450	0	-18	0
3	-91	-28	22	22

For the inverse transformation process, the inverse transform coefficients are appear to be 'flat'. Output of $\mathbf{X}'$:

	j=0	1	2	3
i=0	2902	3120	3531	4083
1	2431	2523	2689	3122
2	2803	2415	2523	2659
3	4060	3690	3553	3426

The final step is to reconstruct, so that the reconstruct block is similar to the original block **X** Output of $\mathbf{X}''$:

	j=0	1	2	3
i=0	45	49	55	64
1	38	39	42	49
2	44	38	39	42
3	63	58	56	54



(a) Original 4 x 4 pixels (b) Reconstruction 4 x 4 pixel

Figure 3.9: Comparison of the 4 × 4-pixel sub-block extracted from the "Foreman" sequence, where (a) is the original 4 × 4-pixel sub-block before transformation and quantisation, while (b) is the reconstructed 4 × 4-pixel sub-block after it has undergone the H.264 standard's integer transformation and quantization process.

Finally, Figure 3.9 depicts the original and the reconstructed 4 × 4-pixel block of the selected "Foreman" video sequence.

3.4.4 Entropy Coding

After all the transform coefficients have been quantized, the last step in the video coding process of Figure 3.1 is entropy coding. The H.264 codec has adapted two different approaches for entropy coding. The first approach is based on the use of Universal Variable Length Codes (UVLCs) [10] and the second invokes Context-Based Adaptive Binary Arithmetic Coding (CABAC) [167]. Note that the type of entropy coding employed is selected by the user during the encoding process. When the picture parameter set flag `'entropy_coding_mode'` is set to '0', the block of quantised coefficients of either 4 × 4 or 16*times*16 are codded using the UVLC scheme, whereas when the picture parameter set flag `'entropy_coding_mode'` is set to '1', the CABAC scheme is employed.

3.4.4.1 Universal Variable Length Coding

Entropy coding [10] based on Variable Length Codes (VLCs) is the most widely used method of further compressing the quantized transform coefficients, motion vectors and other encoding parameters. VLCs are based on assigning shorter codewords to symbols having higher probability of occurrence and longer codewords to symbols with less frequent occurrences. The symbols and the associated codewords are organized in look-up tables, referred to as VLC tables, which are stored at both the encoder and decoder [167].

In some of the video coding standards such as H.263 [168] and MPEG-4 [144], a number of VLC tables are used, where each of the parameter types such as quantized DCT coefficients, motion vectors etc. have their respective VLC tables. The H.264 codec employs a single universal VLC table that is to be used in entropy coding of all symbols in the encoder, regardless of the type of data those symbols represent.

3.4.4.2 Context-Based Adaptive Binary Arithmetic Coding

Arithmetic coding makes use of a probability model at both the encoder and decoder for all the syntax elements such as transform coefficients, motion vectors. To increase the coding efficiency of arithmetic coding, the underlying probability model is adapted to the changing statistics with a video frame, through a process called context modelling. CABAC has three distinct advantages:

1. context model provides estimation of conditional probabilities of the coding symbols.
2. arithmetic code permits non-integer number if bits to be assigned to each symbol.
3. adaptive arithmetic code permits the entropy coder to adapt itself to non-stationary symbol statistics.

3.4.4.3 H.264 Conclusion

The H.264 standard provides video coding mechanisms that are optimised for compression efficiency and aim for supporting practical multimedia communication applications. The range of available coding tools available is more restricted than in the MPEG-4 Visual standard owing to its narrower focus of applications, but there are still numerous possible choices for the coding parameters and strategies. The success of a practical H.264 implementation depends on the careful design of the codec and on the effective choices of the coding parameters. In the next section, we investigate achievable the performance of both codecs and in Chapter 5, we propose a range of videophony schemes, where the H.264 video codec is employed.

3.5 Comparative Study of the MPEG-4 and H.264 Codecs

3.5.1 Introduction

Following the previous chapter introducing the basic features of the MPEG-4 codec and the above section on H.264 codec emerging from the ITU-T VCEG project, in this section we highlight the

differences between the MPEG-4 and H.264 standards. Throughout this section, familiarity with the MPEG-4 standard is assumed.

The ISO/IEC standard MPEG-4 video coding standard [50] and the ITU-T VCEG H.264 codec [50, 161] exhibit some similarities in that they both define so-called block-based hybrid video codecs. Each video frame is divided into fixed size macroblocks of 16x16 pixels, that can be encoded in several coding modes. They both distinguish between INTRA-frame coding and INTER-frame coding or predictive modes. In INTRA-frame mode, a macroblock is encoded without referring to other frames in the entire video sequence, while in the INTER-frame coded or predictive mode, previously encoded images are used as the reference for forming the MCER signal. The resultant MCER signal is encoded using transform coding, where a macroblock is subdivided into a number of fixed size blocks. Prior to MCER encoding, each of these blocks is transformed using a block transform, and the transform coefficients are quantised and transmitted using entropy coding methods, as highlighted in Sections 3.3 and 3.4 [161].

Although both the MPEG-4 Visual layer and the H.264 codec define similar coding algorithms, they contain features and enhancements that render them different. These differences mainly concern the formation of the MCER signal, as well as the block-sizes used for transform coding and the entropy coding methods. Following this brief introduction, let us now compare the features of these codecs in a little more detail in Sections 3.5.2 to 3.5.6.

functionalities allowing the applicability in a wide range of applications. Since the subject of this section is a comparison between MEPG-4 and H.264 in terms of coding efficiency for non-interlaced rectangular video with natural content, we will only investigate told in

3.5.2 Intra-frame Coding and Prediction

The MPEG-4 Visual layer and the H.263 codec have much in common [168], since the H.263 ITU-T Recommendation was the starting point of the MPEG-4 project and most additional work was carried out in the area of object-based coding, where each of the objects in a video frame was referred to as a Video Object (VO). Thus the MPEG-4 coding standard supports three different picture types during the encoding of a so-called video object plane (VOP), namely I-VOPs, P-VOPs, and B-VOPs. In I-VOPs each macroblock is encoded in INTRA-frame mode, and there is only one INTRA-frame mode in the MPEG-4 standard which uses 8×8-pixel sub-blocks. By comparison in the H.264 codec, sub-blocks of 4×4 samples are used for transform coding, and thus a sub-block consists of 16 luminance and 8 chrominance pixels, as shown in Figure 3.6. The conventional picture types known as I, P and B pictures are still supported in the standard. Unlike in MPEG-4, in H.264, a macroblock can always be encoded in one of several INTRA-frames modes. There are two classes of INTRA-frame coding modes, which are denoted as INTRA 16×16 and INTRA 4×4 in the following. Moreover, in contrast to

MPEG-4, where only some of the DCT-coefficients can be predicted from neighbouring INTRA-frame blocks, in H.264 INTRA-frame pixel prediction is always utilised in the spatial domain by referring to the neighbouring pixels of already encoded blocks. When using the INTRA-frame coded 4 × 4-pixel mode, each 4 × 4-pixel block of the luminance component utilises six prediction modes. The chosen modes are encoded and transmitted as side information. In the INTRA-frame coded 16 × 16-pixel mode, a uniform prediction is performed for the whole luminance component of a macroblock. Four different luminance modes are supported, as described in Section 3.4.1. Let us now consider the issue of intra-frame prediction in the next section.

3.5.3 Inter-frame Prediction and Motion Compensation

Both the MPEG-4 and the H.264 codec employ inter-frame prediction based on frame differencing as well as motion compensation using motion vectors. In MPEG-4, three different predictive coding modes are provided for P-VOPs, namely the INTER 16 × 16, INTER 8 × 8 and the Skip mode. For the Skip mode, a single-bit flag is required for signalling to the decoder that all samples of the entire macroblock are repeated from the reference frame. By contrast, the H.264 standard provides seven motion-compensated coding modes for macroblocks encoded in the INTER-frame mode. Each motion-compensated mode corresponds to a specific partition of the macroblock into fixed size blocks used for motion description. Blocks having sizes of 16 × 16, 16 × 8, 8 × 16, 8 × 8, 8 × 4, 4 × 8, and 4 × 4 pixels are supported by the H.264 syntax [161] and thus up to 16 motion vectors can be transmitted for a macroblock.

Furthermore, H.264 supports the employment of multi-frame motion compensated prediction, where more than one previously encoded pictures can be used as reference for motion compensation. For the current version of the H.264 codec, up to five reference frames are available for motion prediction. By contrast, in the MPEG-4 standard only one reference frame is used for motion compensation.

Half- or quarter-pixel resolution motion compensation based on the previously coded I- or P-VOP is applied in MPEG-4 for forming the MCER signal in the INTER 16 × 16 and INTER 8 × 8 modes. In the INTER 16 × 16 mode, the motion trajectory of an entire macroblock is specified by a single motion vector, whereas the INTER 8 × 8 mode uses four motion vectors rounded to values associated with half-pixel accuracy. The horizontal and vertical components of each motion vector are encoded differentially by using median prediction based on three neighbouring blocks, which have already been encoded.

The H.264 syntax supports both quarter- and 8^{th}-pixel resolution motion compensation. The motion vector components are differentially encoded using either median or directional prediction relying on the motion vectors of the neighbouring blocks. The selection of a specific prediction mode

depends on the block's shape and on its position within the macroblock. Let us now focus our attention on the encoding of the MCER in the next section.

3.5.4 Transform Coding and Quantisation

In MPEG-4 the Discrete Cosine Transform (DCT) is employed for encoding of an INTRA- or INTER-frame coded MB, but no arithmetic procedure is specified for computing the inverse transform. H.264 codec is basically similar to the MPEG-4 scheme and to other previous coding standards in that it utilises transform coding of the prediction error signal. However, in the H.264 codec the transformation is applied to 4 × 4-pixel blocks and instead of the DCT, the H.264 codec uses the separable integer transform of Section 3.4.3, which has similar properties to the 4 × 4-pixel DCT. Since the inverse DCT of Section 3.4.3 is defined by exact integer operations, inverse-transform mismatches will never be encountered. An additional 2 × 2-pixel transform is applied to the four DC-coefficients of each chrominance component. If the INTRA 16 × 16 mode is in use, a similar operation extending the length of the transforms basis functions is performed for the 4 × 4 DC-coefficients of the luminance signal.

The H.264 codec uses scalar quantisation for encoding the DCT coefficients, but without the quantiser dead-zone around zero of the MPEG-4 codec. One of the 32 quantisers is selected for each macroblock by the quantisation parameter QP. The quantiser levels are arranged such that there is an approximately 12.5% quantiser step-size increase from one QP to the next in the list of 32 quantisers. The quantised transform coefficients are scanned in a zig-zag fashion [4] and converted to compressed symbols by run-length coding (RLC). In the MPEG-4 standard the quantised DCT-coefficients of an 8 × 8-pixel block are zig-zag scanned, whereas in the H.264 standard zig-zag scanning is carried out in 4 × 4-pixel blocks. In the next section we consider the entropy coding schemes used.

3.5.5 Entropy Coding

Two different methods of entropy coding are supported by H.264. The first method is referred to as Universal Variable Length Coding (UVLC) [161], which relies on potentially infinite-length codeword sets. Instead of designing a different VLC table for each syntax element, only the mapping to the single UVLC table is customised according to the statistics of the data to be encoded. The efficiency of entropy coding can be improved if Context-Adaptive Binary Arithmetic Coding (CABAC) [167] is used. On the one hand, the employment of arithmetic coding [161] allows the assignment of non-integer number of bits to each symbol of an alphabet, which is particularly beneficial in conjunction with symbol probabilities higher than 0.5. On the other hand, the use of adaptive codes supports adaptation to non-stationary symbol statistics. Another important property of the CABAC scheme [167] is its context modelling capability [161], where the statistics of already encoded syntax elements can be

used for estimating the conditional probabilities of the symbols to be encoded. The inter-symbol dependencies may be exploited by switching amongst several estimated probability models according to the already coded symbols in the neighbourhood of the symbol to be encoded.

3.5.6 De-blocking filter

For the sake of mitigating the detrimental visual effects of block-edge artifacts, the H.264 codec design includes a deblocking filter. In contrast to the MPEG-4 codec, where the deblocking filter is an optional part of the recommendation, in the H.264 codec the filter is applied within the motion prediction loop and hence forms an integral part of the recommendation. The cut-off frequency of the filtering is adaptively controlled by the values of several coding syntax elements [161]. Following our rudimentary comparison of the elementary of the H.264 and MPEG4 codecs, let us now focus our attention on the comparison of their performance.

3.6 Performance Results

3.6.1 Introduction

Exclusive simulations were carried out using both the MEPG-4 and H.264 codecs in order to comparatively study their performance. In our experiments well-known video sequences, such as the "Miss America", "Suzi" and "Foreman" clips were used, which were scanned at video frame rates of 10 and 30 frames/s. The performance of both codecs depends highly on the quantisers step sizes used in the encoder. The upper and lower limits of video quality performance can be recorded when the finest or coarsest quantisers are employed.

Note that in our MPEG-4 and H.264 simulations we used identical YUV format input video sequences. Since both codecs support 4:2:0 format colour YUV video sequences, our experiments were carried out using QCIF format colour video sequences. Let us now turn our attention to their performance comparison.

3.6.2 MPEG-4 Performance

As stated above, in our MPEG-4 simulations we used the QCIF-format colour "Miss America", "Suzi" and "Foreman" sequences scanned at 10 and 30 frames/s (fps). Simulations were carried out using the MPEG-4 codec in the following conditions :

- Only one Intra-frame coded Picture was transmitted at the beginning of the sequence.

- Two B-pictures have been inserted between two successive P-pictures. More specifically, the frame-sequence had the following syntax: I,P,B,B,P,B,B,P.....
- The video sequence length of the "Miss America" clip was 150 frames.
- The video sequence length of the "suzi" clip was 150 frames.
- The video sequence length of the "Foreman" clip was 300 frames.
- The object based coding option was deactivated, thus we treated the entire frame as one video object.

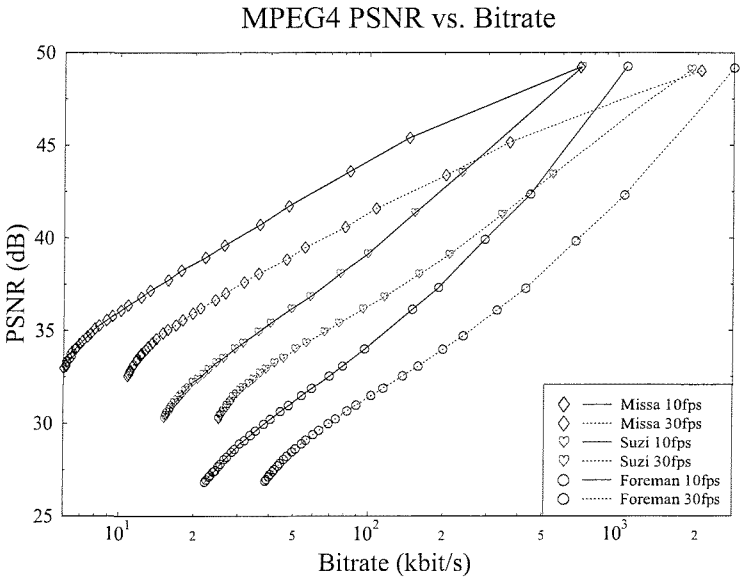


Figure 3.10: Image quality (PSNR) versus coded bit-rate performance of the MPEG-4 codec for various QCIF- resolution video sequences scanned at 10 and 30 frames/s.

The simulations were performed at a wide range of bitrates. The results of these simulations are shown in Figures 3.10 and 3.11, where Figure 3.10 characterises the average PSNR versus bit-rate performance, while Figure 3.11 plots the results from a different perspective, namely as the average PSNR versus the compression ratio. As can be seen from Figure 3.10 for a given fixed bitrate, the simulations for the "Foreman" sequence resulted in the lowest image quality, followed by the "Suzi" sequence and finally by the "Missa America" sequence. This is because the "Miss America" clip is a pure head and shoulder's type video sequence, which is associated with the least motion activity, hence less motion vector information has to be transmitted. Therefore the Skip-mode is used more often during the encoding process which generate a single bit flag. Hence a reduced number of bits

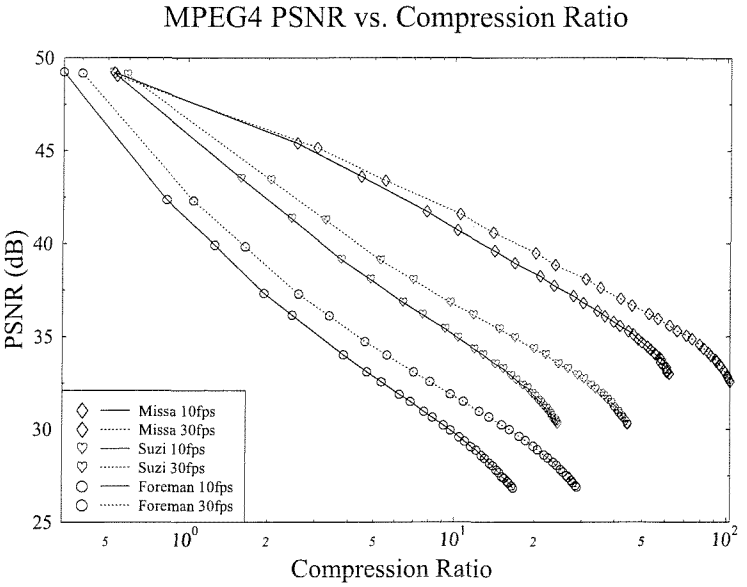


Figure 3.11: Image quality (PSNR) versus compression ratio performance of the MPEG-4 codec for various QCIF- resolution video sequences scanned at 10 and 30 frames/s.

is necessary for coding the entire sequence. By contrast, in the "Suzi", sequence more motion takes place in the sequence. Hence slightly more motion vector information was required by the motion compensation process. Additionally, less Skip-mode flags were used by the encoding process, which results in a higher coded bit-rate compared to the "Miss America" video sequence. Needless to say that the "Foreman" sequence exhibits an even higher motion activity, than the other two sequences used in our experiments, therefore it requires the highest number of bits during the encoding process, in particular for the motion vector parameters.

The image quality expressed in PSNR improved almost linearly as the bit rate was increased. When maintaining a constant bit-rate, the 30 fps curves typically exhibit a PSNR penalty due to the fixed bit-rate constraint imposed, since on the basis of simple logic at 30 fps a threefold bit-rate increase is expected in comparison to 10fps. Moreover, due to the higher inter-frame correlation of the 30 fps clip the MCER has a typically reduced variance in comparison to 10 fps coding, which mitigates the above threefold bit-rate increment estimate to an approximate ratio of two.

investigations from a different perspective, as average PSNR versus the compression ratio. The curves suggest a nearly linear relationship over the range

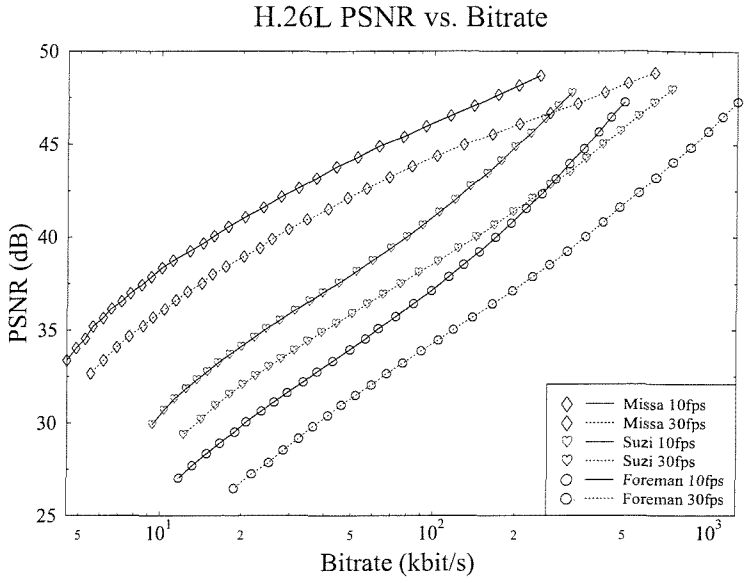


Figure 3.12: Image quality (PSNR) versus coded bit-rate performance of the H.264 codec for various QCIF- resolution video sequences scanned at 10 and 30 frames/s.

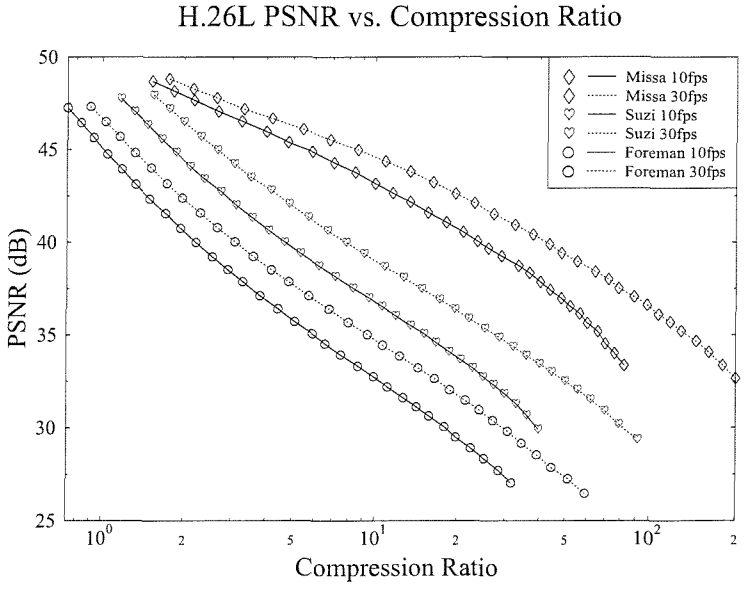


Figure 3.13: Image quality (PSNR) versus compression ratio performance of the H.264 codec for various QCIF- resolution video sequences scanned at 10 and 30 frames/s.

3.6.3 H.264 Performance

In this section, we report our findings on the ITU-T standard H.264 codec using simulation conditions similar to those employed in the MPEG-4 experiments described in previous section. Again, all simulations were conducted using the three well-known video sequences at "Miss America", "Suzi" and "Foreman" at both 10 and 30 frames/s. The additional coding options used in the H.264 simulations are listed as follows:

- Entropy coding was performed using the CABAC technique [167].
- Five reference frames were used for inter-frame prediction.
- 8^{th}-pixel resolution motion compensation was used.

Figure 3.12 portrays the image quality (PSNR) versus coded bitrate performance of the H.264 codec for the "Missa America", "Suzi" and "Foreman" sequences scanned at 10 and 30 frames/s. As can be seen from the figure, it exhibits trends to those of the MEPG-4 scheme previously seen in Figure 3.10. This was expected, since both the MPEG-4 and H.264 scheme have similar coding algorithms, as discussed earlier in Sections 3.5.

As in MPEG-4, for a given bitrate, the "Missa America" sequence had the highest image quality (PSNR) followed by the "Suzi" then the "Foreman" clips. This is in line with the amount of motion content in the sequences. As shown in Figure 3.12, the 30fps scenarios typically required a factor of two higher bit rates for maintaining a certain fixed PSNR. The same tendencies were observed in terms of the compression ratio in Figure 3.13. As expected from the corresponding PSNR versus bit-rate curves, the H.264 codec's compression ratio performance curves are near-linear on this log-log scale graph, resulting in a predictable PSNR versus compression ratio relationship.

3.6.4 Comparative Study

Our investigations presented in this section were conducted in order to compare the MPEG-4 codec's performance to the characteristics of the H.264 codec. Both the MPEG-4 and the H.264 experiments were conducted at 10 and 30 frames/s and at a wide range of bit rates. We varied the quantiser step sizes for the sake of attaining a given target bit rate in our experiments. Thus the quantisers having indices ranging from 0 to 31 were set approximately in each of the simulations. Figure 3.14 shows a comparison of the MPEG-4 and H.264 video codecs for the "Miss America" video sequence in terms of the achievable image quality expressed in PSNR versus bit-rate (Kbit/s). As can be seen from the graphs, the performance of the H.264 codec is significantly better than that of the MPEG-4 arrangement. Comparing the MPEG-4 codec's performance at 30 frames/s to that of the H.264 scheme

at a frame rate of 10 frames/s produces interesting revelations. The results show that by replacing the MPEG-4 codec by the H.264 scheme allows the frame rate to increase from 10 to 30 frames/s without any significant change in the bit-rate requirements or without loss of image quality. This was achieved by invoking more sophisticated and hence higher-complexity signal processing, resulting in a better image quality perception.

Figure 3.15 provides a comparison of the MPEG-4 and H.264 video codecs in a different context, expressed as image PSNR versus comparison ratio. Again, as can be seen from the corresponding graphs, the H.264 codec exhibits a higher compression performance than the MPEG-4 codec.

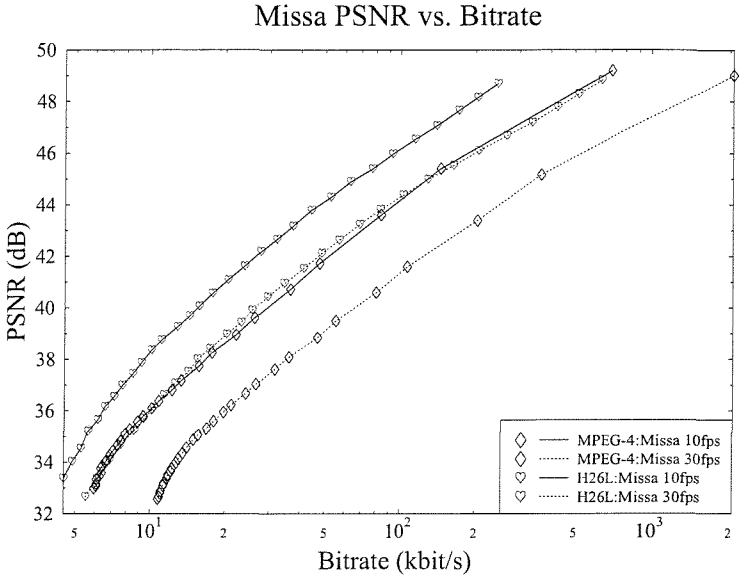


Figure 3.14: Image Quality (PSNR) versus coded bit rate, for the MPEG-4 and the H.264 codecs for "Miss America" colour QCIF- resolution video sequence scanned at 10 and 30 frames/s.

Let us now verify the above findings, which we arrived at on the basis of the "Miss America" sequence, also using the "Suzi" sequence. The experimental conditions were unchanged, we simply replaced the "Miss America" video sequence by the "Suzi" sequence. The corresponding results are plotted in Figures 3.16 and 3.17 in terms of PSNR versus bit rate and versus compression ratio, respectively. As expected, the results are similarly to those of the Miss America experiments. However, there is less similarity between the MPEG-4 curves recorded at 10 frames/s and the H.264 curves generated at 30 frames/s. More specifically, the H.264 codec operating at 30frames/s outperforms the MPEG-4 scheme transmitting at 10 frames/s, at higher bit rates.

The final set of experiments provided in this chapter was based on replacing the "Miss America"

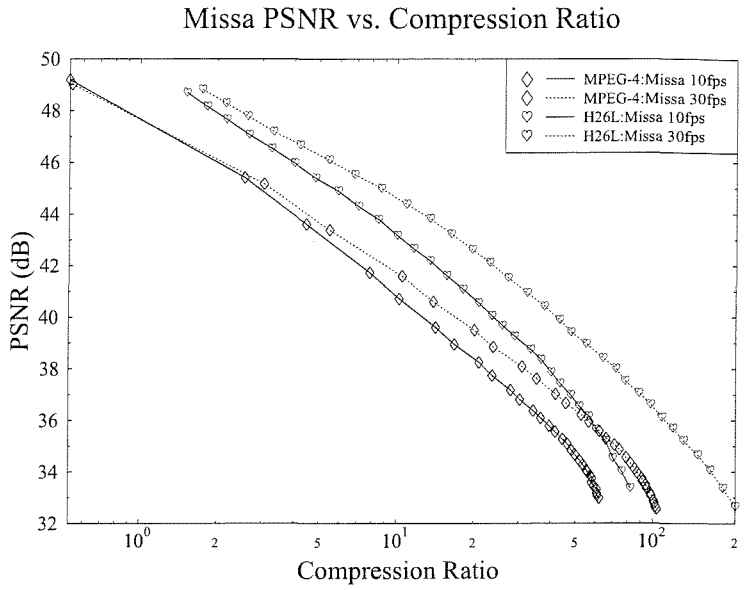


Figure 3.15: Image Quality (PSNR) versus compression ratio, for the MPEG-4 and the H.264 codecs for "Miss America" colour QCIF- resolution video sequence scanned at 10 and 30 frames/s.

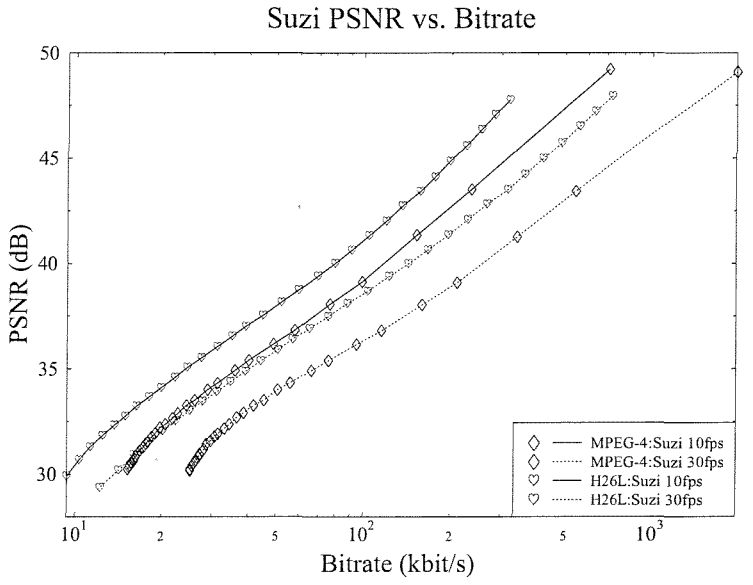


Figure 3.16: Image Quality (PSNR) versus coded bit rate, for the MPEG-4 and the H.264 codecs for "Suzi" colour QCIF- resolution video sequence scanned at 10 and 30 frames/s.

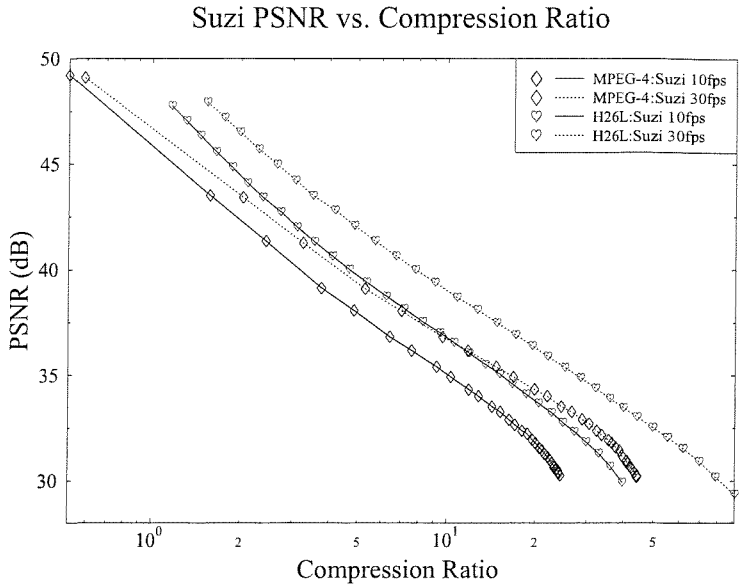


Figure 3.17: Image Quality (PSNR) versus compression ratio, for the MPEG-4 and the H.264 codecs for "Suzi" colour QCIF- resolution video sequence scanned at 10 and 30 frames/s.

video sequence by the "Foreman" video sequence. As before, our experiments were carried out at both 10 and 30 frames/s. The corresponding performance curves are shown in Figure 3.18 and Figure 3.19. Since the amount of image fine detail is quite different for the "Miss America" and "Foreman" video sequences, the image quality (PSNR) performance was slightly degraded at a given coded bit rate. Observe at that the curves seen in Figure 3.18 do not cross each other, which implies that the H.264 codec in operating at both 10 and 30 frames/s achieves a the higher picture quality, than the MPEG-4 codec. Similar conclusions are valid also for the compression ratio.

3.6.5 Summary and Conclusions

In these two chapters, we described the differences between the MPEG-4 and H.264 video codecs, and a range of comparative performance curves were plotted. The differences between the two codecs in terms of standards, functionalities and coding algorithm were described. From the range of algorithmic differences, prediction methods (spatial prediction and motion compensation), transform coding and entropy coding have the greatest effect to the coding efficiency. A variety of comparative performance curves with various video sequences were plotted, demonstrating how the performance of the H.264 codec varied after some of its parameters were altered.

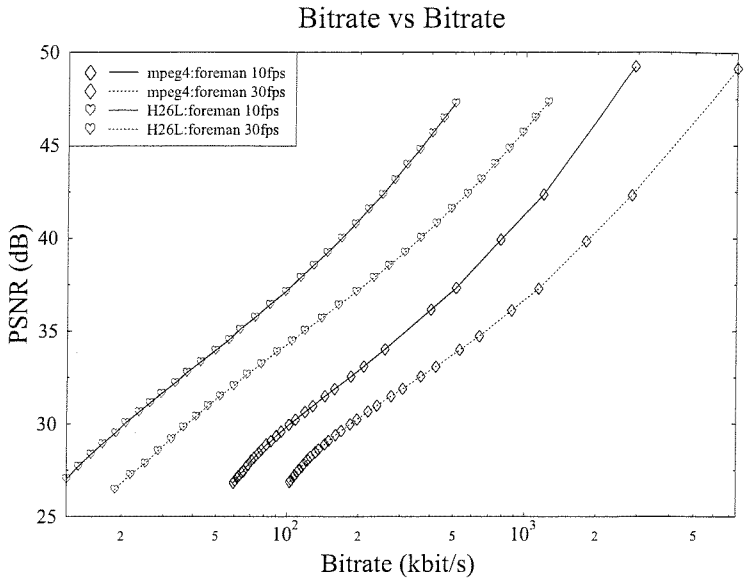


Figure 3.18: Image Quality (PSNR) versus coded bit rate, for the MPEG-4 and the H.264 codecs for "Foreman" colour QCIF- resolution video sequence scanned at 10 and 30 frames/s.

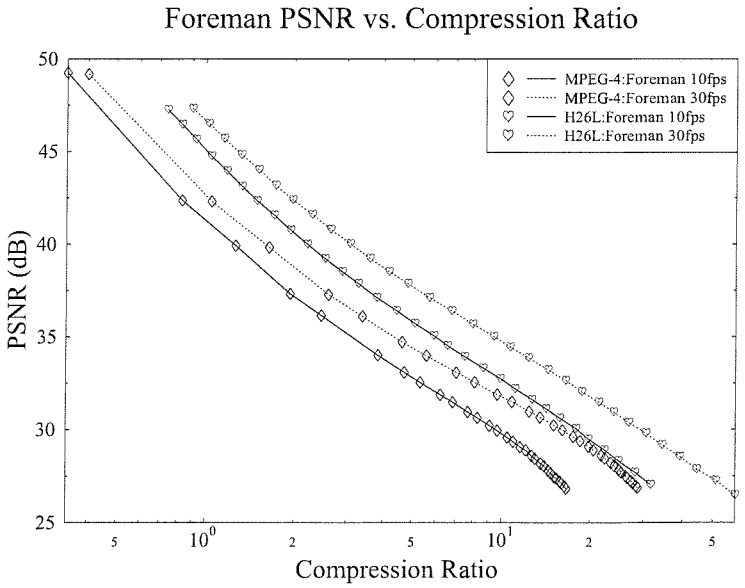


Figure 3.19: Image Quality (PSNR) versus compression ratio, for the MPEG-4 and the H.264 codecs for "Foreman" colour QCIF- resolution video sequence scanned at 10 and 30 frames/s.

Chapter 4

MPEG-4 Bitstream and Bit-Sensitivity Study

4.1 Motivation

The MPEG-4 video coding standard was briefly introduced in Chapter 2. As seen in Figure 4.1, the encoded video bitstream can be represented in terms of a number of hierarchical layers, where each higher-order layer contains lower-order layers, which in turn contain further lower-order layers. This figure will be discussed in more detail during our further discourse. We note however that it is rather difficult to visualise the corresponding bitstream in a well presented, highly organised manner, because the MPEG-4 format is far more complicated than the MPEG-1, MPEG-2 as well as the ITU-T H.263 bitstream formats. Nonetheless, in order to ease this task, the MPEG-4 bitstream structure has been partitioned into several categories.

In Chapter 2, we outlined the basic foundations for the MPEG-4 Visual standard's coding profiles and algorithms. In Chapter 3, we provided a comparative study in comparison to H.264. This chapter attempts to outline the MPEG-4-encoded hierarchical bitstream structure, which forms the basis of our error resilience study in Section 4.7.

4.2 Structure of Coded Visual Data

The MPEG-4 encoded visual information may assume several different types, such as video information, still texture information, 2D mesh information or facial animation parameter information [159].

Synthetic objects such as video data, still image data, mesh data and animated data and their attribution are structured in a hierarchical manner for the sake of supporting both bitstream scal-

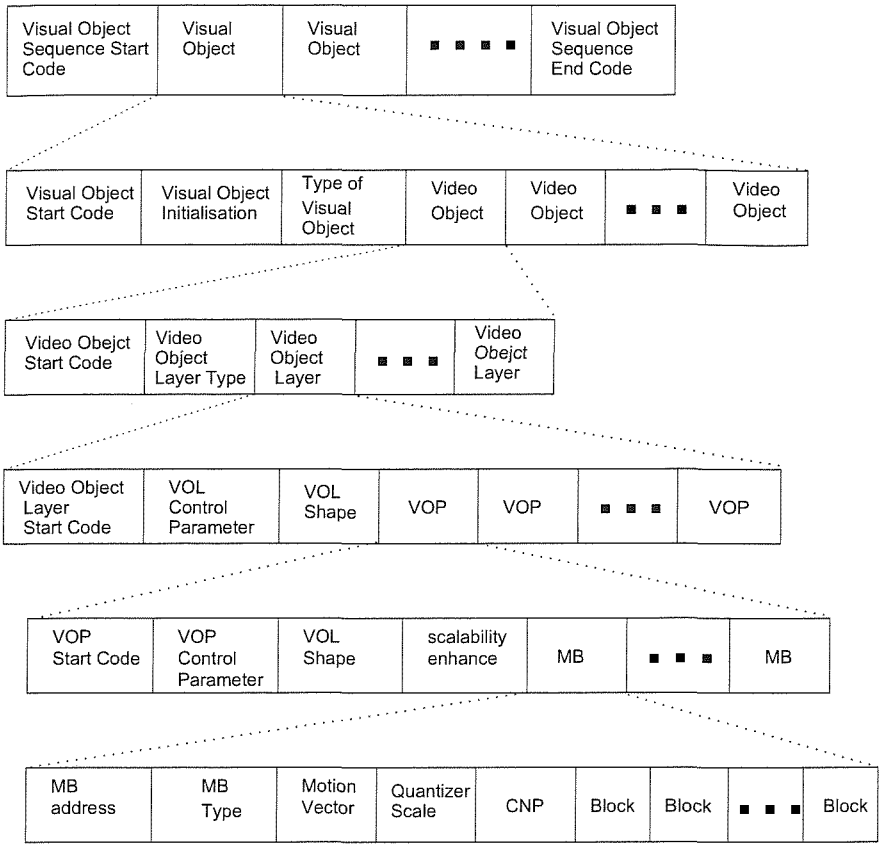


Figure 4.1: MPEG-4 simplify bitstream structure

ability and object scalability. The system-oriented part of the MPEG-4 specifications outlines the philosophy of spatial-temporal scene composition, including normative 2D/3D scene graph nodes and their composition supported by the Binary Interchange Format Specification (BIFS) [159, 169]. At this level, synthetic and natural object composition relies on systems using subsequent(non-normative) rendering performed by the application to generate specific pixel-oriented views of the models.

□ *VIDEO DATA*

The MPEG-4 codec facilitates the employment of the so-called embedded encoding model. Specifically, a low-quality low-rate mode referred to as base-rate mode may be used by the decoder, if not all the encoded bits reach the remote receiver. For the sake of ensuring that both the encoder's local decoder and the remote decoder use the same reconstructed video buffer contents during the process of motion compensation, the encoder also refrains from using the so-called enhancement bits during this process. However, if the remote decoder does receive the enhancement bits, they allow the improvement of decoded video quality. If there is only a single encoded bitstream, the coded video data is referred to as a non-scalable video bitstream. By contrast, if there is both a base and an enhancement

bitstream, the encoded video bit pattern is termed a scalable video bitstream. In MPEG-4, video object layers are often differentiated as base layer and enhancement Layer.

□ *STILL TEXTURE DATA*

Visual texture coding was designed for maintaining a high perceptional quality while maintaining a low rate transmission rate and for rendering of texture under widely varied viewing conditions, in particular, when encoding and for manipulating 2D/3D synthetic scenes. Still texture encoding facilitates the multi-layer representation of luminance, colour and shape. This supports progressive transmission of the texture for image quality improvements, when more information is received by a terminal.

□ *MESH DATA*

Coded mesh data was introduced in [144] and it was well presented by Tekalp and Ostermann in [170]. In Chapter 2, we discussed that the MPEG-4 is an object-based multimedia compression standard, which allows for encoding of different audio-visual objects. The Mesh data in this category are some of the synthetic visual objects supported by MPEG-4, namely animated faces and animated arbitrary 2D uniform and Delaunay meshes [170], the mesh data consists of a single non-scalable bitstream. This bitstream defines the structure and motion trajectory of the 2D mesh. The texture of the mesh has to be encoded as a separate video object.

□ *FACE ANIMATION PARAMETER DATA*

The coded face animation parameter data also consists of a single non-scalable bitstream. It defines the animation of the face model of the decoder. Face animation data is structured in a specific format outlined by the standard, which invoked downloadable face models and their animation controls, as well as a signal layer of compressed face animation parameters used for remote manipulation of the face model. The face constitutes a node in a scene graph that includes the face geometry ready for rendering. The shape, texture and facial expressions of a synthetic face are generally controlled by the bitstream containing Facial Definition Parameter (FDP) sets [144] or Facial Animation Parameter (FAP) sets [144].

However, in this treatise the focus of research is related to the encoding of video data. Therefore, still texture data, 2D mesh data and face animation parameter data will not be discussed further in this discourse.

As mentioned above, the structure of the bitstream is organised into separate video parameter layers.We have seen in Figure 4.1, that it consists of visual sequence information, visual objects, video objects, the video object layer ,video object plane, macroblock and video block data.

4.3 Visual Bitstream Syntax

4.3.1 Start Codes

Start codes are constituted by a specific bit patterns that may be mimicked only with a low probability by the video stream. Each start code seen at the various layers of Figure 4.1 consists of a start code prefix followed by a start code value. The start code prefix is a string of 23 bits zero value followed by 1 bit value one. The start code prefix is thus the bit string '00000000000000000000001'. The start code value is an eight-bit integer, which identifies the type of the start code concerned. In the MPEG-4 bitstream syntax, the majority of the start codes has a single legitimate start code value. However, the video object start code and the video object layer start code may be represented by numerous different start code values, where each video object start code marks the beginning of a new video object layer hosting several Video Object Planes (VOPs), as seen in Figure 4.1.

All start codes are byte-aligned. This is achieved by inserting padding bits, namely a bit having a zero value and then, if necessary, by inserting bits having a logical one value before the start code prefix, such that the first bits of the start code prefix becomes the first (most significant) bit of the byte concerned [159]. Table 4.1 explicitly defines the start code values for all start codes used in the video bitstream.

- The Visual Object Sequence is the highest-order syntactic structure of the coded visual bitstream in Figure 4.1. A visual object sequence commences with a visual object start code, which is followed by a one or more visual objects. The visual object sequence is terminated by a visual object sequence end code.
- A Visual Object sequence commences with the visual object start code of Table 4.1, which is followed by profile and level identification, and a visual object identifier and by a video object, a still texture object, a mesh object, or a face object which were introduced in Section 4.2.
- A Video Object commences with a video object start code, that is followed by one or more video object layers, as seen in Figure 4.1

4.4 Introduction to Error-Resilient Video Encoding

The MPEG-4 aims for achieving error resilience in order to access image or video information remotely, while communicating over hostile transmission media [171]. In particular, it is important to support access to audio and video information via wireless networks at low bit-rates.

The so-called error resilience tools developed for the MPEG-4 codec can be divided into three

Start Code Type	Start Code Value (hexadecimal)
Video object start code	00 to 1F
Video object layer start code	20 to 2F
Visual object sequence start code	B0
Visual object sequence end code	B1
User data start code	B2
Group of VOP start code	B3
Video session error code	B4
Visual object start code	B5
VOP start code	B6
Face object start code	BA
Face object plane start code	BB
Mesh object start code	BC
Mesh object plane start code	BD
Still texture object start code	BE
Texture spatial layer start code	BF
Texture SNR layer start code	C0

Table 4.1: Start code values expressed in hexadecimal format

classes, namely in resynchronization, data recovery and error concealment algorithms. It should be noted that these categories are not unique to MPEG-4, they have been used by numerous researchers working in the area error resilience for video [172].

4.5 Error-Resilient Video Coding in MPEG-4

This section discusses the error resilient encoding measures employed in the MPEG-4 standard and describes the different encoding tools adopted by the standard for enabling robust video communication over noisy wireless channels. When compressed video data is transmitted over a wireless communication channel, it is subjected to channel errors in the form of either single bit errors or burst of errors or both [173]. In order to render the video codec more resilient to channel-induced quality degradations, various forward error correction (FEC) may be employed by the encoder for protecting the bitstream prior transmission to the decoder [174, 175]. At the decoder, these FEC codes are then used for correcting the transmission errors which employs the following processing stages [159]:

- Error detection and localisation;
- Resynchronisation;
- Data recovery;
- Error concealment.

Before any error concealment technique can be applied at the decoder, it is necessary to ascertain, whether a transmission error has occurred. FEC techniques can also be used for error detection [174]. In the context of a typical block-based video compression technique that uses motion compensation and DCT, a number of various error detection techniques may be invoked, such as the motion vectors are out of range, an invalid VLC table entry is encountered, the number of DCT coefficients in a block exceeds 64. When one or more of these conditions is detected in the process of video decoding, the video decoder flags an error and activates the corresponding error-handling procedure. Due to the specific nature of the MPEG-4 video compression algorithm, which relies Variable Length Coding (VLC), the location where the decoder detects an error is not the same, where the error has actually occurred. As shown in Figure 4.2, these events may be an undetermined distance away from each other. Naturally, an error event may also induce the loss of synchronisation with the encoder.

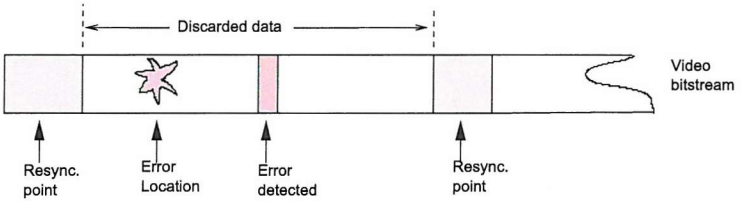


Figure 4.2: Stylised illustration of error event and its detection. It is usually impossible to detect the exact location of error occurrence location. Hence, all the data between the two corresponding resynchronisation points may have to be discarded.

While constructing the encoded bitstream, the encoder inserts unique resynchronisation markers into the bitstream at approximately equally-spaced intervals. Upon detecting of an error, the decoder searches through the incoming bitstream, until the next resynchronisation maker is found. Once this resynchronisation marker is found, the decoder reestablishes synchronisation with the encoder. Hence, all data that corresponds to the macroblocks between the resynchronisation points has to be discarded, since the perceptual effects of displaying an image reconstructed from erroneous data are typically highly annoying.

After synchronisation has been re-established, the MPEG-4 codec's data recovery tools attempt to recover the data that otherwise would be lost. More specifically, data recovery techniques such as

reversible decoding [176] enable the decoder to salvage some of the data between the two resynchronisation points by decoding the bitstream in both the forward and reverse direction. The philosophy of this method will be described in greater detail in Section 4.6.

Error concealment is a further important component of any error-resilient video codec [177]. Similarly to the error resilience tools discussed above, the efficiency of an error concealment strategy is highly dependent on the performance of the resynchronisation scheme used. More specifically, if the resynchronisation method is capable of accurately localising the error, then the error concealment becomes more efficient. In low-bitrate, low-delay applications the above-mentioned resynchronisation scheme provides acceptable results in conjunction with a simple concealment strategy, such as the method of copying blocks from the corresponding video frame fraction of the previous frame. Upon recognising the need to provide enhanced error concealment capabilities, the MPEG-4 standardisation body has developed an additional error resilient mode that further improves the ability of the decoder to localise transmission error. Specifically, this approach utilises data partitioning by separating the motion and texture related parts of the bitstream. This approach requires that a second resynchronisation marker be inserted between the motion and texture information related bits. If the texture information is lost, this approach utilises the motion information for concealing these errors. More explicitly, the corrupted texture information is discarded, but the motion vectors still can be used for motion compensation on the basis of the previous decoded VOP instead of the current one.

4.6 Error Resilience Tools in MPEG-4

A number of measures referred to in MPEG-4 parlance as 'tools' have been incorporated into the MPEG-4 video encoder for the sake of rendering it more error-resilient. These tools include various techniques such as:

- Video packet resynchronisation [150];
- Data Partitioning (DP) [178];
- Reversible VLCs (RVLCs) [176];
- Header extension codes (HECs) [150].

which are discussed below in a little more detail.

4.6.1 Resynchronisation

As mentioned in Section 4.5, an erroneous video bitstream is expected to arrive at the decoder, when compressed video data is transmitted over error-prone communication channels. The decoder might lead to the loss of synchronisation with the encoder, when an error-infested bitstream is received. Resynchronisation tools, as the terminology implies, attempt to reestablish synchronisation between the encoder and the decoder, when transmission errors have been detected. Generally, the data between the synchronisation point prior to the error and the first point where synchronisation is reestablished, is discarded. If the resynchronisation approach is successful in determining the amount of data discarded by the decoder, then the ability of other types of tools which recover data and/or conceal the effects of errors is greatly enhanced.

The resynchronisation approach adopted by MPEG-4, referred to above as the video packet resynchronisation approach, is similar to the Group of Blocks (GOBs) structure exploited by the ITU-T standards of H.261 and H.263 [51] codecs for reestablishing synchronisation after encountering transmission errors. Let us hence briefly highlight the resynchronisation procedure used by these codecs [179]. In these standards a GOB is defined as one or more rows of macroblocks (MBs) within a specific video frame. At the commencement of a new GOB a GOB header is incorporated into the bitstream. This header information contains a GOB start code, which is different from a picture start code, and allows the decoder to locate this GOB. Furthermore, the GOB header contains information which allows the decoding process to be restarted by resynchronising the decoder with the encoder, while resetting all predictively coded data. Figure 4.3 shows the GOB numbering scheme for H.263 for QCIF video format.

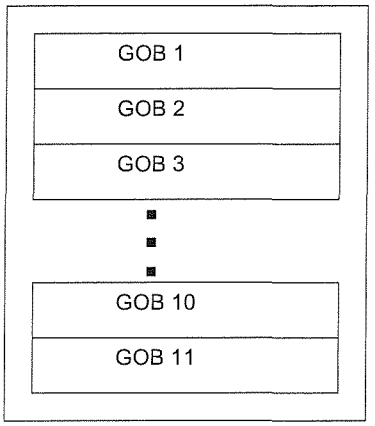


Figure 4.3: H.263 GOB numbering for a QCIF-resolution 176 x 144-pixel image.

Again, while constructing the MEPG-4 bitstream, the encoder inserts unique resynchronisation markers into the bitstream at approximately equally spaced intervals as seen in Figure 4.4. Upon

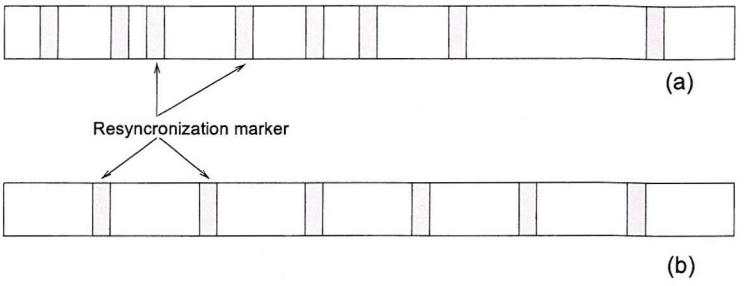


Figure 4.4: (a) Position of the synchronisation markers in the bitstream for a baseline H.263 encoder showing the GOB headers. (b) Position of the resynchronisation markers in the bitstream of an MPEG-4 encoder generating equal-length video packets.

detecting of an error, the decoder searches through the received bitstream, until the next resynchronisation maker is found. Once this resynchronisation marker is found, the decoder reestablishes synchronisation with the encoder. Hence, all data that corresponds to the macroblocks between the resynchronisation points has to be discarded, since the effects of displaying an image reconstructed from erroneous bits may inflict highly annoying visual artifacts.

After synchronisation has been reestablished, the MEPG-4 data recovery tools attempt to recover data that in general would be lost. As mentioned earlier, data recovery techniques, such as reversible decoding [180] enable the decoder to salvage some of the data between the two resynchronisation points, since the received bitstream can be decoded in both the forward and reverse direction, as it will be described in Section 4.6.4.

The GOB start coded aided resynchronisation is based on spatial resynchronisation. More explicitly, once a particular macroblock location is reached during the encoding process, a resynchronisation marker is inserted into the bitstream. A potential problem associated with this approach is that since the encoding process results in a variable rate, these resynchronisation markers will most likely be unevenly spaced throughout the bitstream. Therefore, certain portions of the scene, such as highly motion-active areas, will be more susceptible to errors, and these will also be more difficult to conceal.

MPEG-4 provides a similar method of resynchronisation to that of the H.263 codec with one important difference, however the MPEG-4 encoder is not restricted to insert the resynchronisation markers only at the beginning of each row of macroblocks. Instead, The encoder has the option of dividing the image into video packets, as seen in Figure 4.4 each constituted by an integer number of consecutive macroblocks. Hence, as portrayed in Figure 4.4, it forms periodic resynchronisation markers throughout the bitstream. If the number of bits contained in the current video packet exceeds a predetermined threshold, then a new video packet is created at the start of the next macroblock. Thus, in the presence of a short burst of transmission errors, the decoder has the ability to promptly localise the error within a few macroblocks.

A resynchronisation marker is used for distinguishing the start of a new video packet, as seen in Table 4.1 This marker is distinguishable from all possible VLC codewords as well as from the VOP start code. Header information is also provided at the start of a video packet as seen in Table 4.1. The information necessary for restarting the decoding process is contained in this header and includes the macroblock index of the first macroblock contained in this packet as well as the quantisation parameter necessary for decoding the first macroblock. The macroblock index provides the necessary spatial resynchronisation, while the knowledge of the quantisation parameter allows the differential decoding process to be resynchronised.

4.6.2 Data Partitioning

Figure 4.5 depicts the organisation of the video data within a packet for the MPEG-4 video compression scheme using no data partitioning. The video packet commences with a resynchronisation marker, followed by the macroblock index and the Quantiser Parameter (QP) and finally by the combined motion and DCT data. Note that the combined motion and DCT data in general contains a lot more data than the three preceding header fields.

To elaborate a little further, Figure 4.6 shows the syntactic elements of each MB in the case of the MPEG-4/H.263 video encoder. This data is repeated for all macroblocks contained in the packet. The subscripts indicate the MB index. The parameter `COD` is a 1-bit field used for indicating whether a certain MB is encoded or skipped, By contrast `MCBPC` is a variable-length field used for indicating, first of all the encoding mode of a MB, namely the `INTRA, INTER, INTER4V` [1] and `INTRA+Q` [2] coded modes. Secondly, `MCBPC` also signals, which of the two chrominance blocks of the MB is coded. Furthermore, `DQUANT` is an optimal 2-bit fixed-length field used for indicating the incremental modification related to the quantiser index with respect to the previous MB's quantiser index. The parameter `CBPY`[3] is a VLC that indicates which of the four blocks of the MB are encoded. The MVs are transmitted in the form of the motion vector differences by a VLC and finally, the DCT data comprises the 64 DCT coefficients actually encoded, which are transmitted after zig-zag scanning and run-length encoding, using the VLC table of the MPEG-4 standard [144].

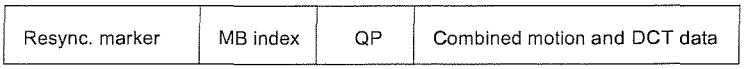


Figure 4.5: Organization of the data within an MPEG-4 video packet

In MPEG-4, the data partitioning mode separates the data of a video packet into a motion-selected

[1]The acronym implies the transmission of four motion vectors per macroblock. Each luminance block in the MB is allowed to have its own motion vector. This allows higher flexibility in obtaining the best match for the MB.

[2]The quantiser parameter is modified for this MB with respect to the previous MB.

[3]coded block pattern for luminance

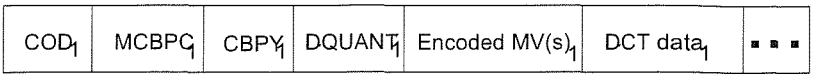


Figure 4.6: Bitstream components for each macroblock within the MPEG-4 video packet.

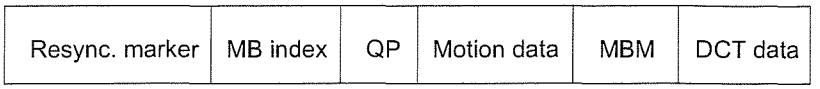


Figure 4.7: MPEG-4 Bitstream organisation using data partitioning for separating the motion and DCT data.

part and a texture-selected part separated by a unique motion boundary marker (MBM), as shown in Figure 4.7. Compared to Figure 4.5, the motion- and DCT-related parts are now separated by an MBM. All the syntactic elements that have motion-related information are placed in the motion partition, and all those related to DCT data are placed in the DCT data partition. Figure 4.8 shows the bitstream structure after reorganisation of the motion part, while Figure 4.9 shows the bitstream structure of the DCT part. Note that the elements in both Figures 4.8 and 4.9 have been defined in the context of Figure 4.5.

When a transmission error is detected in the motion section of Figure 4.8, the decoder declares an error and replaces all the MBs in the current packet with skipped blocks, until the next resynchronisation marker is found. When a transmission error is detected in the texture section, but no errors are detected in the motion section, the corresponding MB's motion vectors are used during motion compensation for reconstructing the transmitted MBs using the block DCT data, since the texture part is discarded and the decoder resynchronises, when detecting the next resynchronisation marker. If no error is detected in the motion and texture sections, but the resynchronisation marker is not found at the end of decoding, an error is declared. In this case, only the texture part of the MBs contained in the current packet is discarded. As reported in [181], data partitioning provides improved video quality under a different error conditions.

4.6.3 Reversible Variable-Length Codes

Reversible variable-length codes (RVLCs) [180] constitute a specific class of VLCs that can be uniquely decoded in both directions. This advantageous property is a consequence of the fact that no codeword may constitute a prefix or postfix of another codeword, which allows their unambiguous decoding in

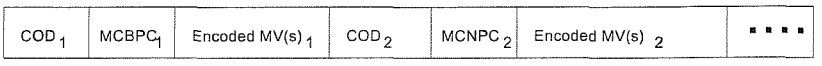


Figure 4.8: Bitstream components of the motion data

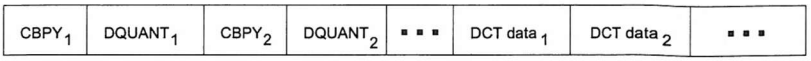


Figure 4.9: Bitstream components of the DCT data.

both forward and backward directions. When the decoder detects an error, while decoding in the forward direction, it will immediately look for the next resynchronisation marker and start decoding in the reverse direction, until an error is found. Based on the position of the detected errors in the forward and backward directions, the decoder is likely to be able to locate the error within a more limited region in the bitstream and hence recover additional data. As illustrated in Figure 4.10, only the data contained in the shaded region is discarded. It is important to mention that if RVLCs were not invoked, all the data between the two consecutive resynchronisation markers seen in Figure 4.10 would have to be discarded. In MPEG-4, RVLCs have been shown to provide a significant gain in subjective video quality terms in the presence of channel errors by enabling more data to be recovered [176, 180]. In MPEG-4, when the data partitioning mode is employed, the DCT coefficient information is encoded by means of RVLCs.

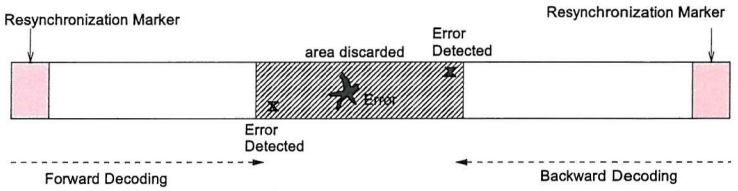


Figure 4.10: Reversible variable-length codes enable the recovery of data by allowing decoding to take place both in the forward and reverse directions.

4.6.4 Header Extension Code

In video compression, the Header Extension Code (HEC) is the most important information required by the decoder for decoding video bitstream. The header extension code allows the introduction of duplicate copies of important picture header information sequences in the video packets. The HEC is a single-bit fixed, which indicates the presence of additional resynchronisation information, including information such as spatial resolution of the frame, its temporal location and the encoding mode of the frame considered. This technique has the potential of reducing the number of discarded video frames significantly in error-prone environments. This additional information is made available for the decoder, when the VOP header has been corrupted. This mode is supported in the MPEG-4 codec, and a similar tool is included also in the RTP[4] payload specification of the H.263 version 2 standard, as defined in Request for Comments 2429 [182].

[4]Real Time Payload [151]

In conclusion, wireless communication networks do not always guarantee error-free transmission. Hence we discussed a range of error resilient coding methods that facilitate reliable video communications in error-prone environments. Many of these error resilient methods have now been incorporated by the various video standards, such as H.263 and MPEG-4.

4.7 MPEG-4 Bit-Sensitivity Study

4.7.1 Objectives

The purpose of the error sensitivity investigations in this section is to use the knowledge gained in order to design an MPEG-4 based video videophone system for a mobile environment with the following aims:

- Better error resilience than that of the original MPEG-4 standard
- A data stream that conforms to the MPEG-4 standard or is amenable to conversion with a simple transcoder.
- A reconfigurable system so that it can adjust to different channel conditions.

4.7.2 Introduction

In this section, we embark on presenting the results of our MPEG-4 bit-sensitivity study. Over the years a number of different techniques were used in the literature for quantifying the bit error sensitivity of video codecs. The outcome of the investigations to be conducted will highly depend on a number of factors, such as the video material used, the target bitrate of the video codec, and on the averaging procedure employed. Our experiments were carried out using the MPEG-4 encoder and decoder with the error resilient mode disabled. This implies that whenever a bit was corrupted, the MPEG-4 codec did not attempt to correct or conceal the error effects. We purposely selected this mode in our experiments so that when a bit was corrupted, the corresponding video artifacts and video degradation were not obfuscated.

Here, we propose a simplified objective video quality measure based bit-sensitivity evaluation procedure, which attempts to examine all the major factors influencing the sensitivity of the MPEG-4 encoded bits. Specifically, the proposed procedure takes into account of the MPEG-4 parameters contained in the entire bit stream, which also giving the effects of losing synchronisation as well as to the error propagation effects.

At this stage, we assume that readers are familiar with the MPEG-4 standard at a depth, which

was provided by Chapter 2. The aim of our MPEG-4 bit sensitivity study was to quantify the average PSNR degradation caused by each erroneously decoded video codec parameter in the bitstream, so that appropriate channel coding based error protection can be assigned to each parameter.

4.7.3 Simulated Coding Statistics

Video-coding simulations were conducted in order to find the relative importance of the different coding parameters or codewords in the MPEG-4 bit-stream.

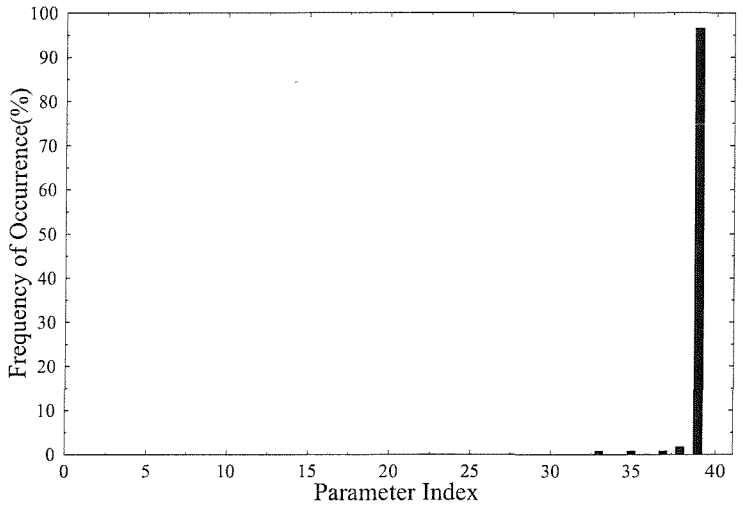


Figure 4.11: Probability of occurrence for the various MPEG-4 parameters for the "Miss America" QCIF video sequence encoded at 30 frame/s and 1.15Mbit/s using the parameter indices listed in Table 4.2.

The results presented in this section quantify the relative frequency of occurrence for various coding parameters using the MPEG-4 codec. Specifically, Figures 4.11 to 4.14 show the probability of occurrence for the various MPEG-4 parameters in the context of the Video Objects (VO), Video Object Layer (VOL), Video Object Plane (VOP) and Macroblock (MB) representations, the respectively statistics figures shown in Figures 4.11 to 4.14 were derived for the same video clip, namely, for "Miss America", but the results are presented on a different vertical scales, since some of the coding parameters have a relatively small probability of occurrence compared to others. Therefore results are presented on various vertical scales and the duplications of results is avoided in Figures 4.11 to 4.14. In addition, Table 4.2 tabulated the parameter name corresponding to the parameter indexes in the Figures4.11 to 4.14.

Index	Descriptions
1	VO_start_code
2	markers_bit
3	VO_id
4	VOL_START_CODE
5	VOL_Id
6	VOL_RandomAccessible
7	VOL_IsObjectLayerIdentifier
8	VOL_ControlParameters
9	VOL_Shape
10	VOL_TimeIncrement_Resolution
11	VOL_FixedVopRate
12	VOL_Width
13	VOL_Height
14	VOL_OBMCDisable
15	VOL_SpriteUsage
16	VOL_Not_8_Bit
17	VOL_QuantType
18	VOL_CompleEstimatDisable
19	resyn_marker_flag
20	VOL_DataPart_Enable
21	VOL_Scalability
22	VOP_START_CODE
23	VOP_Prediction_Type
24	VOP_timemovulo
25	VOP_timeinc
26	Width_buffer
27	VOP_IntraDC_Vlc
28	VOP_Interlaced
29	VOP_Intra_Quantizer
30	VOP_Inter_Quantizer
31	VOP_FCodeFor
32	PutMCBPC_Intra
33	PutMCBPC_Inter
34	ACpred_flag
35	PutCBPY_index
36	PutDCsize_lum
37	PutDCsize_chrom
38	PutMV
39	ALL_TCOEFF

Table 4.2: Reference to the MPEG-4 coding parameter index as shown in Figure 4.11 to 4.14

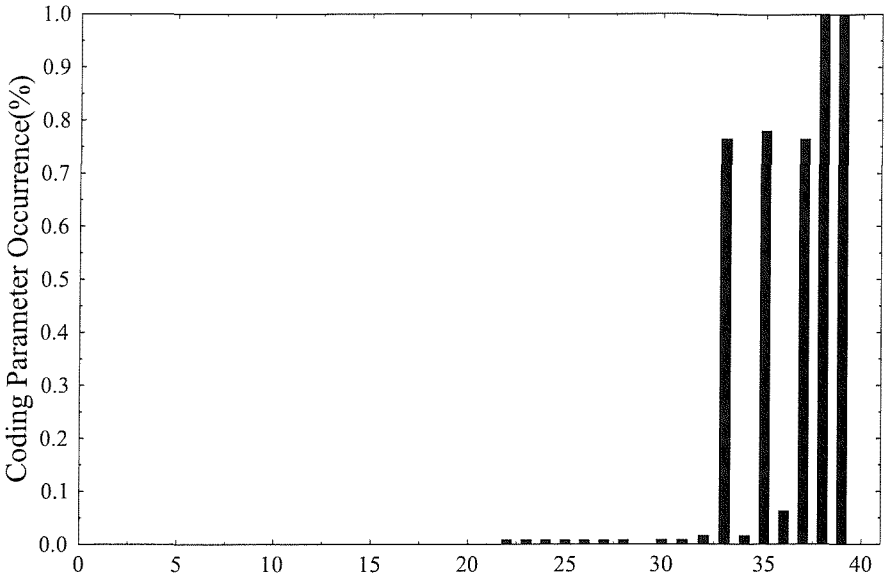


Figure 4.12: Probability of occurrence for the various MPEG-4 parameters for the "Miss America" QCIF video sequence encoded at 30 frame/s and 1.15Mbit/s using the parameter indices listed in Table 4.2. This graph is mainly illustrates the probability in Macroblock MB.

We will now discuss results shown in Figure 4.14, which shows the probability of occurrences confined to the range below 0.00015%. It can be seen that all parameters having indices of 1-20 in Table 4.2 except for the `markers_bit` have the same probability of occurrence. The parameter `marker_bit` exhibits a doubled probability of occurrence in comparison to the other VO and VOL header parameters, since it appeared twice in the VOL header. This is a one-bit parameter that is set to logical 1 and assists in preventing emulation of start codes by the video-encoded bits.

Figure 4.13 shows the probability of occurrence for the bits mainly belonging to the Video Object Plane (VOP) seen between the indices of 22 to 31. It is pleasable obvious that the probability of occurrence of the parameters associated with the VOP is significantly higher compared to the parameters of the VO and associated VOL layer, since the VO and VOL header seen at indices of 1-18 only appear once at the very beginning of the entire video sequence. Likewise, in the MPEG-4 VOP coding, each VOP is associated with a video frame, which can be either INTRA- or INTER-frame coded. We used a total of 150 video frames of the Miss America sequence in our experiments. Therefore, there were 150 VOP and hence 150 VOP headers for the entire video bitstream. As a result, the probability of encoded bits being allocated to the various VOP header parameters is higher than that of the parameters in VO and VOL.

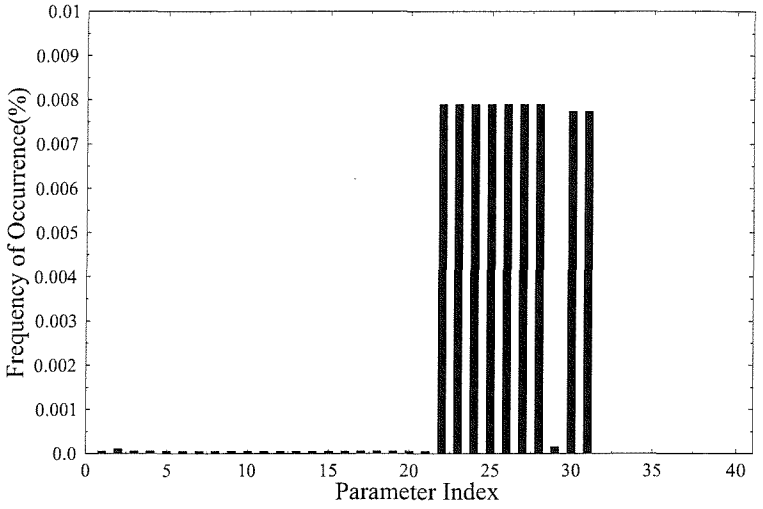


Figure 4.13: Probability of occurrence for the various MPEG-4 parameters for the "Miss America" QCIF video sequence encoded at 30 frame/s and 1.15Mbit/s using the parameter indices listed in Table 4.2. This graph is mainly illustrates the probability in Video Object Plane header.

Hence it is plausible that all VOP header parameters, except for the `VOP_Intra_Quantiser`, `VOP_Inter_Quantiser` and `VOP_FCode_Forward` parameters, we have the same probability of occurrence, since they appear once for every coded video frame. The `VOP_Inter_Quantiser` and `VOP_FCode_Forward` parameters have a slightly lower probability of occurrence, than most of the remaining VOP header parameters seen at indices , since these two parameters are only generated in inter-frame coding mode, while the `VOP_Intra_Quantiser` parameter has a significantly low probability of occurrence, since it is only required for intra-frame coding. There are only three intra-frames in our experiments, hence the low probability seen in Figure 4.13.

Let us now compare the relative frequency occurrence for the parameters incorporated by the MB layer. In the MB layer, there are even more high probability of occurrence parameters, since there were 99 MBs per video frame for the QCIF "Miss America" video sequence we used in these experiments. The discrete Cosine Transform (DCT) parameter has the highest probability of occurrence, exceeding 90%. In our experiments, the video sequence was encoded at a high bitrate of 1.15Mbit/s, hence at this relatively high bitrate it is not surprising to see that the DCT coefficients have the highest probability of occurrence. Let us now focus our attention on the effects of transmission errors in the next section.

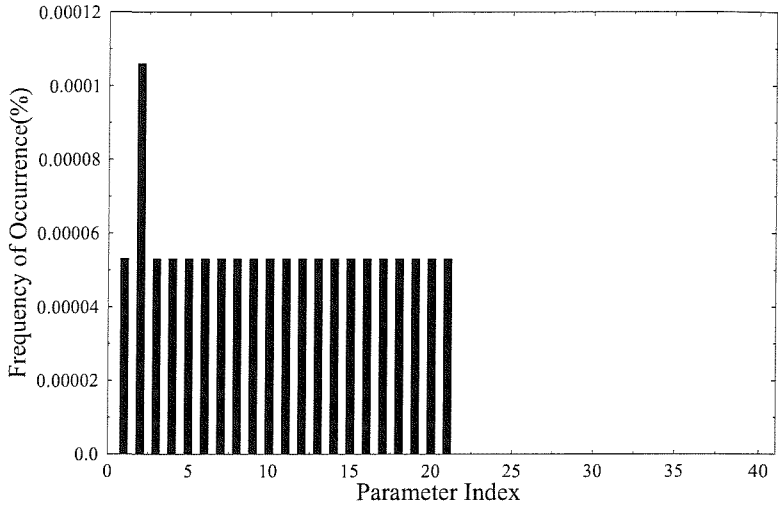


Figure 4.14: Probability of occurrence for the various MPEG-4 parameters for the "Miss America" QCIF video sequence encoded at 30 frame/s and 1.15Mbit/s using the parameter indices listed in Table 4.2. This graph is mainly illustrates the probability in Video Object (VO) and Video Object Layer (VOL) header.

4.7.4 Effects of Errors

In order to investigate the effects of transmission errors on the MPEG-4 parameters, a wide range of simulations were carried out using a 150-frame duration, QCIF-resolution "Miss America" sequence encoded at 1.15Mbit/s. Simulations were conducted mainly for two situations, namely for encountering errors in intra-coded error as well as in inter-coded frames. Almost all of the MPEG-4 parameters from the VO, VOL, VOP and MB domains of Table 4.2 were corrupted during our investigations. The PSNR degradations were recorded for each of the parameters in Table 4.2 under the following conditions

The effects of corruption on the MPEG-4 VO, VOL and VOP header parameters were somewhat ordous to estimate because these parameters contain important information related to the entire video sequence, to a particular video frame or to a relatively small MB.

Figure 4.15 shows the average PSNR degradations inflicted by the various MPEG-4 coding parameters, while their relative frequencies were characterised in Figures 4.11 to 4.14. It can be seen from Figure 4.15 that most of the corrupted parameters from the VO and VOL domains of Table 4.2 resulted on a high PSNR degradation. Furthermore, the video sequence cannot be decoded at all, when param-

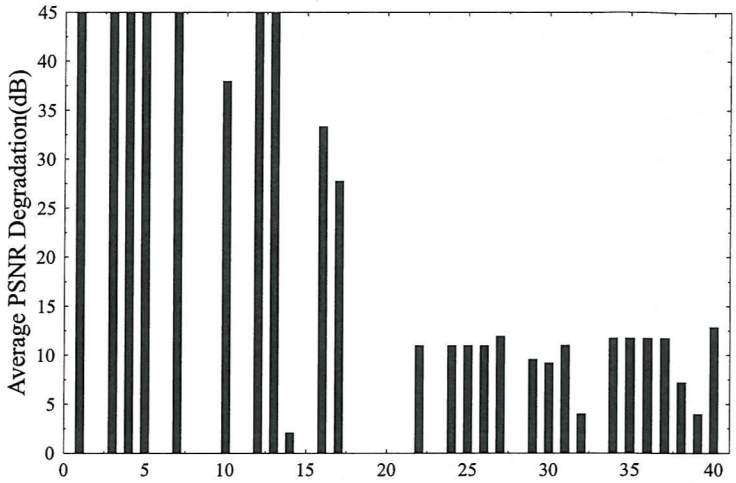


Figure 4.15: Average PSNR degradation for the various MPEG-4 parameters in the VO, VOL, VOP and MB domains for the "Miss America" QCIF video sequence encoded at 30 frame/s and 1.15Mbit/s using the parameter indices of Table 4.2.

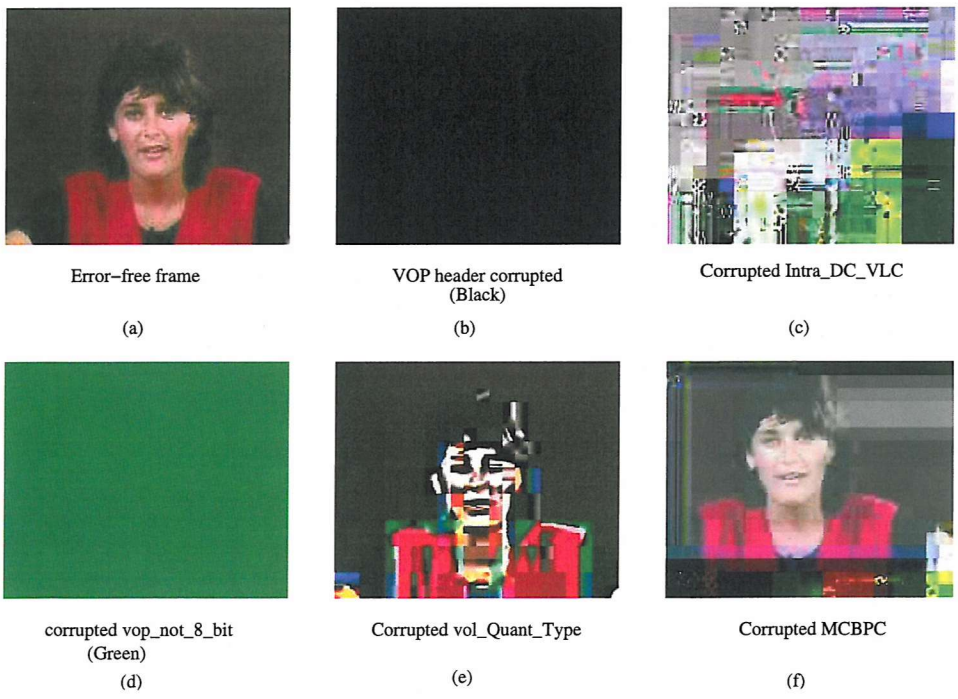


Figure 4.16: Image degradation due to the corruption of the various MPEG-4 coding parameters in an intra-coded frame.

eters such as the VO_start_code, VO_id, VOL_START_CODE, VOL_Id, VOL_IsObjectLayerIdentifier, VOL_Width and VOL_Height are corrupted. The decoder cannot even commence the decoding operations, or it would decoding, when those parameters are corrupted. This situation is indicated by setting the PSNR degradation to the average PSNR of the video sequence concerned, which implies the assumption of encountering a PSNR of 0 dB.

By contrast, some of the parameters, such as markers_bit, VOL_RandomAccessible, VOL_ControlParameter, VOL_Shape, VOL_FixedVopRate, VOL_SpriteUsage, VOP_ComplexEstimatDisable, resyn_marker_flag, VOL_DataPart_Enable, VOL_Scalability, VOP_Interlaced and PutMCBPC_Inter are more error resilient and hence even when these parameters were corrupted, they did not dramatically affect the picture decoding process. Thus no significant PSNR degradation was recorded. Since advanced functionalities, such as for example binary shape coding [138] and sprite coding [144] have not been included in our MPEG-4 studies, therefore the related parameters of Table 4.2 have been disabled during the decoding process.

When the parameter VOL_TimeIncrement_Resolution was corrupted, the PSNR degradation shown in Figure 4.15 became high, around 37 dB. As suggested by the acronym, when this parameter was corrupted, it imposed the wrong timing information during the decoding process. Specifically, the problems caused by the corruption of this parameters resulted in decoding the video frames in incorrect order, while some video frames might have been skipped as a consequence of using an incorrect time increment.

The flag VOL_Not_8_bit contained at index 16 in the VOL header of Table 4.2 is set, when the video pixel amplitude resolution is not 8 bits per pixel. Specifically, when this single bit parameter is set to binary '0' - the video pixel amplitude resolution is 8 bits per pixel. By contrast, when this single bit parameter is set to binary '1' - the video pixel amplitude resolution is not 8 bits per pixel. Hence the video decoder will search through the bitstream for another parameter referred to as 'bit_per_pixel'. This 'bit_per_pixel' parameter specifies the video pixel amplitude resolution in bits per pixel between 4 and 12. Since we used the 8 bits per pixel video pixel amplitude resolution in our experiments, therefore the parameter 'bit_per_pixel' is not necessary, hence it is absent in the bitstream. However, when the VOL_Not_8_bit parameter is corrupted, the decoder is unable to find the 'bit_per_pixel' parameter, therefore the video decoder does not know the video pixel amplitude resolution and as a result, the decoded video frame appeared as a block 'Green' frame, as depicted in Figure 4.16(d) .

When the VOL_Quant_Type parameter of Table 4.2 is corrupted, the entire video frame appears to be 'blocky', as illustrated in Figure 4.16(e). As expected, the PSNR degradation also becomes rather high.

When any of the parameters associated with the VOP header seen at indeces of Table 4.2 is

corrupted, it results in a lower PSNR degradation compared to those in the VO and VOL layer, as seen in Figure 4.15. The parameters `VOP_START_CODE`, `VOP_timemovulo`, `VOP_timeinc`, `VOP_Width_buffer`, `VOP_IntraDCVlcThr`, `VOP_Intra_Quantizer` result in similar PSNR degradations. The parameters appearing at the beginning of every video frame bitstream may be considered vital in the VOP header. Again, as the VO and VOL header found at indices of Table 4.2, the integrity of the VOP header is also important, although not for the entire video sequence, but for a complete video frame. When these bits were corrupted, the entire video frame was skipped, but the decoder remained capable of decoding the rest of the video sequence. In our experiments, we corrupted the first INTRA VOP header found at indices 22-31 of Table 4.2, and hence as expected, the decoder skipped the first frame, since there was no previous information, therefore it appeared to be a black frame, as demonstrated in Figure 4.16(b).

The parameter `VOP_IntraDCVlcThr` allows the codec to switch between two sets of VLC which can be used for the encoding of the intra DC coefficients. When this parameter is corrupted, decoder uses the wrong VLC index for decoding the intra-frame DC coefficients. The associated error effects are severe and since most of the MPEG-4 bitstream is encoded using VLCs, even a single bit error will detrimentally affect the achievable video quality. Figure 4.16(c) portrays an example, when the `VOP_IntraDCVlcThr` parameter is corrupted and consequently the rest of the DC coefficients in the entire video frame are affected.

Another parameter, which has to be characterized here is the so-called Macroblock type and Coded Block Pattern for Chrominance (`MCBPC`) parameter, which is also a variable-length code that is used for signalling the MB type and the coded block pattern for chrominance. This parameter is always included for every encoded MB. When it is corrupted, it may result in another 'new' VLC, which is not included in the VLC index table. This would also affect the rest of the VLC positions in the bitstream or even more seriously, it would result in annoying artifacts, as mentioned before. Figure 4.16(f) shows the effects of the corrupted `MCBPC` of MB1, which is the first MB at the top left corner of the picture. Obviously, the affected part was concentrated in the area of MB1, but this parameter is variable-length coded, it may also gradually contaminate the rest of the MB in the picture.

To elaborate a little further, Figure 4.17 illustrates the effects of a bit error occurred in MB83, where we observe that the picture is less contaminated compared to the one illustrated in Figure 4.16(f). Annoying artifacts only start to emerge from MB83 onwards and there are 99 MBs in a QCIF-resolution video frame.

During our inter-frame coded bit error sensitivity study, investigations were conducted using the second frame of the "Miss America" sequence and the bitrate was the same as above. The corrupted frames are displayed in Figure 4.18, while the corresponding PSNR degradation is shown at index of Figure 4.15. Most of the video artifacts caused by errors that occurred in the VOP header of

Figure 4.17: Image degradation due to the corruption of DCT coefficient in frame 1, MB83 for the "Miss America" QCIF video sequence encoded at 30 frame/s and 1.15Mbit/s.

(a) Corrupted F_code_Forward (b) Corrupted DCT_coefficient (c) Error–free frame

Figure 4.18: Image degradation due to the corruption of the MPEG-4 coding parameters in an inter-frame coded sequence.

inter-frame pictures are similar to these recorded Intra-frame coded scenario described above.

The `VOP_fcode_forward` parameter of Table 4.2 is only required in the inter- frame coding mode. This is used for decoding of the motion vectors in the forward direction. When this parameter is corrupted, it results in wrong forward prediction for the motion vectors (MV) and hence some of the MBs, which also have associated MVs might be lost or not decoded properly as shown in Figure 4.18(c). Figure 4.18(b) also displays the video degradation inflicted by errors in the inter-frame DCT coefficients. As can be seen in figure 4.18 that the video degradation is not as severe as in intra coding (Figure 4.16), information used to encode an inter frame are much more lesser than Intra-frame coding, and inter-frame coding uses predicting information from the reference frame. So if the inter DCT-coefficient is corrupted, the corrupted information can be concealed by other DCT information from the predicted video frame.

In summary, as these graphs and figures reveal, the intra-frame errors result in image degradations that diminish very little with time, additionally, errors will propagate to the consequence inter frames until the next intra frame commence. In contrast, the errors in an inter-coded frame do not affect the image quality as adversely as the intra-coded frame errors.

4.8 Chapter Conclusions

In this chapter we present the error resilience aspects of the MPEG-4 video standard. A number of error resilience tools have been adapted into the MPEG-4 video standard, which facilitate the robust transmission of compressed video over error-prone communication channels, such as wireless links. In Section 4.6, we described these tools in detail and highlighted their relative merits.

Having discussed the MPEG-4 bitstream and error resilience tools, in Section 4.5, we investigated the hierarchial MPEG-4 bitstream organization, more specifically the VO layer, VOL layer, VOP layer and MB layer. We also quantified the relative frequency of certain parameters. We have shown that the transform coefficient (TCOEFF) parameter has the highest occurrence, probability of representing in excess of 90% of the total bits at a bitrate of 1 Mbps using the QCIF "Miss America" video sequence. Then a series of MPEG-4 bit sensitivity studies were carried out, demonstrating that the header information of the VO, VOL, VOP and MB hierarchies are extremely sensitive, because if the header's parameters are corrupted, the entire encoded bitstream may become undecodable. By contrast, in the presence of errors in the TCOFF, the bitstream can still decoded, although the reconstructed video texture may be perceptually degraded, as seen in Figures 4.17 and 4.18.

The perceptual effects of some of the erroneous or corrupted bits can be concealed to a certain degree with the aid of the error-resilience tools introduced in the MPEG-4 standard. However, we also proposed a number of other novel methods that further improve the performance of the wireless video codec, which were not incorporated the standard. If the encoder and decoder are aware of the limitations imposed by the communication channel, they are capable further improving the resilience of the codec. This chapter was mainly dedicated to the error resilience aspects of the video codec. Hence in the forthcoming chapters we will focus our attention on the system architecture of a range of attractive wireless videophone systems.

Chapter 5

Wireless Video Telephony

5.1 Introduction

Following the launch of the third-generation (3G) systems, interactive video telephony at various bit rates and video qualities become a commercial reality. Motivated by these events, the goal of this chapter is to quantify the expected video performance of various sophisticated video phone schemes.

More specifically, in this chapter we consider a range of dynamically or near-instantaneously reconfigurable modems. These near-instantaneously reconfigurable transceivers are also often referred to as burst-by burst adaptive modems [71]. Their philosophy is that a higher order modulation scheme is employed when the channel quality is favourable, in order to increase the system's bits per symbol capacity. Conversely, a more robust lower order modulation scheme is employed, when the channel exhibits an inferior channel quality, in order to improve the attainable mean bit error ratio (BER) performance. A reliable, low-delay feedback path is created between the transmitter and receiver, for example, using the system's control channel or by superimposing the estimated channel quality perceived by the receiver on the reverse-direction messages of a duplex interactive channel. The transmitter then adjust its modem mode according to this perceived channel quality.

More specifically, in this chapter, three videophone systems are proposed. In section **??**, a burst-by-burst adaptive coded modulation-aided joint detection-based CDMA for wireless video telephony. In Section 5.3 multi-mode joint-detection CDMA/H.264 based wireless video telephony is proposed. Finally in Section 5.4, both LDPC and turbo coding assisted space-time block Coded OFDM is invoked for H.264-compressed wireless video telephony.

5.2 Burst-by-Burst Adaptive Coded Modulation-Aided Joint Detection-Based CDMA for Wireless Video Telephony.[1]

5.2.1 Introduction

The radio spectrum is a scarce resource. Therefore, one of the most important objectives in the design of a digital cellular system is the efficient exploitation of the available spectrum in order to accommodate the ever-increasing traffic demands. Trellis Coded Modulation (TCM) [184], which is based on combining the functions of coding and modulation, is a bandwidth efficient scheme that has been widely recognised as an excellent error control technique suitable for applications in mobile communications. Turbo Trellis Coded Modulation (TTCM) [185] is a more recent channel coding scheme, which has a structure similar to that of the family of power efficient binary turbo codes [186], but employs TCM codes as component codes. In our TCM and TTCM schemes, random symbol interleavers were utilised for both the turbo interleaver and the channel interleaver.

Another powerful Coded Modulation (CM) scheme utilising bit-based channel interleaving in conjunction with Gray signal labelling, which is referred to as Bit-Interleaved Coded Modulation (BICM), was proposed in [187]. It combines conventional non-systematic convolutional codes with several independent bit interleavers. The number of parallel bit-interleavers used equals the number of coded bits in a symbol [187]. Recently, iteratively decoded BICM using SP based signal labelling, referred to as BICM-ID has also been proposed [188].

However, the above fixed mode transceivers failed to counteract the time varying nature of mobile radio channels. By contrast, in Burst-by-Burst (BbB) adaptive schemes [189] a higher order modulation mode is employed, when the instantaneous estimated channel quality is high in order to increase the number of Bits Per Symbol (BPS) transmitted and conversely, a more robust lower order modulation mode is employed, when the instantaneous channel quality is low, in order to improve the mean Bit Error Rate (BER) performance. Both uncoded adaptive schemes [189] and coded adaptive schemes [190] have been widely investigated for transmissions over narrow-band fading channels in the context of both single- and multi-carrier modems. They are however also powerful in the context of Direct Sequence Code Division Multiple Access (DS-CDMA) transceiver, which perform best in conjunction with Joint Detection (JD) of all users.

Joint detection [75] receivers are derivatives of the well-known single-user equalisers, which were originally designed for equalising signals that have been corrupted by Inter-Symbol Interference (ISI) due to the multipath effect of wireless channels. The Minimum Mean Square Error Decision Feedback

[1]This section is based on **J.Y. Chung, S.X. Ng, E.L. Kuan and L. Hanzo**: Burst-by-Burst Adaptive Coded Modulation-Aided Joint Detection-Based CDMA for Wireless Video Telephony, Proceedings of IEEE VTC Spring, Birmingham, USA, May 2002, vol. 3, pp. 1317-1321 and it was based on collaborative research with the co-authors [183].

Equaliser (MMSE-DFE) based JD (JD-MMSE-DFE) scheme constitutes a powerful approach to mitigating the effects of multi-user interference (MUI) and ISI [71], while at the same time improving the system's performance by benefiting from the multipath diversity effects of the channels.

In this study, a BbB Adaptive Coded Modulation-Aided Joint Detection-Based CDMA (ACM-JD-CDMA) scheme is proposed for wireless video telephony and characterised in performance terms over the UMTS Terrestrial Radio Access (UTRA) wideband vehicular fading channel. The BbB-Adaptive Coded Modulation (ACM) scheme is assisted by a JD-MMSE-DFE based receiver, when transmitting over the dispersive Rayleigh fading channels of a multiuser CDMA system. In our practical approach, transmitter A obtains the channel quality estimate generated by receiver B upon receiving the transmission of transmitter B. In other words, the modem mode required by receiver B for maintaining a certain target integrity is superimposed on the transmission burst of transmitter B. Hence a delay of one transmission burst duration is incurred.

The rest of this treatise is organised as follows. Our system overview is presented in Subsection 5.2.2, while the associated video aspects are summarised in Subsection 5.3.3. The channel model and the system parameters are described in Subsection 5.4.3. Our simulation results are discussed in Subsection 5.4.4 and finally our conclusions are offered in Subsection 5.4.5.

5.2.2 System Overview

The information bits generated by the video encoder are first channel encoded in order to generate non-binary symbols according to the Coded Modulation (CM) mode chosen. The CM-encoded symbols are then spread with the aid of the spreading code assigned to the user, modulated on to the carrier and transmitted. At the receiver, the data symbols of all the users are detected jointly, employing the MMSE-BDFE and the detected symbols are channel decoded by the corresponding CM decoder, before they are processed by the video decoder.

Joint detection CDMA is suitable for combining with ACM, because the implementation of the joint detection algorithms does not require any knowledge of the modulation mode used. The system matrix required for joint detection is constructed by using only the Channel Impulse Response (CIR) estimates and the spreading sequences of all the users. Therefore, the joint detection receiver does not have to be reconfigured, when the modulation mode is switched and its complexity is essentially independent of the modulation mode used.

In joint detection systems, the SINR of each user at the output of the MMSE-BDFE can be calculated by using the channel estimates and the spreading sequences of all the users. By assuming that the transmitted data symbols and the noise samples are uncorrelated, the expression for calculating

the SINR, γ_o, of the n-th symbol transmitted by the k-th user was given by Klein *et al* [191] as :

$$\gamma_o(j) = \frac{\text{Wanted Signal Power}}{\text{Res. MAI and ISI Power + Eff. Noise Power}}$$

$$-1), = g_j^2[\mathrm{D}]_{jj}^2 - 1, \qquad \text{for} \quad j = n + N(k-1), \tag{5.1}$$

where SINR is the ratio of the wanted signal power to the residual MAI and ISI power plus the effective noise power. The number of users in the system is K and each user transmits N symbols per transmission burst. The matrix D is a diagonal matrix that is obtained with the aid of the Cholesky decomposition [192] of the matrix used for linear MMSE equalization of the CDMA system [191]. The notation $[\mathrm{D}]_{jj}$ represents the element in the j-th row and j-th column of the matrix D and the value g_j is the amplitude of the j-th symbol.

After the output SINR is calculated, the best-matching modulation mode is chosen accordingly and communicated to the transmitter. Let us designate the choice of modulation modes to be V_m, where the total number of modulation modes is $M = 4$ and $m = 1, 2, \ldots, M$. The modulation mode having the lowest number of constellation points is V_1 and the one associated with the highest is V_M. The rules used for switching the modulation modes are as follows:

$$\begin{aligned}
\Gamma_o(k) \leq t_1 &\implies V_1 = 4QAM \\
t_1 < \Gamma_o(k) \leq t_2 &\implies V_2 = 8PSK \\
t_2 < \Gamma_o(k) \leq t_3 &\implies V_3 = 16QAM \\
t_3 \leq \Gamma_o(k) &\implies V_4 = 64QAM,
\end{aligned}$$

where $\Gamma_o(k)$ is the SINR of the k-th user at the output of the MMSE-BDFE, which was calculated by using Equation 5.1 and $\Gamma_o(k) = \frac{1}{N}\sum_{n=1}^{N}\gamma_o(j)$, $j = n + N(k-1)$. The values $(t_1, \cdots, t_{M-1})$ represent the switching thresholds used for activating the modulation modes, where we have $t_1 < t_2 < \cdots < t_{M-1}$.

We invoke four channel encoders, each adding one parity bit to each information symbol, yielding the coding rate of 1/2 in conjunction with the modulation mode of 4QAM, 2/3 for 8PSK, 3/4 for 16QAM and 5/6 for 64QAM. The TCM scheme invokes Ungerböck's codes [184], while the TTCM scheme invokes Robertson's codes [185]. The BICM scheme was constructed using Paaske's non-systematic convolutional codes [193]. The rate 5/6 code of the BICM scheme was constructed using Paaske's rate 1/2 code and puncturing, following the approach of [194, 195]. The BICM-ID scheme was created using Li's [188] approach.

Soft decision trellis decoding utilising the Log-Maximum A Posteriori (Log-MAP) algorithm [196] was invoked for the decoders. The Log-MAP algorithm is a numerically stable version of the MAP algorithm operating in the log-domain, in order to reduce the implementational complexity and to mitigate the numerical problems associated with the MAP algorithm [197].

Features	Multi-rate System			
Mode	4QAM	8PSK	16QAM	64QAM
Transmission Symbols	240			
Bits/Symbol	2	3	4	6
Transmission bits	480	720	960	1440
Packet Rate	100/s			
Transmission bitrate (kbit/s)	48	72	96	144
Code Termination Symbols	6			
Data Symbols	234			
Coding Rate	1/2	2/3	3/4	5/6
Information Bits/Symbol	1	2	3	5
Unprotected bits	234	468	708	1170
Unprotected bitrate (kbit/s)	23.4	46.8	70.8	117.0
Video packet CRC (bits)	16			
Feedback protection (bits)	9			
Video packet header (bits)	11	12	12	13
Video bits/packet	198	431	671	1138
Effective Video-rate (kbit/s)	19.8	43.1	67.1	113.8
Video framerate (Hz)	30			

Table 5.1: Operational-mode specific transceiver parameters for TTCM.

5.2.3 Video Overview

In this study, we transmitted 176x144-pixel Quarter Common Intermediate Format (QCIF) resolution video sequences at 10 frames/s using a reconfigurable TDD/CDMA transceiver, which can be configured as a 2, 3, 4 or 6 bit/symbol scheme. The proposed video transceiver is based on the H.263 video codec [198].

The associated codec parameters are summarised in Table 5.5. The channel-coded video coded bit stream is combined with an intelligent BbB adaptive wideband multi-mode modem [72]. This system would support an increased throughput expressed in terms of the average number of bits per symbol, when the instantaneous channel quality was high, leading ultimately to an increased video quality in a constant bandwidth.

Let us now highlight the philosophy of the video packetisation method. The size of the video packets varies depending on the current operating mode of multi-mode modem. The proposed multi-mode system can be configured to switch amongst the 2, 3, 4 and 6 bit/symbol modulation schemes based upon the near-instantaneous channel conditions. As seen in Table 5.5, when the channel is benign, the video bitrate is approximately 113Kbps. However, as the channel quality degrades, the modem will switch to the 4QAM mode of operation, where the video bitrate drops to 19 kbps.

The video transmitter is informed of the packet's transmission success or failure with the aid of a highly protected feedback channel, which is integrated into the reverse link. More explicitly, the associated feedback flag is protected with the aid of a repetition code and superimposed on the reverse-direction information packets. The use of packet acknowledgement allows the video encoder and the remote decoder to keep synchronised, operating on the basis of identical reconstruction frame buffer contents without the need of retransmissions, which are wasteful in terms of bandwidth efficiency and transmission delay.

5.2.4 Channel Model and System Parameters

Parameter	Value
Carrier Frequency	1.9GHz
Vehicular Speed	30mph
Doppler frequency	85Hz
System Baud rate	3.84 MBd
Normalised Doppler frequency	$85/3.84=2.21 \times 10^{-5}$
Channel type	UMTS Vehicular Channel A
Number of paths in channel	6
Data modulation	Adaptive Coded Modulation (4QAM, 8PSK, 16QAM, 64QAM)
Receiver type	JD-MMSE-DFE
No. of symbols per JD block	15

Table 5.2: Modulation and channel parameters

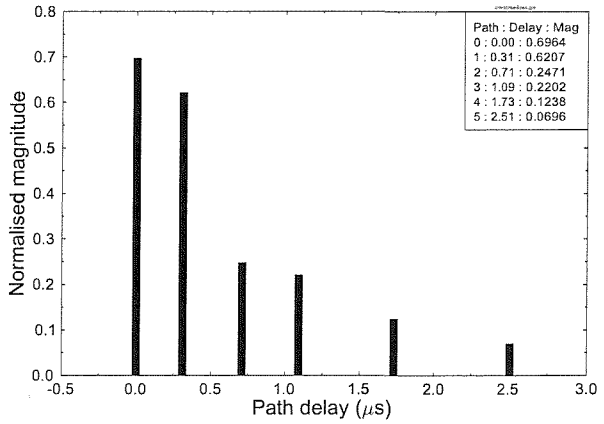


Figure 5.1: UTRA vehicular channel A [199].

Table 5.2 shows the modulation and channel parameters employed. The multi-path channel model is characterized by its discretised chip-spaced UTRA vehicular channel A [199]. The corresponding

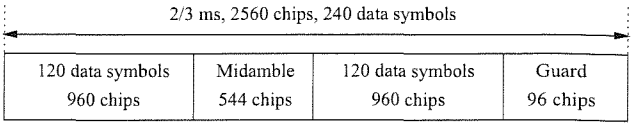


Figure 5.2: A modified UTRA Burst 1 [4] with a spreading factor of 8. The original UTRA burst has 244 data symbols.

channel impulse response is shown in Figure 5.1, where each path is faded independently according to the Rayleigh distribution. The transmission burst structure of the modified UTRA Burst 1 [4] using a spreading factor of eight is shown in Figure 5.2. The number of data symbols per JD block is 15, hence the original UTRA Burst 1 was modified to host a burst of 240 data symbols, which is a multiple of 15. The remaining system parameters are shown in Table 5.3.

Features	**Value**
Multiple access	CDMA, TDD
No. of Slots/Frame	15
Spreading factor	8
Frame length	10ms
Slot length	2/3ms
Data Symbols/Slot	240
No. of Slot/User	1
User Data Symbol Rate (KBd)	240/10 = 24
System Data Symbol Rate (KBd)	24x15 = 360
Chips/Slot	2560
Chips/Frame	2560x15=38400
User Chip Rate (KBd)	2560/10 = 256
System Chip Rate (MBd)	38.4/10 = 3.84
System Bandwidth (MHz)	3.84 x 3/2 = 5.76
Eff. User Bandwidth (kHz)	5760/15 = 384

Table 5.3: *Generic system features of the reconfigurable multi-mode video transceiver, using the spread data burst 1 of UTRA proposal [4, 199] shown in Figure 5.2.*

We compare the performance of TTCM employing constraint length three constituent TCM schemes using four iterations, to that of non-iterative TCM along with a constraint length of six. We also compare these results to non-iterative BICM employing a constraint length of six as well as BICM-ID in conjunction with a constraint length of three and using eight iterations. Hence the associated computational complexity of these four CM schemes is similar.

5.2.5 Simulation Results and Discussions

Initial simulations were performed using the transceiver configured in one of the four fixed modulation modes of Table 5.5. Figure 5.3 portrays the Packet Loss Rate (PLR) for the multi-mode system, in each of its modulation modes for a range of channel SNRs. It can be seen that above 25 dB the 64QAM mode offered an acceptable packet loss ratio of less than 5%, while providing a video rate of about 113Kbps. When the channel SNR drops below 25 dB, the multi-mode system is switched to 16QAM, then to 8PSK and eventually to 4QAM.

Figure 5.3 shows the PLR performance of the four different joint coding and modulation schemes for the four fixed modulation modes considered. It can be seen that the TTCM schemes have the best performance. This is because turbo decoding of the TTCM scheme is very effective in terms of reducing the number of bit errors to zero in packets exhibiting a moderate number of bit errors before decoding.

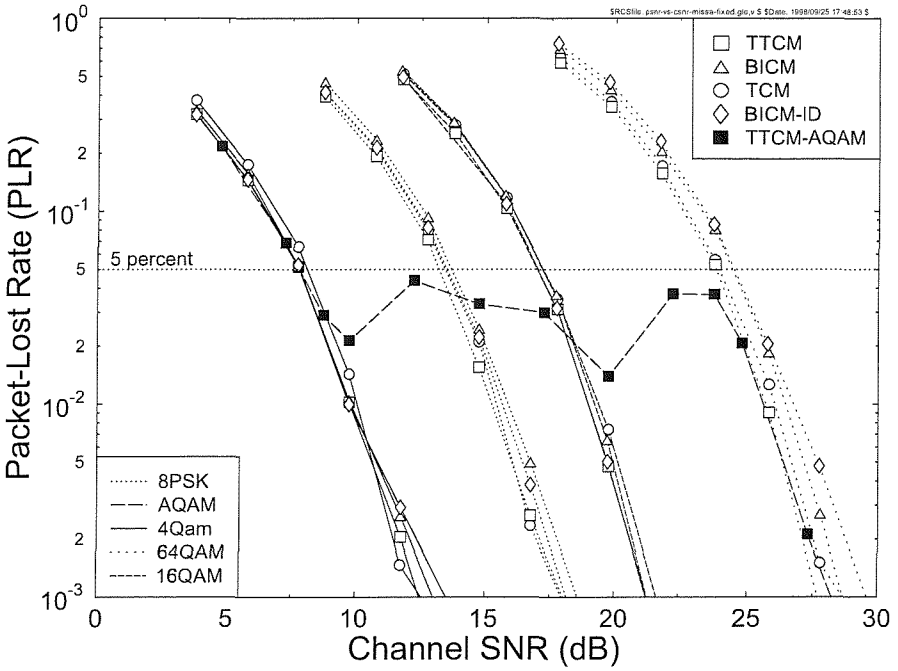


Figure 5.3: Packet loss ratio versus channel SNR for the four fixed modem modes, using the four joint coding/modulation schemes considered, namely BICM-ID, BICM, TCM and TTCM, when communicating over the UTRA channel of Figure 5.5.

The video quality is commonly measured in terms of the Peak-Signal-To-Noise-Ratio (PSNR). Figure 5.5 shows the video quality in terms of the PSNR versus the channel SNR for each of the modulation modes. As expected, the higher throughput bitrate of the 64QAM mode provides a better video quality. However, as the channel quality degrades, the video quality of the 64QAM mode

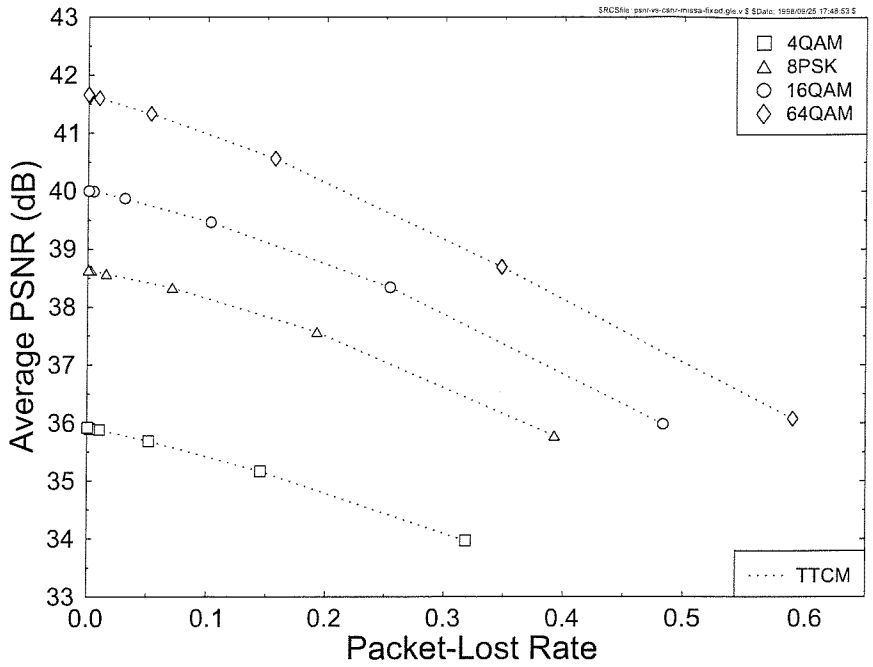


Figure 5.4: Decoded video quality (PSNR) versus video packet loss ratio for the four modulation modes.

is reduced and hence it becomes beneficial to switch from 64QAM to 16QAM, then to even lower throughput modulation modes, as the channel quality degrades.

The effect of packet losses on the video quality is quantified in terms of the PSNR, as depicted in Figure 5.4. The figure shows how the video quality degrades, as the PLR increases. It has been found that in order to ensure a seamless degradation of the video quality as the channel SNR reduced, it was beneficial to switch to a more robust modulation mode, when the PLR exceeded 5%. Although this inherently reduced the bitrate and the associated video PSNR, this was less objectionable in subjective video quality terms, than a PLR in excess of 5% would have been.

The mode switching operation of the adaptive scheme is characterised by a set of switching thresholds, by the delay imposed by the corresponding random TTCM symbol-interleavers and by the component codes, which has been also mentioned in Section 5.2.2. The average PSNR versus channel SNR performance of the four fixed TTCM modes and also of the four-mode TTCM AQAM scheme is shown in Figure 5.5. Finally, Figure 5.6 portrays the average PSNR versus Channel SNR performance of the four-mode TTCM AQAM scheme for PLRs of 1%, 5% and 10%.

As expected, the figure portrays that the AQAM modem's video performance degrades smoothly, when the channel SNR degrades, while the fixed modem's video performance degrades more rapidly, when the channel SNR becomes insufficiently high for the specific modem mode concerned.

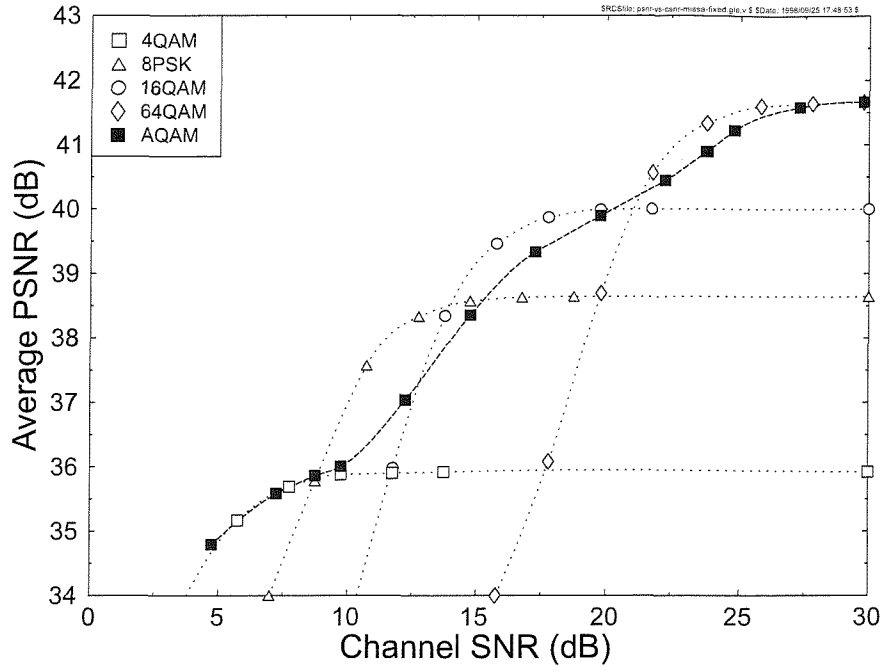


Figure 5.5: Average PSNR versus channel SNR for the four fixed TTCM modes and for the four-mode TTCM AQAM scheme using the QCIF video sequence at 30 frame/s.

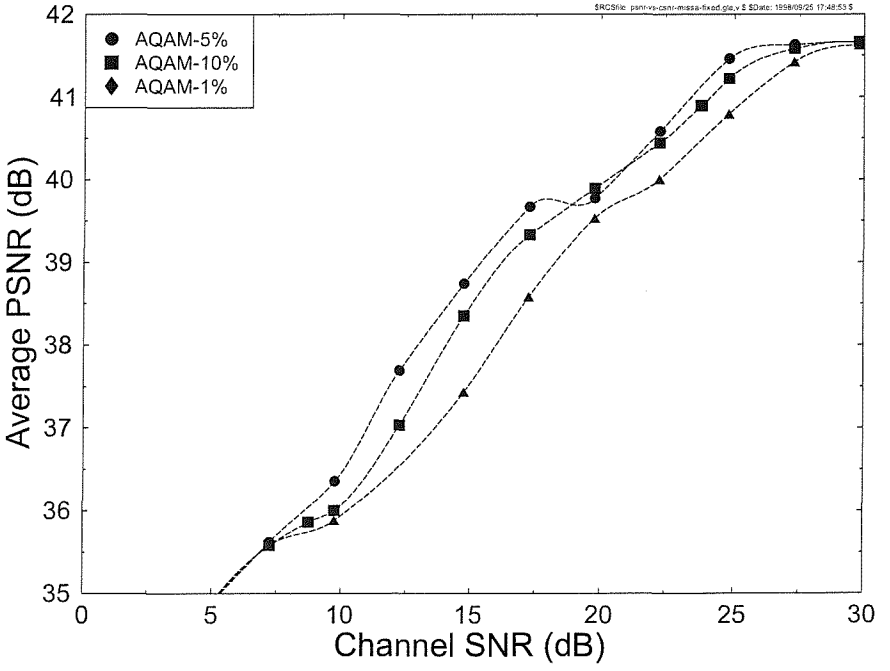


Figure 5.6: Average PSNR versus channel SNR for four-mode TTCM AQAM scheme in 1%, 5% and 10% packet loss representation.

5.2.6 Conclusion on BbB AQAM aided JD-CDMA for H.263 Video Telephony.

In this study, various burst-by-burst adaptive coded modulation aided joint detection based CDMA video transceivers have been studied, when communicating over the UTRA wideband vehicular fading channel. The adaptive transceiver is capable of operating in four different coded modulation modes, namely using 4QAM, 8PSK, 16QAM and 64QAM.

Various coded modulation schemes used in our experiments. The advantage of the adaptive coded modulation schemes, such as adaptive TTCM is that when invoking higher-order modulation modes in case of encountering a higher channel quality, the coding rate approaches unity. As a result, this would allow the video transceiver to maintain a high throughput and high video quality. In our case the bitrate of the 4QAM and 64QAM modes was 23.4 kbps and 117 kbps, respectively. The coding rate of 4QAM was 1/2, while that of 64QAM was 5/6. This would support effective video bitrates between approximately 20 kbps to 114 kbps.

The burst-by-burst adaptive modem guaranteed the same video performance, as the lowest- and highest-order fixed-mode modulation schemes at a range of low and high channel SNRs. However, between these extreme SNRs the effective video bitrate smoothly increased, as the channel SNR increased, whilst maintaining a near-constant PLR. By controlling the AQAM switching thresholds a near-constant PLR can be maintained.

5.3 Multi-Mode Joint-Detection CDMA/H.264 Based Wireless Video Telephony[2]

5.3.1 Introduction

As we discussed earlier in Section 5.2, Turbo Trellis Coded Modulation (TTCM) [74,185,200] is capable of outperforming other coded modulation schemes, both in terms of its achievable BER and in the context of H.263 based video telephony [198], when communicating over the UMTS Terrestrial Radio Access (UTRA) wideband vehicular fading channel [4,199], while maintaining a comparable decoding complexity.

Another powerful channel coding scheme involved in our study is a representative of the family of Low Density Parity Check (LDPC) codes, which were devised by Gallager [201] in 1962. LDPC codes belong to the class of linear block codes, which are defined by a parity check matrix having M rows and N columns. The column weight and row weight is low compared to the dimension M and N of

[2]This section is based on **J.Y. Chung, F. Guo, S.X. Ng and L. Hanzo**: Multi-Mode Joint-Detection CDMA/H.26L Based Wireless Video Telephony; published in VTC Fall 2003and it was based on collaborative research with the co-authors

the parity check matrix. During the early evolutionary phase of channel coding, LDPC schemes made limited impact on the research of the channel coding community, despite their impressive performance, which was unprecedented prior to the turbo coding era. This lack of popularity was a consequence of its relatively high storage requirement and complexity. However, in recent years research interests in LDPC codes have been rekindled owing to their capability of approaching Shannon's predicted performance limits. In order to lend the LDPC codes a high spectral efficiency, a LDPC-based Block Coded Modulation (LDPC-BCM) scheme was proposed in [202]. When employing a long codeword length of 3000 modulated symbols, regular LDPC-BCM was found to slightly outperform TTCM in the context of non-dispersive uncorrelated Rayleigh fading channels at a similar coding rate and a comparable decoding complexity [203].

In order to counteract the time varying nature of mobile radio channels, in this study Adaptive Coded Modulation (ACM) was employed [73]. Explicitly, a higher order modulation mode is employed, when the instantaneous estimated channel quality is high, in order to increase the number of Bits Per Symbol (BPS) transmitted. Conversely, a more robust lower order modulation mode is employed, when the instantaneous channel quality is low, in order to improve the mean Packet Loss Ratio (PLR) performance. The Minimum Mean Square Error Decision Feedback Equaliser (MMSE-DFE) based Joint Detector (JD-MMSE-DFE) is invoked in the proposed Code Division Multiple Access (CDMA) [71] ACM scheme. Specifically, joint detection [75] receivers are derivatives of the well-known single-user equalisers, which were originally designed for equalising signals that have been corrupted by Inter-Symbol Interference (ISI) due to the multipath effect of wireless channels. The JD-MMSE-DFE scheme constitutes a powerful approach to mitigating the effects of both Multi-User Interference (MUI) and ISI [71], while at the same time improving the system's performance by benefiting from the multipath diversity gain provided by the dispersive channel.

In this section, a Burst-by-Burst ACM-aided Joint Detection-Based CDMA (ACM-JD-CDMA) scheme is proposed for H.264/H.263 based wireless video telephony [166]. The system is characterised in performance terms when communicating over the uplink (mobile station-to-base station) UTRA [71] wideband vehicular fading channel. Specifically, regular LDPC-BCM is studied and compared to TTCM in the context of the proposed ACM-JD-CDMA system.

5.3.2 System Overview

We invoke four different-rate channel encoders in the context of the TTCM/LDPC-BCM scheme, each adding one parity bit to each information symbol, yielding a coding rate of 1/2 in conjunction with the modulation mode of 4QAM, 2/3 for 8PSK, 3/4 for 16QAM and 5/6 for 64QAM. The TTCM scheme invoked Robertson's codes [185] and the Log-Maximum A Posteriori (Log-MAP) decoding algorithm [196] was utilised. The regular LDPC-BCM scheme was designed by adjusting the row

weight of its generator matrix, in order to arrive at the desired coding rate, while stipulating a fixed column weight of three [202]. The decoding algorithm employed was the sum-product algorithm [204]. For the sake of a fair comparison, the parameters of TTCM and LDPC-BCM were adjusted such that their decoding complexity became similar. Explicitly, TTCM schemes employing TCM codes of memory three were invoked for the various modulation modes in conjunction with a fixed number of decoding iterations, namely four. For the sake of maintaining a similar complexity, the column weight of LDPC-BCM was fixed to three and the maximum number of decoding iterations of the LDPC-BCM scheme was set to 15, 10, 10 and 17 in conjunction with 4QAM, 8PSK, 16QAM and 64QAM, respectively [202].

At each mobile station, the information bits generated by the user's video encoder are first channel encoded in order to generate non-binary symbols according to the specific Coded Modulation (CM) mode chosen by the base-station on the basis of the instantaneous channel quality encountered, as suggested in [71]. The CM-encoded symbols are then direct-sequence spread with the aid of the CDMA multipath spreading code assigned to the user, modulated on to the carrier and transmitted over the dispersive UTRA channels. At the receiver of the base-station, the data symbols of all users are jointly detected by the JD-MMSE-DFE and the detected symbols are channel decoded by the corresponding CM decoder, before they are processed by the video decoder.

For the sake of benchmarking and a reasonable comparison with our previous system as discussed in Section 5.2, the joint detection systems we employed in this system has the same characteristic as in Section 5.2.2.

5.3.3 Video Overview

5.3.3.1 ITU-T VCEG project H.264

In this study, we transmitted 176x144-pixel Quarter Common Intermediate Format (QCIF) resolution video sequences at 30 frames/s using a reconfigurable TDD/CDMA transceiver, which can be configured as a 2, 3, 4 or 6 bit/symbol scheme. The proposed video transceiver is based on the H.264 video codec [166].

The H.264 codec [166] was devised by the ITU-T Video Coding Expert Group (VCEG). This study refers to the version described in the Test Model Long Term Number 8 (TML-8) [205]. The H.264 video coding layer's (VCL) algorithm has a design similar to that of the ratified video coding standards, such as those specified by the ISO MPEG visual and the ITU-T standards. In addition, it contains new features that enable to enhance further to achieve a significant improvement in compression efficiency in relation to the previously ratified coding standards, which was studied in Chapter 3.

In Chapter 3, we detailed various additional coding algorithm of the H.264 video standard. In

Features	Multi-rate System			
Mode	4QAM	8PSK	16QAM	64QAM
Transmission Symbols	240			
Bits/Symbol	2	3	4	6
Transmission bits	480	720	960	1440
Packet Rate	100/s			
Transmission bitrate (kbit/s)	48	72	96	144
Data Symbols	234			
Coding Rate	1/2	2/3	3/4	5/6
Information Bits/Symbol	1	2	3	5
Unprotected bits	240	480	720	1200
Unprotected bitrate (kbit/s)	24.0	48.0	72.0	120.0
Video packet CRC (bits)	16			
Feedback protection (bits)	9			
Video packet header (bits)	11	12	12	13
Video bits/packet	204	443	683	1162
Effective Video-rate (kbit/s)	20.4	44.3	68.3	116.2
Video framerate (Hz)	30			

Table 5.4: Operational-mode specific transceiver parameters for LDPC.

summary, every input Macroblock (MB) has to be predicted in the H.264 scheme before the transform coding process. Sub-blocks of 4×4 samples are used for transform coding. The conventional picture types known as Intra-frame coded (I), Inter-frame coded or Prediction (P) based and Bidirectional (B) prediction aided coding modes are still supported. There are two classes of Intra-frame coding modes, which are referred to as INTRA 16×16 and INTRA 4×4 modes. Moreover, in the H.264 coding scheme prediction is always utilised in the spatial domain by referring to the neighbouring video pixels of already coded blocks. When the INTRA 4×4 [74, 185] mode is used, each 4×4-pixel block of luminance samples utilises one of the six proposed prediction modes [166]. The identifiers of the prediction modes chosen are then transmitted to the decoder as side-information. For picture regions having less spatial detail, the Intra 16×16 prediction mode is employed, in which one of four prediction modes is chosen for the prediction of the entire MB [166].

As for the Inter-frame coded prediction modes, H.264 provides seven motion-compensated coding modes for the MBs of Inter-frame coded (P) pictures. Each motion compensated mode corresponds to a specific partitioning of the MB into fixed-size blocks used for enhancing the achievable quality of motion description. Currently, block sizes of 16×16, 16×8, 8×16, 8×4, 4×8 and 4×4 pixels are supported by the H.264 syntax [166], and thus up to 16 motion vectors can be transmitted for a MB. The so-called spiral search aided motion compensation procedure [166] finds the 'minimum cost' coding solution for each block size within a specified search window. The most popular block-

matching measure in this context is the Sum of Absolute Difference (SAD) [166]. The 'cost' includes the SAD-based coding distortion and the corresponding overhead bits required for encoding the block size information and the motion vectors. The optional block size is decided based on finding the minimum cost solutions. If a 4×4 block size is the 'winner', there are 16 motion vectors for the entire MB. The resolution of the motion vectors is at least a quarter of a pixel. When requiring a higher coding efficiency, the H.264 codec allows 1/8-pixel accuracy motion prediction, although this imposes a high encoding complexity.

The motion prediction residual is transformed to the spatial frequency domain using 4×4 pixel integer-valued DCT. Instead of the conventional two-dimensional (2D) DCT, H.264 uses a separable vertical and horizontal DCT, processing integer values while maintaining a similar performance to the conventional 2D 4×4 DCT. More explicitly, both the DCT and the inverse transform use exact integer operations, and hence the detrimental effects of quantisation errors are avoided [166]. The scanning order of the DCT coefficients is still based on zig-zag scanning, similarly to that used for example in the H.263 standard [4]. There are two different entropy coding techniques that are used in the H.264 codec for compressing the quantised DCT coefficients, namely the Universal Variable Length Codes (UVLC) [166] and Context-Based Adaptive Binary Arithmetic Coding (CABAC) modes [166]. The UVLC scheme provides a simple and robust method for encoding all side-information and the DCT coefficients. However, its achievable performance is modest at moderate and high bit rates, where the CABAC mode performs better [166].

5.3.3.2 Packetisation for Mobile Environments

Let us now consider the video packetisation algorithm. The video packet size varies in response to the current operating mode of the multi-mode modem. The proposed multi-mode system can be configured to switch amongst the 2, 3, 4 and 6 bit/symbol modulation schemes based upon the near-instantaneous channel conditions. Again, as shown in Table 5.5, when the channel is benign, the video bitrate is approximately 116 kbps. However, as the channel quality degrades, the modem will switch to the 4QAM mode of operation, where the video bitrate drops to 20 kbps.

The video transmitter is informed of the packet's transmission success or failure with the aid of a highly protected feedback channel, which is integrated into the strongly-protected reverse link. More explicitly, the associated feedback flag is protected with the aid of a repetition code and concatenated to the reverse-direction information packets. The use of packet acknowledgements allows the video encoder and the remote decoder to keep 'synchronised', i.e. to operate on the basis of identical reconstruction frame buffer contents without the need of video packet retransmissions, which are wasteful in terms of bandwidth efficiency and transmission delay.

5.3.4 Channel Model and System Parameters

Again, for the sake of benchmarking, the associated video system parameters are the same as we have discussed in Section 5.4.3. Table 5.2 in Section 5.4.3 summarises the modulation and channel parameters used in our experiment. The corresponding UTRA channel impulse response is shown in Figure 5.1.The transmission burst structure of the modified UTRA Burst 1 [4] using a CDMA spreading factor of eight is shown in Figure 5.2. For the sake of maintaining a low MUD complexity, the number of data symbols per JD block was set to 20, hence the original UTRA Burst 1 was slightly modified to host a burst of 240 data symbols, which is a multiple of 20. Furthermore, the number of slots per frame is 15, and the frame duration is 10ms.

5.3.5 Experimental Results and Discussions

Simulations were performed using the QCIF-format head-and-shoulder "Miss America" videophone sequence. The various modulation modes of Table 5.5 were used. Figure 5.7 shows the Packet Loss Ratio (PLR) versus channel SNR performance for each mode of the system. Our video scheme discards all video packets, which are not error-free, hence the PLR is a more accurate measure of the expected video performance than the BER, which is zero. Needless to say, our goal is to maintain as low a PLR as possible. Again, it should be noted that for channel SNRs in excess of 25 dB the 64QAM mode offered an acceptable packet loss ratio of less than 5%, providing a video rate of approximately 113 kbps. However, as the channel quality degrades, the multi-mode system is switched to the lower rate modes of 16QAM, then to 8PSK and eventually to 4QAM.

Having shown the effects of the channel's SNR on the PLR, let us now demonstrate these effects on the decoded video quality measured in terms of the Peak-Signal-to-Noise Ratio (PSNR). The available video quality of our video system is directly related to the PLR performance. Figure 5.8 shows the associated average PSNR versus channel SNR performance, demonstrating that at high channel SNRs the fixed modulation modes of the adaptive signalling regime exhibit a higher PSNR in conjunction with the higher-throughput modulation modes, than in the lower-throughput modulation mode. For example, at high SNRs 64QAM yields a PSNR of 43.60dB, whereas the 4AQM mode achieves a PSNR of 37.10 dB.

The video quality of each of the fixed modes degrades dramatically, as the PLR is increased, which results in parts of the picture being 'frozen' for one or possibly several consecutive video frames. This phenomena is illustrated in Figure 5.9.

As for the adaptive system, it can be seen from Figure 5.8 that the video quality degrades gracefully, when the system gradually switches from the high-throughput modes to the lower-throughput modes. The LDPC-based AQAM curve of Figure 5.8 has been repeated in Figure 5.10 for comparison. Here,

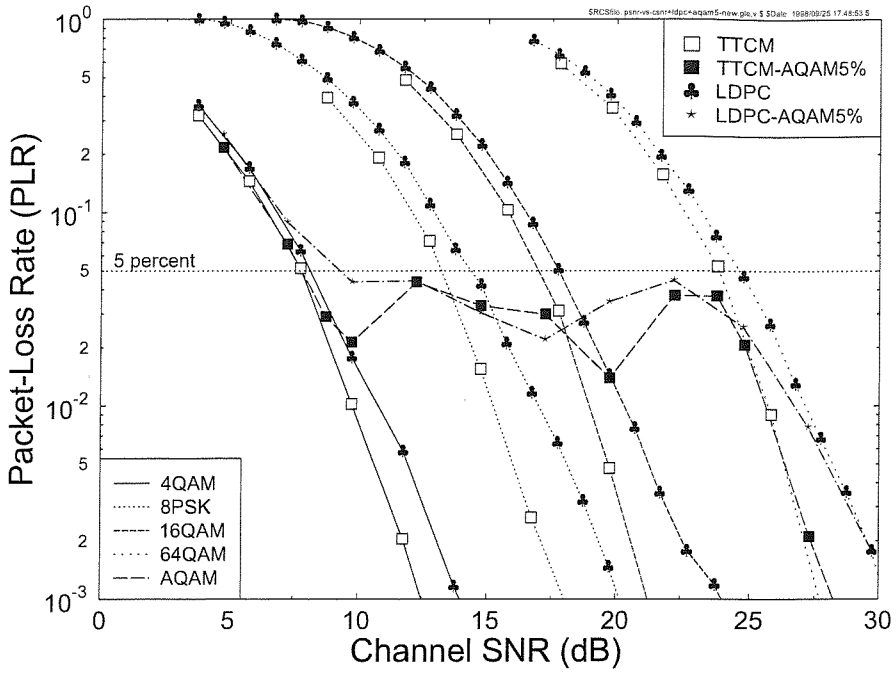


Figure 5.7: Packet loss ratio versus channel SNR for the four fixed modem modes, using the TTCM and LDPC joint coding/modulation schemes considered, when communicating over the UTRA channel of Figure 5.5.

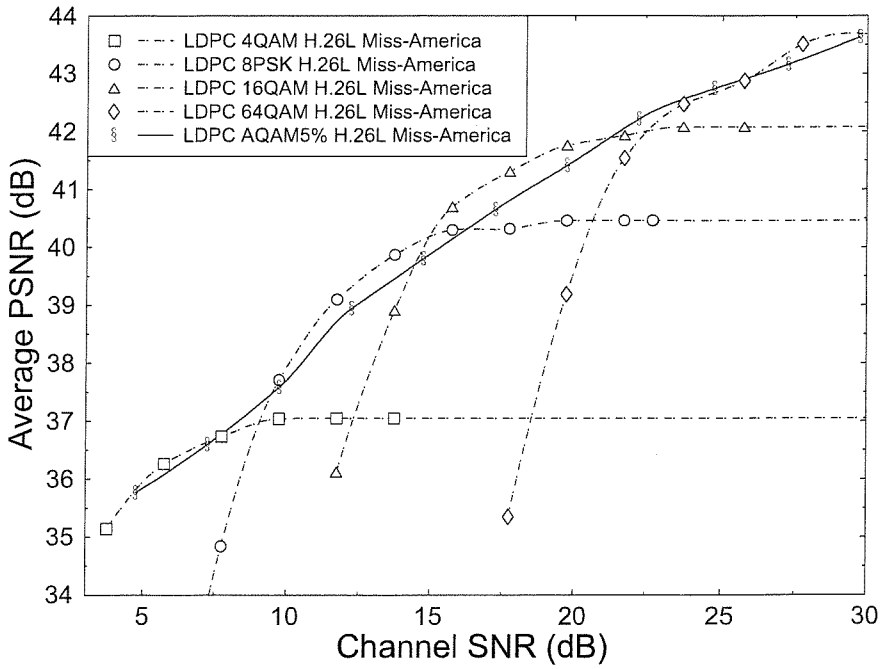


Figure 5.8: Average PSNR versus channel SNR for the four fixed modes and for the LDPC-based ACM/CDMA scheme using the QCIF Miss America video sequence at 30 frame/s.

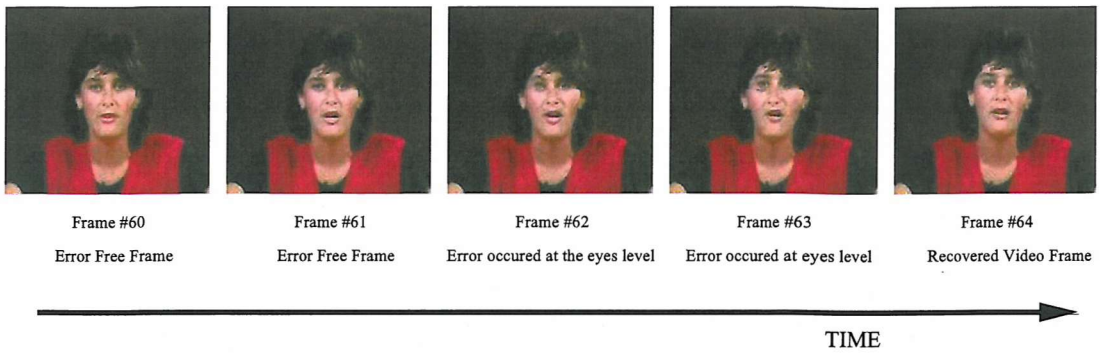


Figure 5.9: The channel has corrupted some of the transmitted packets resulting in part of the video image become frozen as shown in frames62 and 63. However the system soon recovers and replenishes the affected areas as in frame 64.

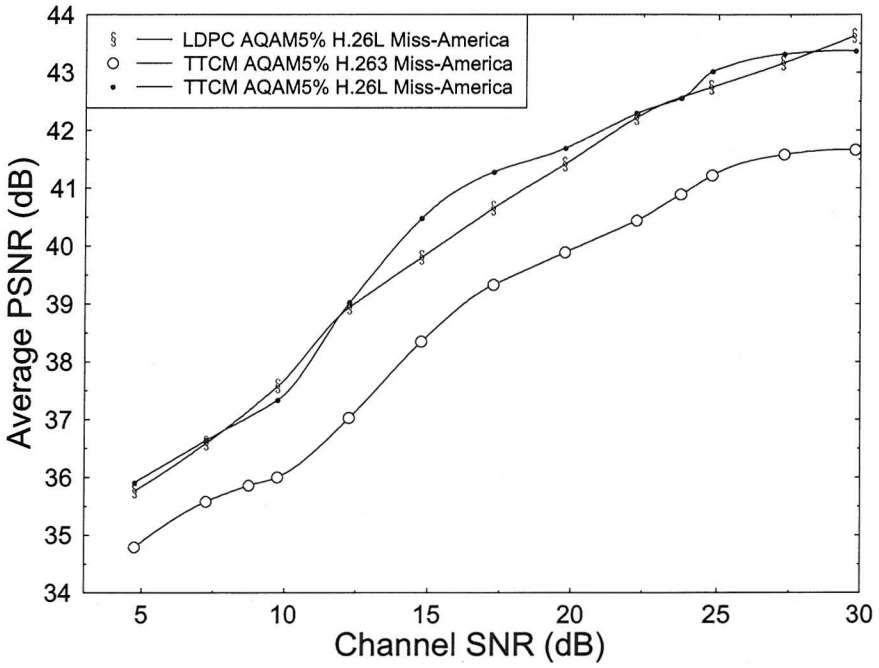


Figure 5.10: Average PSNR versus channel SNR for the LDPC and TTCM CDMA AQAM schemes at PLR = 5%.

we demonstrate the additional performance gains that are achievable, when TTCM coding is used in preference to the LDPC coding. Observe in Figure 5.10 that the overall performance of the TTCM assisted H.264 video codec is slightly higher than that of the LDPC aided H.264 video codec. These results were recorded at a similar decoding complexity, when a short codeword length of 240 modulated symbols was employed by both schemes. Note however that LDPC coding has been shown in [203] to slightly outperform TTCM at a given decoding complexity, when a long codeword length is invoked, while communicating over a fast-fading/uncorrelated channel.

For the sake of further benchmarking, our results recorded for a similar TTCM ACM/CDMA system using the H.263 codec of Section 5.2 [183] are also depicted in Figure 5.10. These results show that under the same channel conditions and using the same system parameters, the H.264 codec based system outperformed the H.263 coding aided arrangement.

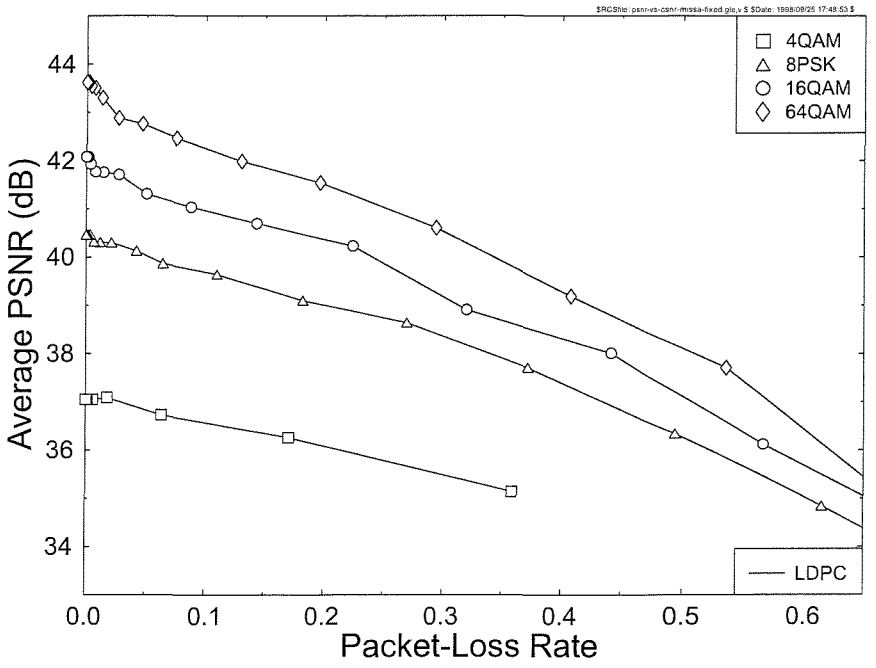


Figure 5.11: Decoded video quality (PSNR) versus video packet loss ratio for the four modulation modes.

The video quality versus PLR performance can be more explicitly observed in Figure 5.11. The figure shows how the video quality degrades, as the PLR increases. Observe that in order to ensure a seamless degradation of the video quality as the channel SNR reduced, it was beneficial to switch to a more robust modulation mode, when the PLR exceeded 5%. Although this inherently reduced the effective video bitrate and the associated video PSNR, the video degradation was less objectionable in subjective video quality terms, than inflicting a PLR in excess of 5% would have been.

5.3.6 Conclusions on Multi-Mode JD-CDMA/H.264 Based Wireless Video Telephony

As a conclusion, various burst-by-burst adaptive coded modulation aided joint detection based CDMA video transceivers have been studied, when communicating over the UTRA wideband vehicular fading channel.

A burst-by-burst adaptive ACM/CDMA modem maximises the system's throughput by using the most appropriate modulation mode in response to the instantaneous channel conditions. Furthermore, we have quantified the achievable video performance gains as a benefit of employing the proposed burst-by-burst adaptive CDMA modem.

The burst-by-burst adaptive modem guaranteed the same video performance, as the lowest- and highest-order fixed-mode modulation schemes at a range of low and high channel SNRs, respectively. However, between these extreme SNRs the effective video bitrate gracefully increased, as the channel SNR increased, whilst maintaining a near-constant PLR. By controlling the AQAM switching thresholds, a near-constant PLR can be maintained. Finally, as shown in Figure 5.10, the H.264 video codec outperformed the H.263 codec in video PSNR terms.

Following our discussions on joint-detection assisted CDMA-based burst-by-burst adaptive interactive videophony, in the next section, we focus our attention on the family of multi-carrier modems.

5.4 LDPC and Turbo Coding Assisted Space-Time Block Coded OFDM for H.264 Compressed Wireless Video Telephony[3]

5.4.1 Introduction

In the video system of Sections ?? and 5.3, single-carrier modems were used. However, in recent years the family of multi-carrier Orthogonal Frequency Division Multiplex (OFDM) modems [76] has also received considerable attention. Therefore in this section, we show that OFDM systems are also capable of supporting interactive videophony. In wireless video telephony a low transmission frame error rate (FER) is required for maintaining a high video quality. However, owing to the scarcity of radio spectrum, a bandwidth-efficient implementation is required so that a low FER can be achieved despite maintaining a high transmission bit rate, which necessitates the employment of multi level modulation schemes. Orthogonal Frequency Division Multiplexing (OFDM) is an attractive technique that can be invoked for high-bit-rate data transmission, especially in a highly dispersive 5.4.3

[3]This section is based on **J.Y. Chung, M.Y. Alias, F. Guo and L. Hanzo**: LDPC and Turbo Coding Assisted Space-Time Block Coded OFDM for H.26L Compressed Wireless Video Telephony, Proceedings of PIMRC'2003, Beijing, China, September 2003, pp. 2702-2706 and it was based on collaborative research with the co-authors

multipath-fading environment that inflicts inter-symbol interference (ISI) [76, 206].

A considerable amount of research has been invested into the design and implementation of space-time block coded (STBC) techniques in the context of OFDM systems [74, 207]. Specifically, STBC invokes antenna array aided spatial diversity and achieves significant diversity gains in wireless channels. For the sake of further improving the attainable performance, Forward-Error Correction (FEC) schemes such as Turbo Codes (TC) [74] and Reed-Solomon codes [74] may be invoked for protecting the OFDM subcarriers against frequency-selective fading in an OFDM environment.

In recent years, the family of Low Density Parity Check (LDPC) codes [201] has re-emerged as an attractive alternative to turbo coding [74]. LDPC codes were originally proposed by Gallager [201] in 1962. Owing to the codes' capability of approaching Shannon's performance limits, LDPC codes have been applied in conjunction with BPSK for transmission over both AWGN and frequency selective fading channels in conjunction with OFDM systems [208]. It has also been shown in [207] that LDPC-based space-time coded OFDM systems are capable of efficiently exploiting the achievable spatial diversity in wireless channels.

In this section, we studied the H.264 video standard combined with LDPC assisted space-time block coded OFDM using Alamouti's $\mathbf{G_2}$ space-time block code [209]. We investigated the performance when transmitting over frequency selective fading channels, in particular over the UTRA channel [183].

5.4.2 Video Transmission

Video sequences having 176x144-pixel Quarter Common Intermediate Format (QCIF) resolution scanned at 30 frames/s and transmitted using space-time block coded OFDM are employed in our study. The OFDM modem can be configured as a 2, 3, 4 or 6 bit/symbol scheme. The proposed video transceiver is based on the H.264 video codec which has been outlined in [166], discussed in Chapter 3 and highlighted in Section 5.3.3.1. The associated codec parameters are summarised in Table 5.5. The H.264 video codec employs variable-length compression techniques and therefore achieves a high compression ratio. However, as all entropy and variable-length coded bit streams, its encoded bits are potentially sensitive to transmission errors.

In the first one of our prototype systems which we refer to as **System 1**, this error sensitivity has been counteracted by invoking a packetisation and packet dropping technique [4], where we refrained from decoding the corrupted video packets in order to prevent error propagation through the reconstructed video frame buffer. Hence the corrupted video packets were dropped at both the transmitter and receiver and the video codec's reconstructed frame buffer was not updated, until the next error-free video packet replenished its contents.

When a packet is transmitted, the packetisation algorithm receives feedback from the receiver

Features	Multi-rate System			
Mode	4QAM	8PSK	16QAM	64QAM
Transmission Symbols	240			
Bits/Symbol	2	3	4	6
Transmission bits	480	720	960	1440
Packet Rate	100/s			
Transmission bitrate (kbit/s)	48	72	96	144
Data Symbols	234			
K, uncoded information	240	480	720	1200
N, LDPC coded bits	480	720	960	1440
TC(2,1,4) interleaver length	240	480	720	1200
Coding Rate	1/2	2/3	3/4	5/6
Information Bits/Symbol	1	2	3	5
Unprotected bitrate (kbit/s)	24.0	48.0	72.0	120.0
Video packet CRC (bits)	16			
Feedback protection (bits)	9			
Video packet header (bits)	11	12	12	13
Video bits/packet	204	443	683	1162
Effective Video-rate (kbit/s)	20.4	44.3	68.3	116.2
Video framerate (Hz)	30			

Table 5.5: Operational-mode specific transceiver parameters.

concerning the success or failure of the packet concerned with the aid of a highly protected feedback channel, which is integrated into the reverse link. The use of these packet acknowledgement flags allows the video encoder and the remote decoder to keep synchronised, operating on the basis of identical reconstruction frame buffer contents without the need of packet retransmissions, which are wasteful in terms of bandwidth efficiency and transmission delay [4].

By contrast, in our second prototype system referred to as **System 2**, we omit the above-mentioned packet transmission feedback from the system. To elaborate a little further, when using the packet transmission feedback, the receiver invokes one of the three following types of feedback messages: (1) Packet received without error. (2) Packet received with error, retransmission request. (3) Packet received with error, packet dropping request. In the receiver of System 2, which uses no transmission feedback, the entire video packet is dropped, when it was corrupted and no retransmission is required

Parameter	Value
OFDM subcarriers	256 (16 virtuals)
Space-time code	$\mathbf{G}_2$
Channel coding	LDPC
Column weight, j	3
Maximum iterations	25
Channel type	UMTS Vehicular Channel A
Number of paths	6
Carrier frequency	1.9 GHz
Vehicular speed	30 mph
Normalized Doppler	2.21×10^{-5}

Table 5.6: Common Tranceiver and channel parameters.

in this case. In case of packet corruption, both System 1 and 2 conceals the packet loss by replacing the affected area of the picture with the corresponding area from the previous video frame. However, owing to the lack of a feedback flag, the encoder of System 2 remains unaware of receiving a corrupted packet and hence cannot leave the corresponding video frame area unupdated.

5.4.3 Channel Model and System Parameters

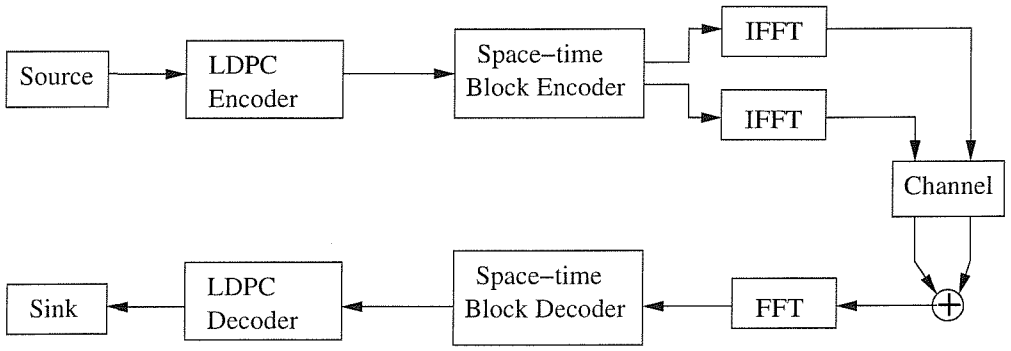


Figure 5.12: System overview of one-receiver LDPC assisted $\mathbf{G}_2$ space-time block coded OFDM.

The architecture of our system is shown in Figure 5.12. At the transmitter, the video encoder generates the video information data bits. The information bits are then encoded by the LDPC encoder. It has been shown in [203] that in the investigated scenario the LDPC codec performed better than turbo convolutional codes in terms of the achievable Frame Error Rate (FER) . We invoked an (N, K) LDPC code defined by the $(M \times N)$-dimensional parity-check matrix [201], where $K = (N - M)$ is the uncoded-information block length. LDPC codes belong to the family of linear

block codes. These codes are defined as codes using a sparse parity-check matrix having the same number of logical 1s per column (column weight, k) and the same number of logical 1s per row (row weight, j), where both of these numbers are small compared to the LDPC block length N. In our simulations we chose k and j values of 3. The code rate, R, of the LDPC codes is given by $R = K/N$. The appropriate values of K and M can then be calculated for the different values of N and R.

An example of the parity check matrix is given by:

$$\mathbf{C} = \begin{array}{c} \overset{\longleftarrow\ N\ \longrightarrow}{} \\ \left[\begin{array}{cccccccccc} 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \end{array}\right] \end{array} \begin{array}{c} \uparrow \\ N-K \\ \downarrow \end{array} \tag{5.2}$$

LDPC codes can be represented by a Factor Graph that contains two types of nodes: the "bit nodes" and the "check nodes". Each bit node corresponds to a column of a parity-check matrix, which also corresponds to a bit in the codeword. Each check node corresponds to a row of a parity-check matrix, which represents a parity-check equation. An edge between a bit node and a check node exists if and only if the bit participates in the parity-check equation represented by the check node, i.e. by a non-zero entry in the parity-check matrix. Figure 5.13 shows an example of a Factor Graph.

matrix $\mathbf{C}$. For example, the non-zero entry at the bottom right corner of matrix $\mathbf{C}$ corresponds to the connection between the 6^{th} node at the bottom and the 3^{rd} node on the top of Figure 5.13.

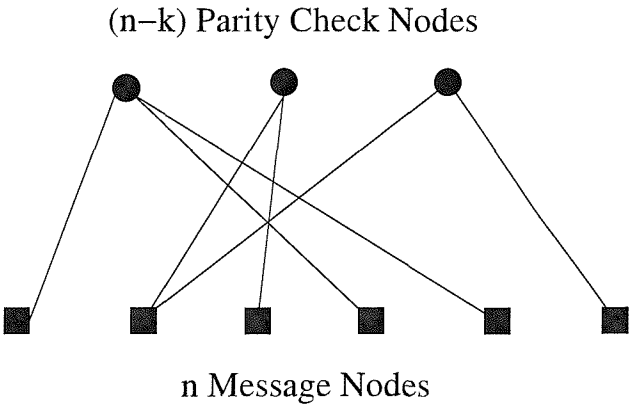


Figure 5.13: Factor Graph representation of the parity-check matrix.

The output bits of the LDPC channel encoders are then passed to the space-time block encoder of Figure 5.12. In our system, we employed Alamouti's $\mathbf{G_2}$ space-time block code since it was shown in [74] that from the specific set of schemes investigated, the best performance was achieved by concatenating the space-time block code $\mathbf{G_2}$ with channel codes. The $\mathbf{G_2}$ space-time block code is

associated with a twin-transmitter-based scheme, whose generator matrix is defined as follows:

$$\mathbf{G_2} = \begin{pmatrix} x_1 & x_2 \\ -\bar{x}_2 & \bar{x}_1 \end{pmatrix}. \tag{5.3}$$

The output of the space-time encoder is then OFDM modulated with the aid of the Inverse Fast Fourier Transform (IFFT) blocks of Figure 5.12 and transmitted by the corresponding antenna. The number of transmit antennas is fixed to two, while the number of receive antennas constituted a design parameter. We used an OFDM system having 256 subcarriers. However, since only 240 data symbols are being transmitted, the remaining 16 subcarriers were used as virtual subcarriers. Virtual subcarriers are usually used for accommodating a frequency-domain raised-cosine Nyquist-filtering based roll-off of the OFDM signal and to avoid a direct current (DC) offset in the demodulation process. The utilization of virtual subcarriers appears in most of wireless LAN standards, such as, IEEE 802.11a [206], and in the High Performance Local Area Network type 2 (HiperLAN/2) [206].

At the receiver, the signal of each receive antenna is OFDM demodulated. Figure 5.12 shows the example of a one-receiver based system. For the two-receiver system two FFT blocks are required at the receiver, thus increasing its complexity. The demodulated signals of the receiver antennas are then fed to the space-time block decoder of Figure 5.12. The space-time decoders apply the Logarithmic Maximum A-Posteriori (Log-MAP) decoding algorithm [74] for providing soft outputs for the channel decoders. The LDPC code can be decoded by the so-called sum-product algorithm [210] or by belief propagation [211]. The decoded bits are then fed to the video decoder.

For our simulations, we used the multi-path channel model characterised by the UTRA vehicular channel A. The corresponding channel impulse response is shown in [183], where each path is faded independently according to the Rayleigh distribution. Table 5.6 shows the corresponding transceiver parameters and channel parameters employed.

5.4.4 Simulation Results and Discussions

In this section, we provide our simulation results for the modulation schemes specified in Table 5.5 using the space-time block coded OFDM modem concatenated with LDPC coding, when transmitting over the UTRA channel detailed in Table 5.6. We will first characterise the achievable FER versus channel SNR performance expressed in dB, followed by the video quality results.

5.4.4.1 FER Using Fix-mode Modulation Schemes

Initial simulations were performed using the transceiver configured in the four fixed modulation modes of Table 5.5. Figure 5.14 portrays the achievable FER for each of the modulation modes employed

versus the channel SNR.

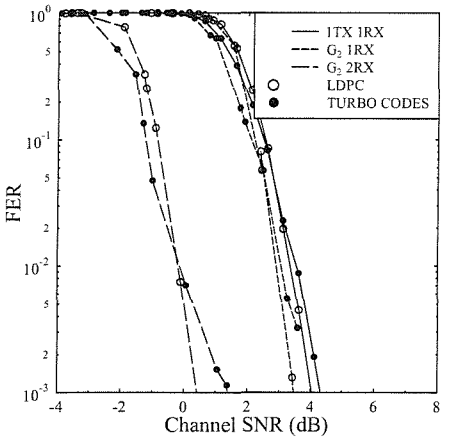
We can see from Figure 5.14 that the performance of the system improved, although only slightly, when we employed the $\mathbf{G}_2$ space-time block code using one receiver antenna. The reason for this modest improvement was that when using two transmit antennas, the power of the individual antennas has to be halved for the sake of fair comparison. On the other hand, the attainable improvement increased significantly, when two receiver antennas were used. This is because, the employment of two receiver antennas allows the space-time block codes and the channel decoder to decode the incoming frames more reliably, thus significantly reducing the FER even at a low SNR. However, this improvement comes at the expense of an increased receiver complexity, since twice as many FFT blocks will be required.

As we reported in [203], it is shown in Figure 5.14 that LDPC codes may perform better than turbo convolutional codes (TC) in terms of the achievable FER, especially at high coding rates. In the investigated scenario the ratio of the estimated complexity of the LDPC codes of Table 5.5 to the TC(2,1,4) code's complexity was $1125/2576 \approx 0.44$, which was estimated for 25 LDPC iterations and 8 TC iterations. In other words, the LDPC codec's complexity is about half of that of the TC(2,1,4) code. For example, in Figure 5.14(d), when using the 5/6 rate-coded 64-QAM, LDPC coding has a 2.5, 4.5 and 3.5 dB better SNR than TC at a FER of 10^{-3} in the STBC transceiver using two receiver antennas, one receiver antenna and no STBC, respectively. The differences are smaller in the 1/2 rate-coded 4-QAM situation characterised in Figure 5.14(a), where LDPC coding performs better than TC by about 1, 0.5 and 0.5 dB at a FER of 10^{-3} in the context of the STBC system using two receiver antennas, one receiver antenna and no STBC, respectively. Therefore, the employment of LDPC is more beneficial in video telephony, where the video codec is very sensitive to the transmission errors experienced in a frame.

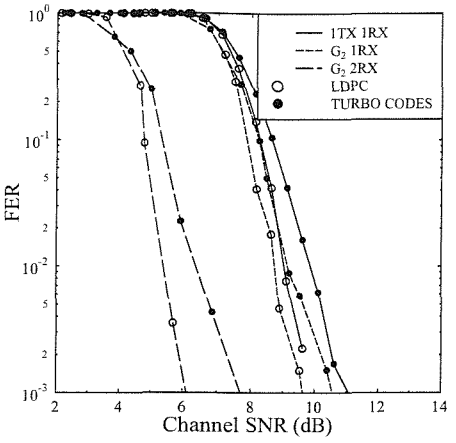
modulation

5.4.4.2 Video Performance

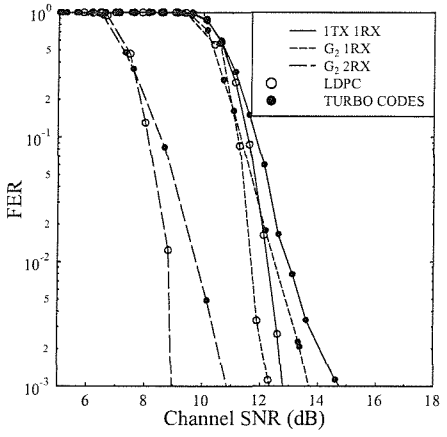
The various transmission scenarios were also studied in terms of their video PSNR performance using the LDPC assisted $\mathbf{G}_2$ space-time block coded OFDM system assisted by one and two receiver antennas for transmission over the UTRA channel. Experiments using one transmitter and one receiver antenna were also conducted for the sake of comparison. Figure 5.15 shows the achievable video quality expressed in terms of the PSNR versus the channel SNR for each of the modulation modes of the system. As expected, the higher throughput bitrate of the 64QAM mode provides an inherently better video quality than that of the lower-throughput but more robust video modes. However, as the channel quality degrades, the attainable video quality of the 64QAM mode degrades rapidly. In our previous work [4], it has found that in order to ensure a tolerable degradation of the video quality



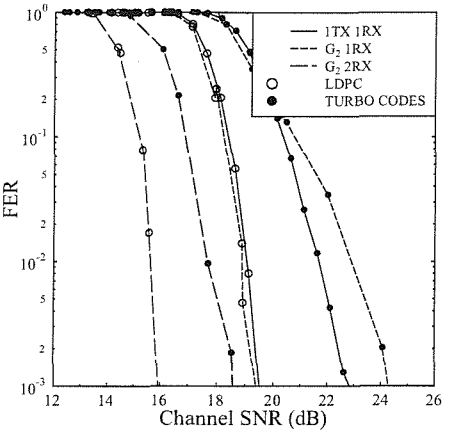
(a) 1/2 Coded 4-QAM



(b) 2/3 Coded 8-PSK



(c) 3/4 Coded 16-QAM



(d) 5/6 Coded 64-QAM

Figure 5.14: FER versus channel SNR performance for (a) 1/2-rate coded 4-QAM, (b) 2/3-rate coded 8-PSK, (c) 3/4-rate coded 16-QAM, and (d) 5/6-rate coded 64-QAM using the LDPC and TC assisted $\mathbf{G}_2$ space-time block coded OFDM system employing 1 and 2 receiver antennas for transmission over the UTRA channel. A one-transmitter and one-receiver antenna aided benchmarker system is also shown for comparison.

as the channel SNR reduced, it was beneficial to switch to a more robust modulation mode, when the FER exceeded 5%. Although this inherently reduced the video bitrate and the associated video PSNR, this was less objectionable in subjective video quality terms, than a FER in excess of 5% would have been.

As discussed in Section 5.4.4.1, the system using LDPC codes performs better than the one that employs TC, hence we concentrate our attention on the LDPC coded video scenario. Three transmission schemes have been characterised in Figure 5.15, which are the system using one transmitter and one receiver, one transmitter as well as two receivers, and two transmitters as well as two receivers. It can be seen from Figure 5.15 that the PSNR video performance curves are similar for the 1TX, 1RX and for the $\mathbf{G}_2$, 1RX transmission systems, but both of them are outperformed by the two-transmitter and two-receiver system.

For the sake of comparison, Figures 5.16 and 5.17 show the video performance in terms of both PSNR as well as FER versus channel SNR for the transmission schemes using LDPC assisted $\mathbf{G}_2$ space-time block coded OFDM in conjunction with one and two receiver antennas, respectively. The corresponding PSNR and FER results of the video transmission Systems 1 and 2 operating with and without transmission feedback, respectively are also depicted in both figures. As seen from Figures 5.16 and 5.17, Systems 1 and 2 have a similar performance. For each of the modulation modes, System 1, which invokes transmission feedback has a marginally lower video quality, when the FER is relatively low, i.e. the SNR is high. However, it has a slightly better video quality than that of System 2, which was using no transmission feedback, when the FER is high.

Observe furthermore that the video quality degradation of System 1 is more visible in the lower throughput modulation mode of 4QAM, than in the 64QAM mode. As shown in Table 5.5, in our experiments we use 9 bits for the repedition coded transmission feedback. By contrast, in System 2 using no transmission feedback, an extra 9 bits have been added to the video bitrate budget. In the case of 4QAM there are 240 bits per packet, and 9 bits constitute a relatively higher portion of the total payload of a packet in the 4QAM mode, than in 64QAM, which transmits 1200 bits per packet.

5.4.5 Conclusions on LDPC and TC Assisted STBC OFDM Wireless Video

An LDPC and turbo coding assisted $\mathbf{G}_2$ space-time block coded OFDM video transceiver was proposed. In the 4QAM mode a reasonable video quality can be achieved at low SNRs. By contrast , the video quality can be substantially increased in the 64 QAM mode, when the channel SNR is sufficiently high. No significant performance difference was observed between System 1 and 2, indicating that the employment of transmission feedback is not necessitated.

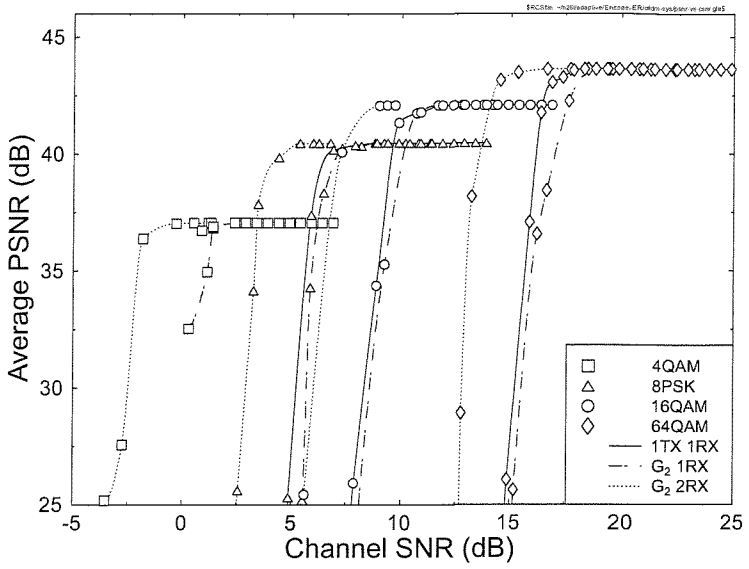


Figure 5.15: Average PSNR versus channel SNR for the four fixed OFDM modes using the LDPC assisted $\mathbf{G}_2$ space-time block coded OFDM System 1 assisted by one and two receiver antennas for transmission over the UTRA channel. The performance of a one transmitter and one receiver antenna aided system is also shown for comparison. The Miss America QCIF video sequence scanned at 30 frame/s was used in our experiments.

5.5 Chapter Conclusions

In Sections 5.2 and 5.3 joint-detection assisted near-instantaneously adaptive CDMA video transceivers were designed, which allow us to switch between a set of different source and channel codec modes as well as transmission parameters, depending on the overall instantaneous channel quality. Various coded modulation schemes, such as BICM, TCM and TTCM were employed, and TTCM exhibited the best performance in the H.263-aided scheme of Section 5.2. In Section 5.3, the H.264 video codec was employed in conjunction with LDPC channel coding. Note however that, we also used the H.263 video codec in conjunction with TTCM for the sake of fair comparison. It became apparent from this comparative study that under similar conditions the H.264 assisted scheme outperformed H.263 aided system. The LDPC and turbo coding assisted $\mathbf{G}_2$ space-time block coded OFDM video transceiver was investigated in Section 5.4. The transmission parameters have been partially harmonised with the UMTS-like systems of Sections 5.2 and 5.3. More specifically , we employed the H.264 video codec at the effective video bit rates of $20.4, 44.3, 68.3, 116.2$ kbit/s. We found that the LDPC codec was capable of outperforming the turbo coded benchmarker by about 1, 0.5 and 0.5 dB at a FER of 10^{-3} in the context of the STBC system using two receiver antennas, one receiver antenna and no STBC,

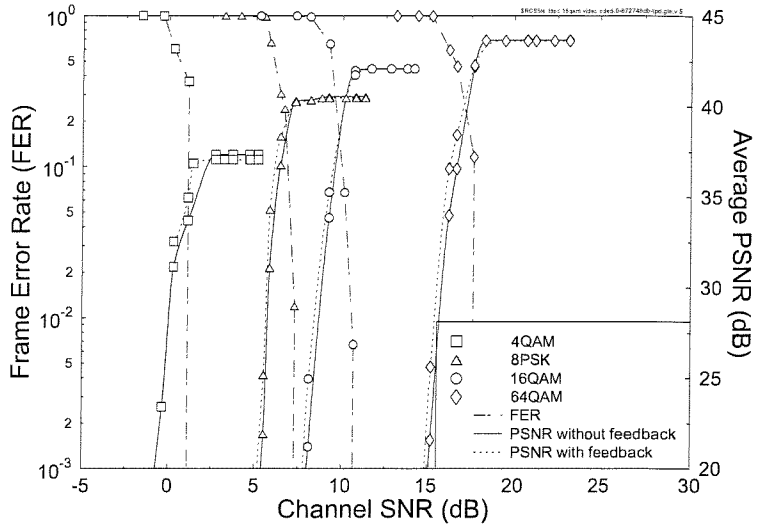


Figure 5.16: Average PSNR and FER versus channel SNR for the four fixed modulation modes using the LDPC assisted $\mathbf{G}_2$ space-time block coded OFDM system relying on one transmitter and two receiver antennas for transmission over the UTRA channel. The QCIF Miss America video sequence scanned at 30 frame/s was used in our experiments.

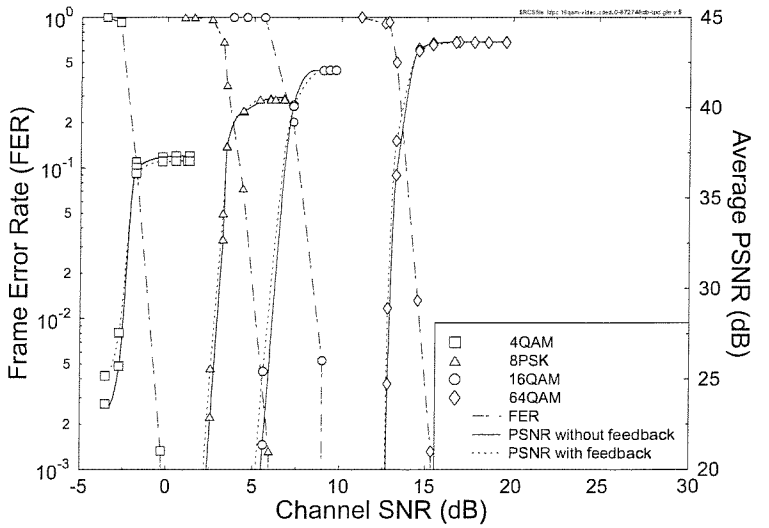


Figure 5.17: Average PSNR and FER versus channel SNR for the four fixed modulation modes using the LDPC assisted $\mathbf{G}_2$ space-time block coded OFDM system relying on two transmitters and two receivers antennas for transmission over the UTRA channel. The QCIF Miss America video sequence scanned at 30 frame/s was used in our experiments.

respectively. Therefore, the employment of LDPC coding worked out to be more beneficial in video telephony, the context of the investigated video system.

Chapter 6

MPEG-4 Video Transceivers

6.1 Introduction

In this chapter we consider several video transceivers employing the MPEG-4 video codec of Chapter 2. Figure 6.1 shows the simplified video communication system's block diagram, constituted by a source encoder, inner and outer channel encoder, modulator, transmit and receive diversity, demodulator, inner and outer channel decoder and source decoder.

In the forthcoming sections, we propose several MPEG-4 video transceivers. Specifically, an iterative parallel interference cancellation aided CDMA video telephony scheme is studied in Section 6.2, while a more sophisticated space-time spreading assisted multi-carrier DS-CDMA arrangement is the subject of Section 6.3. Finally, turbo-detected unequal error protection MPEG-4 video telephony using a serially concatenated convolutional outer code, trellis coded modulation based inner code and space-time coding is studied in Section 6.4. Note that throughout our system studies, we employ the MPEG-4 source codec operating at a constant target bit rate of 69 kbps. By contrast, we adopt numerous different channel coding and modulation schemes in Sections 6.2 to 6.4. However, within all of these systems, the overall transmission bitrate is adjusted to be 150 kbps so that the achievable video performance becomes comparable, allowing us to draw conclusions concerning the robustness of the different transceivers.

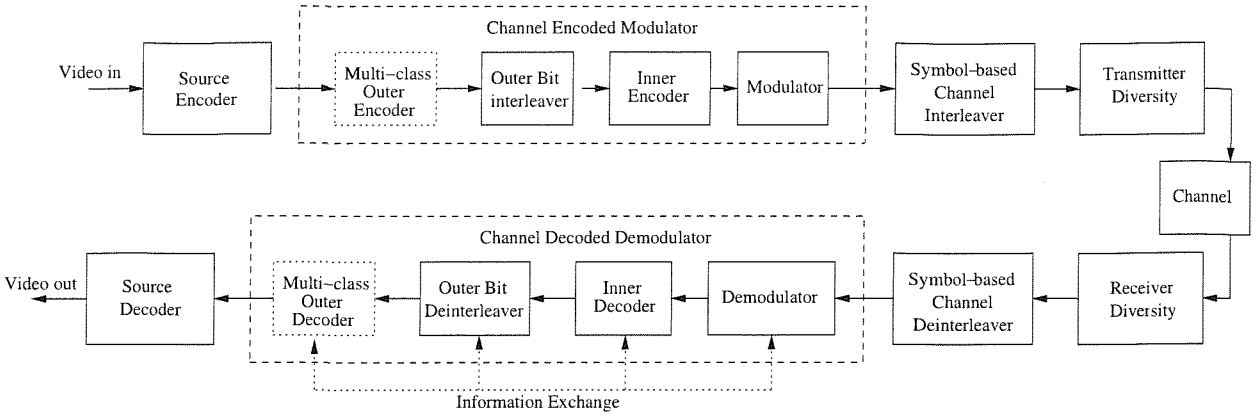


Figure 6.1: Generic block diagram of our proposed video transceiver. We will demonstrate that in its most sophisticated form the receiver of Figure 6.22 exchanges information amongst all decoder blocks in an effort to enhance the achievable performance.

6.2 Iterative Parallel Interference Cancellation Aided CDMA Based MPEG-4 Video Telephony

[1]

6.2.1 Introduction

The Motion Pictures Expert Group's MPEG-4 codec was designed for object-based audiovisual coding, invoking numerous different 'coding tools' [212], as described in Section 2.1. The so-called Video Objects (VO), which are not necessarily rectangular-shaped blocks, are encoded and transmitted separately. The VOs are reconstructed at the decoder, allowing the receiver to choose how a video scene is presented.

Conventional video sequences are coded in MPEG-4 much as they were in the superseded MPEG-1 and MPEG-2 coding schemes. More explicitly, in its classic encoding mode the MPEG-4 codec uses block-based Motion-Compensation (MC) and the MC error residual texture is Discrete Cosine Transformed (DCT) before its encoding. In Chapter 4, we highlighted various MPEG-4 error resilience issues [80], rendering it suitable for use in error-prone environments, such as wireless video telephone applications.

In reference [213], a Parallel Interference Cancellation (PIC) based iterative Multiuser Detector

[1]This section is based on J. Y. Chung, H. Wei and L. Hanzo: Iterative Parallel Interference Cancellation Aided CDMA Based MPEG-4 Video, submitted to and it was based on collaborative research with the co-authors.

(MUD) was proposed, which was further developed in [214], employing numerous sophisticated turbo channel coding schemes. The iterative receiver advocated in this treatise obeys a similar structure to that proposed in [213], except for employing half-rate turbo channel coding. In [215] and [216] the proposed iterative multiuser detectors invoked soft estimation of the real-valued BPSK modulated signal. The Log-MAP turbo decoder [217] was employed and the Recursive Systematic Convolutional (RSC) code's memory length was $L = 3$.

This section is organised as follows. In Subsection 6.2.2 we briefly introduce the concept of the PIC scheme, while in Subsection 6.2.3 we provide a brief MPEG-4 video error resilience overview. In Subsection 6.2.4 we characterise the achievable performance. Finally, in Subsection 6.2.5 we offer our conclusions.

6.2.2 Iterative Multiuser Detection Assisted DS-CDMA

6.2.2.1 Turbo Multiuser Detection

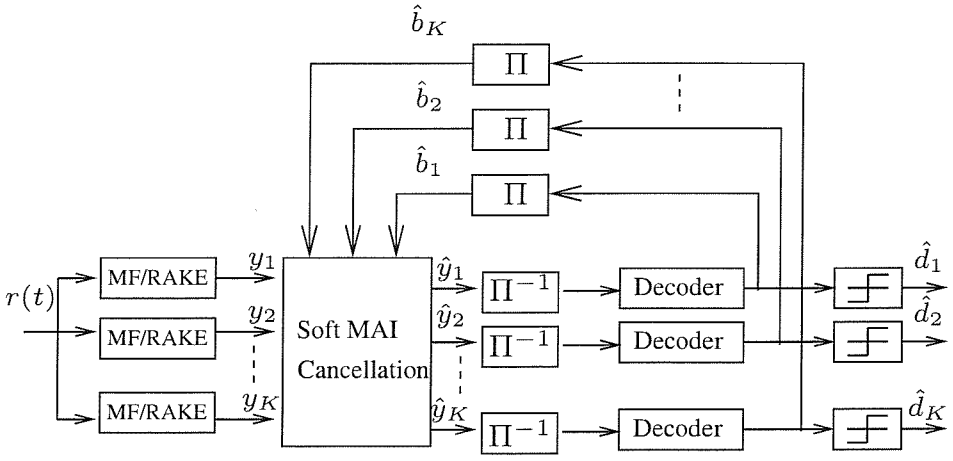


Figure 6.2: Schematic of the iterative PIC-aided detector.

Figure 6.2 shows the schematic of the PIC assisted iterative multiuser detector designed for DS-CDMA communications, where the Matched Filter's (MF) output related to the k users' received signal is given by: $\mathbf{y} = [y_1, ..., y_K]^T$.

Let us assume that the turbo channel decoder generates the received symbol probability $\mathbf{Pr}\{b_k = x_m|\hat{\mathbf{y}}_k\}$, $k = 1 \ldots K, m = 1 \ldots M$, where M is the size of the M-array modulated signal constellation. Then the estimated symbol b_K of user k at a particular time instant may be expressed at the output of the soft Multiple Access Interleaver (MAI) cancellation block of Figure 6.2 as:

$$\hat{b}_k = \sum_{m=1}^{M} \mathbf{Pr}\{b_k = x_m|\hat{\mathbf{y}}_k\} \cdot x_m, \tag{6.1}$$

where $x_m, m = 1 \ldots M$ represents the complex-valued M-array symbols. We define the vector $\hat{\mathbf{b}} = [\hat{b}_1, ..., \hat{b}_K]^T$, seen at the top of Figure 6.2, which contains the estimated symbol of all the K users. The blocks Π and Π^{-1} in Figure 6.2 denote the channel interleaver and deinterleaver, respectively, which arrange the bits at the input of the channel decoders and the MAI canceller in the required order. Once we obtained the soft symbol estimates, soft MAI cancellation is performed, as shown in Figure 6.3, yielding the decontaminated soft output signal vector $\hat{\mathbf{y}} = [\hat{y}_1, ..., \hat{y}_K]^T$. Figure 6.3 shows explicitly that the soft estimates $\hat{b}_k, \quad k = 1, \ldots, K$ of the transmitted symbols, which are output by the channel decoder, are utilised for reconstructing the estimates of the transmitted signal of all the K users. Then, for each user, the reconstructed estimated signals of all the interfering users are subtracted from the composite multiuser signal and the resultant decontaminated signal is processed by a Matched Filter (MF) or RAKE receiver, generating the soft output $\hat{\mathbf{y}}_k$. These modulated signal reconstruction, interference cancellation and re-estimation steps are repeated, until the affordable number of iterations is exhausted.

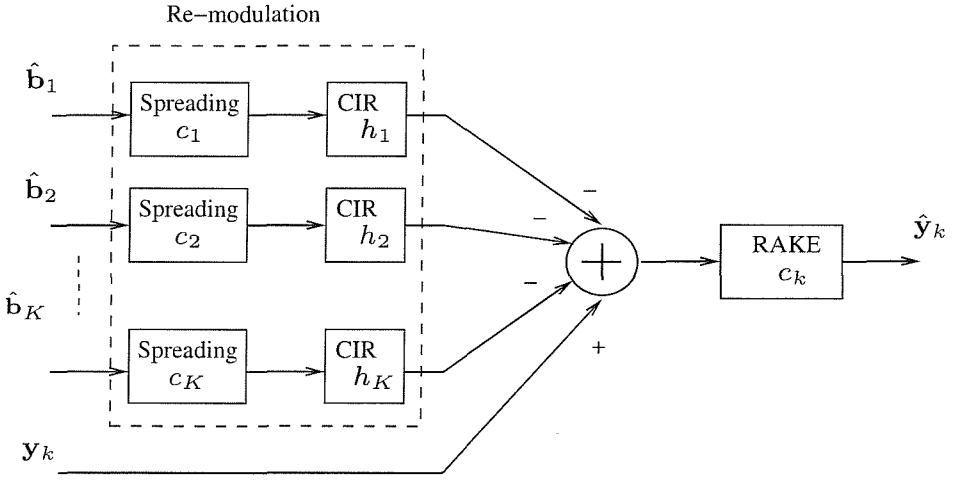


Figure 6.3: Schematic of a single PIC cancellation stage.

In this system, half-rate turbo codes [218] are employed. The Log-MAP turbo decoder [217] was employed and the RSC code's memory length was $L = 3$. However, our initial investigations indicated that no significant performance improvements were achieved, when more than four PIC-based outer iterations were invoked. Furthermore, within each outer PIC iteration, two turbo decoding iterations were employed. The family of m-sequence based spreading codes having a length of $N = 15$ was employed in our system and the transmission burst length was $T = 120$ bits. A random channel interleaver denoted by Π in Figure 6.2 and a memory of eight transmission bursts was employed in our system. The system considered supported $K = 7$ users and communicated over an asynchronous equal-weight symbol-spaced two-path uncorrelated Rayleigh fading channel.

6.2.2.2 Asynchronous Interference Cancellation

In an asynchronous system the symbols transmitted during the ith signalling interval are also affected by all the asynchronous interfering signals transmitted during the intervals $(i+1)$ and $(i-1)$ owing to the dispersed MUI imposed by other users. Hence, according to [219], for an asynchronous DS-CDMA system communicating over a non-dispersive AWGN channel the received signal vector $\mathbf{y}$ can also be expressed as:

$$\mathbf{y}[i] = \mathbf{R}^T[1]\mathbf{A}\mathbf{b}[i+1] + \mathbf{R}[0]\mathbf{A}\mathbf{b}[i] + \mathbf{R}[1]\mathbf{A}\mathbf{b}[i-1] + \mathbf{n}[i], \tag{6.2}$$

where i is the index of time instant and the zero-mean Gaussian channel noise process $\mathbf{n}[i]$ has the crosscorrelation matrix:

$$E[\mathbf{n}[i]\mathbf{n}^T[j]] = \begin{cases} \sigma^2\mathbf{R}^T[1], & \text{if } j = i+1; \\ \sigma^2\mathbf{R}[0], & \text{if } j = i; \\ \sigma^2\mathbf{R}[1], & \text{if } j = i-1; \\ \mathbf{0}, & otherwise, \end{cases} \tag{6.3}$$

while the matrix $\mathbf{R}[1]$ and $\mathbf{R}[0]$ are defined as:

$$\mathbf{R}_{jk}[0] = \begin{cases} 1, & \text{if j = k;} \\ \rho_{jk}, & \text{if } j < k; \\ \rho_{kj}, & \text{if } j > k, \end{cases} \tag{6.4}$$

$$\mathbf{R}_{jk}[1] = \begin{cases} 0, & \text{if } j \geq k; \\ \rho_{kj}, & \text{if } j < k. \end{cases} \tag{6.5}$$

Let us now define the vectors $\hat{\mathbf{b}}_k[i+1], \hat{\mathbf{b}}_k[i], \hat{\mathbf{b}}_k[i-1]$ as the soft estimates of the kth user during the signalling instants $(i+1)$, i, and $i-1$, which have had their kth element set to zero. Therefore, after the soft MAI cancellation stage of Figure 6.2, the decontaminated signal vector $\hat{\mathbf{y}}_k[i]$ can be represented as:

$$\begin{aligned} \hat{\mathbf{y}}_k[i] &= \mathbf{R}^T[1]\mathbf{A}(\mathbf{b}[i+1] - \hat{\mathbf{b}}_k[i+1]) \\ &\quad + \mathbf{R}[0](\mathbf{b}[i] - \hat{\mathbf{b}}_k[i]) \\ &\quad + \mathbf{R}[1](\mathbf{b}[i-1] - \hat{\mathbf{b}}_k[i-1]). \end{aligned} \tag{6.6}$$

According to Equation 6.6, we can observe that the decontaminated signal vector $\mathbf{y}[i]$ will be free from multiple access interference, provided that the symbol estimates $\hat{\mathbf{b}}_k[i+1], \hat{\mathbf{b}}_k[i], \hat{\mathbf{b}}_k[i-1]$ are reliable. The performance of this PIC-aided transceiver was characterised in [214] in conjunction with a suite of channel codecs. In our two-path dispersive channel the effects of the dispersed asynchronous MUI are also aggravated by channel-induced Inter-Symbol Interference (ISI).

pages

6.2.3 MPEG-4 Error Resilience Tools

The video compression scheme adopted in this system is the MPEG-4 codec, which was studied in Chapter 2. Recall from our bit sensitivity studies of Section 4.7 that owing to the employment of Variable Length Codes (VLC) in the MPEG-4 standard, the compressed video bitstreams are particularly sensitive to channel errors [4]. This is, because the boundary between VLC words is not explicit. Transmission errors typically lead to an incorrect number of bits being used during the VLC decoding process, inflicting loss of synchronisation. As a further consequence of VLC encoding, the location where the decoder detects an error in the MPEG-4 encoded bitstream is not the same, as the location where the error has actually occurred. Fortunately, the error resilience tools employed in the MPEG-4 video standard assist in minimising the amount of data that has to be discarded, whenever errors are detected. We have highlighted the basic philosophy of the MPEG-4 error resilience tools in Section 4.6, noting that the following error-resilience tools were included in the codec [80]: resynchronisation markers, data partitioning, Header Extension Codes (HEC) and Reversible Variable Length Codes (RVLC).

Apart from the tools mentioned above, error concealment [55] has also been implemented in the decoder. Finally, for the sake of limiting the error propagation across consecutive video frames, intra-frame coded images are transmitted at regular intervals. Let us now quantify the achievable system performance in next section.

6.2.4 Experimental Results

The performance of the proposed MPEG-4 based video telephone scheme is characterised in this section in terms of the average video Peak Signal to Noise Ratio (PSNR) versus Channel SNR. Our simulations were carried out using the (174×144)-pixel Quarter Common Intermediate Format (QCIF) "Miss America", "Foreman" and "News" video sequences encoded at a video bitrate of 69 kbps. The MPEG-4 source encoder operated at the video frame rate of 30 frames per second. Table 6.1 summarises the modulation and channel parameters employed. Figure 6.4 depicts the MPEG-4 codec's video performance in the context of the PIC aided CDMA system considered, expressed in terms of the average PSNR versus the Signal to Noise Ratio (SNR), namely E_b/N_o for transmission over an uncorrelated symbol-spaced two-path fading channel. By contrast, Figure 6.5 plots the PSNR results from a different perspective, namely as a function of the Bit Error Rate (BER). On the other hand, Figure 6.6 shows the BER versus channel SNR performance of our proposed scheme.

Parameter	Value
Channel type	Two-path, symbol-spaced, equal-weight Rayleigh fading Channel Impulse Response (CIR)
Doppler frequency	100Hz
Spreading factor (m-seq.)	15
Receiver type	Iterative PIC
Modulation type	QPSK
Channel code	1/2-rate turbo code
Transmission burst length	120 bits
No. of PIC iterations	4
No. of turbo decoder iterations	2
No. of users	7
Video codec	MPEG-4
Video frame-rate	30 frame/sec
Video format	176 × 144 QCIF
Video bitrate	69kbps
Channel coded bitrate	150kbps
Interleaver length	960 bit

Table 6.1: Modulation and channel parameters

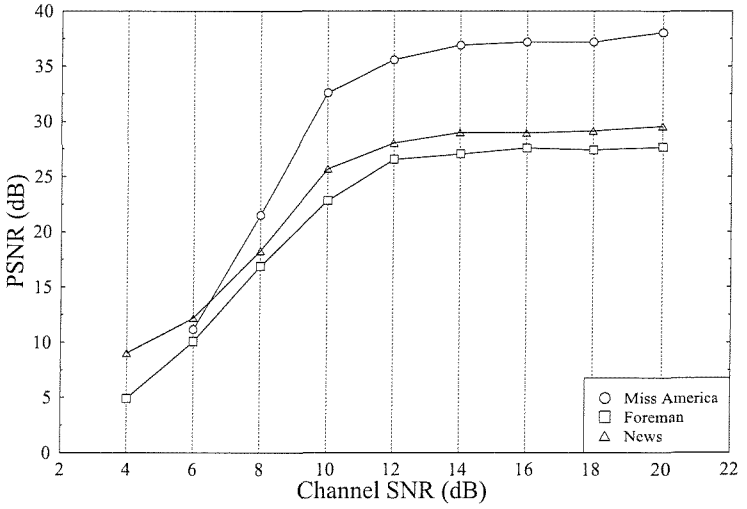


Figure 6.4: Average PSNR versus channel SNR performance of the PIC assisted MPEG-4 videophone scheme, when supporting $K = 7$ users for transmission over an uncorrelated symbol-spaced, equal-weight two-path fading channel employing the parameters of Table 6.1.

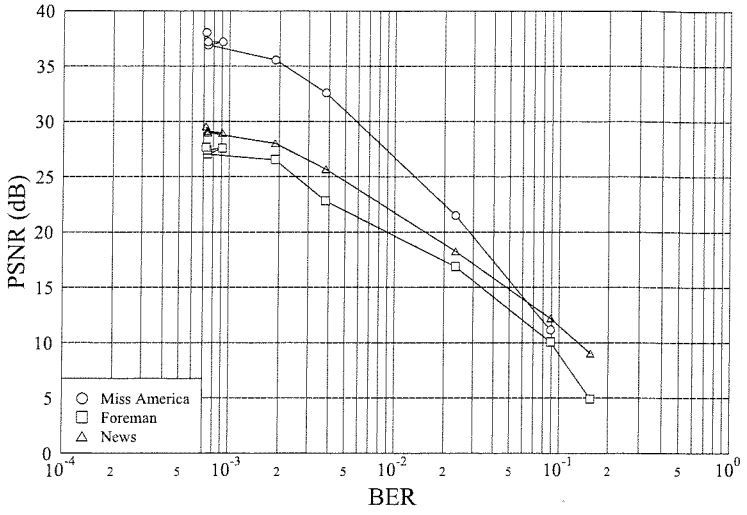


Figure 6.5: Average PSNR versus channel BER performance of the PIC assisted MPEG-4 videophone scheme for $K = 7$ and transmission over an uncorrelated symbol-spaced, equal-weight two-path fading channel employing the parameters of Table 6.1.

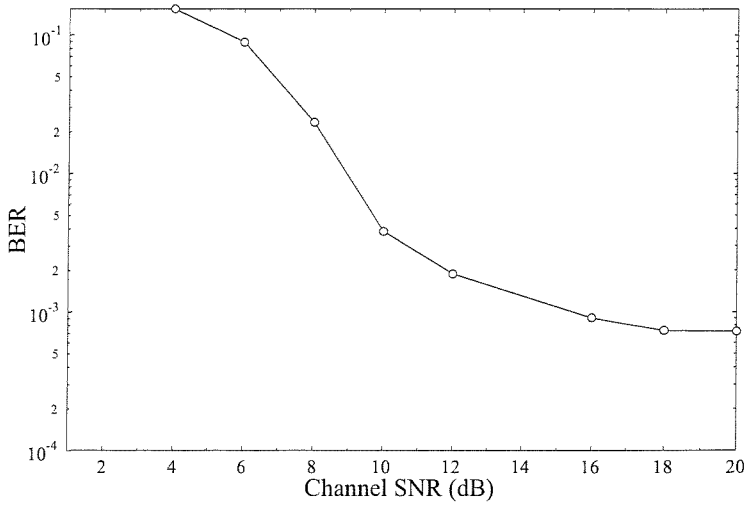
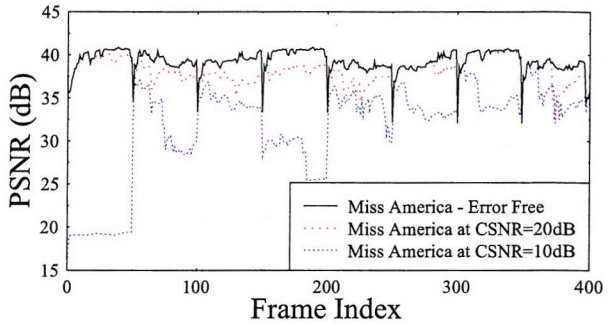
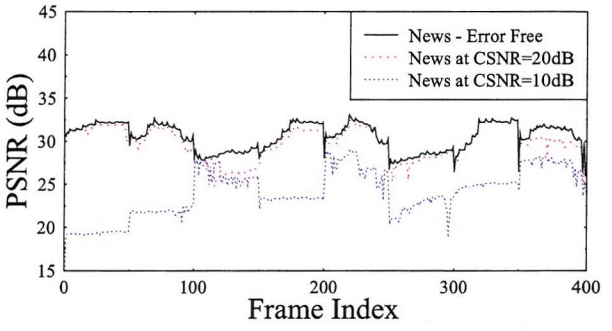


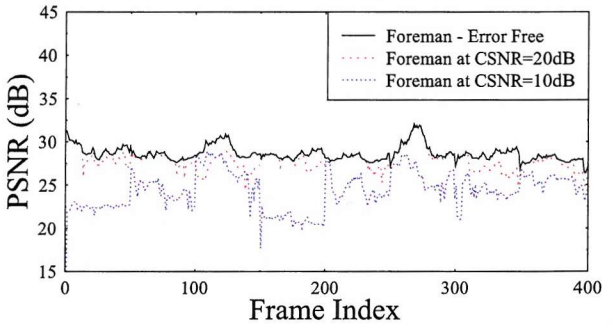
Figure 6.6: Channel BER versus channel SNR performance of the PIC assisted MPEG-4 videophone scheme for $K = 7$ and transmission over an uncorrelated symbol-spaced, equal-weight two-path fading channel employing the parameters of Table 6.1.



(a) Miss America Video Sequence

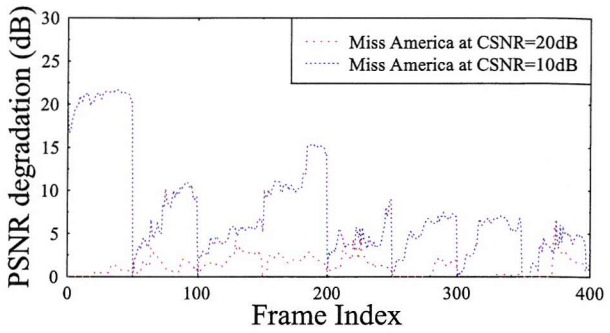


(b) News Video Sequence

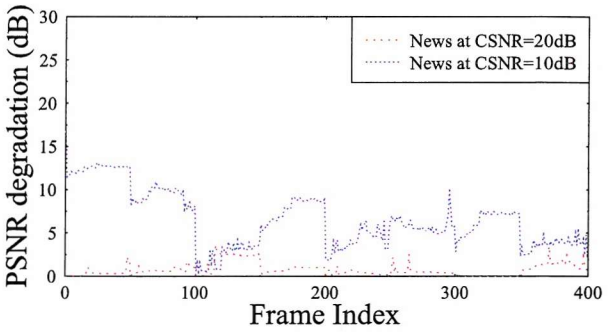


(c) Foreman Video Sequence

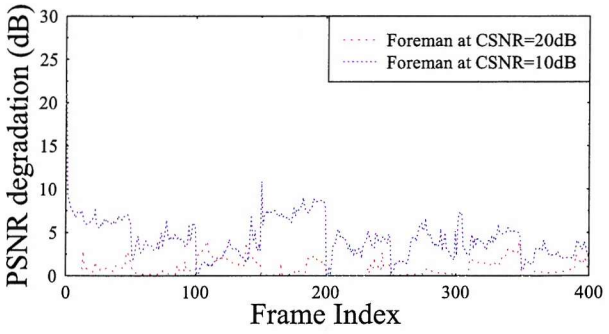
Figure 6.7: PSNR versus video frame index of the PIC assisted MPEG-4 videophone scheme, when supporting $K = 7$ users for transmission over an uncorrelated symbol-spaced, equal-weight two-path fading channel employing the parameters of Table 6.1.



(a) Miss America Video Sequence



(b) News Video Sequence



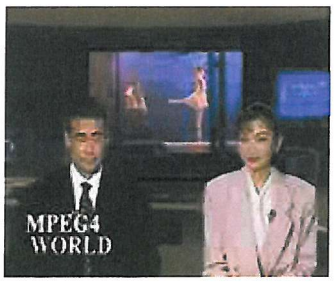
(c) Foreman Video Sequence

Figure 6.8: PSNR degradation versus video frame index of the PIC assisted MPEG-4 videophone scheme, when supporting $K = 7$ users for transmission over an uncorrelated symbol-spaced, equal-weight two-path fading channel employing the parameters of Table 6.1.

As seen in both Figure 6.4 and 6.5, at a given E_b/N_o or BER, the "Foreman" sequence resulted in the lowest PSNR image quality, followed by the "News" sequence and finally by the "Miss America" sequence. This is, because the "Miss America" sequence is a pure 'head and shoulders' type video sequence, which has the lowest motion activity among the three clips. Therefore less motion vectors have to be transmitted, potentially allowing the MPEG-4 codec to dedicate a larger fraction of the available bitrate budget to the representation of the fine details of texture-encoding, which increases the achievable PSNR. Even if some of the "Miss America" bistream is corrupted, the corrupted VOPs can be readily concealed by the previous VOP of the preceding video frame. The video performance of the proposed scheme is further illustrated in Figure 6.7, characterising of each of the consecutive video frames' video quality. The PSNR value recorded at the channel SNR of 20dB (virtually error-free transmission scenario) and at the channel SNR of 10dB were shown in Figure 6.7. The subjective video quality at SNR=10dB becomes somewhat objectable. Figure 6.8 portrays the PSNR degradation versus video frame index for the same system.

Figure 6.9 illustrates the subjective video quality of a particular frame of the three video sequences at E_b/N_o of 20dB, where the average PSNR values of the "Miss America", "News" and "Foreman" sequences are 38.05 dB, 29.55 dB and 27.66 dB, respectively.

(a)



(b)

(c)

Figure 6.9: 10th Frame of the (a) "Miss America", (b) "News" and (c) "Foreman" sequences when supporting $k = 7$ users for transmission over an uncorrelated fading channel at E_b/N_o of 20dB

6.2.5 Conclusions on the PIC-Aided CDMA Transceiver

In this section, a PIC-aided CDMA scheme was investigated, employing the MPEG-4 video codec. It was shown in the context of the PIC-CDMA MPEG4 video telephony scheme that at the video coding bitrate of 69kbps, the proposed system was capable of achieving an adequate video quality. When $K = 7$ users were supported using 15-chip PN-codes, the minimum required channel SNR was about 20 dB, when communicating over a disperssive two-path, sample-spaced, equal-weight having a Doppler frequency of 100Hz. The remaining system parameters were summarised in Table 6.1.

6.3 Space-Time Spreading Assisted Multicarrier DS-CDMA Video Telephony

[2]

6.3.1 Introduction

Having investigated the achievable video performance of an iterative PIC-aided CDMA based MPEG-4 video transceiver, in this section, we study a more sophisticated Space-Time Spreading (STS) assisted multicarrier CDMA (MC-DSCDMA) [220] video transceiver. When the STS scheme is invoked for spreading the signal of each sub-carrier, the fading of each subcarrier is mitigated and hence the system becomes capable of significantly reducing the effects of the time-variant channel fading. Furthermore, turbo convolutional coding is invoked for enhancing the achievable BER versus SNR performance.

6.3.2 System Outline

Let us now consider the architecture of the STS [214] assisted MC DS-CDMA scheme [214,221] using U number of subcarriers and T_x transmitter antennas, but only a single receiver antenna. A synchronous MC DS-CDMA scheme designed for the downlink is considered, where the K users' signals are transmitted synchronously. The transmitter schematic of the kth user is shown in Figure 6.10, where the transmission of complex data symbols using QAM modulation and real-valued spreading [214, 222] was considered. As shown in Figure 6.10, at the transmitter a block of $\sum_{u=1}^{U} \eta_u \cdot L_x$ number of data bits having a bit duration of T_b is Serial-Parallel (S-P) converted to U parallel sub-blocks, where η_u is the number of Bits Per Symbol (BPS) transmitted by the QAM scheme on the uth subcarrier and L_x is the number of QAM symbols per space-time spreading sub-block. For example, if BPSK and 64QAM are invoked on the uth subcarrier, the corresponding value of η_u will be 1 and 6, respectively.

[2]This section is based on collaborative research with H. Wei and L. Hanzo

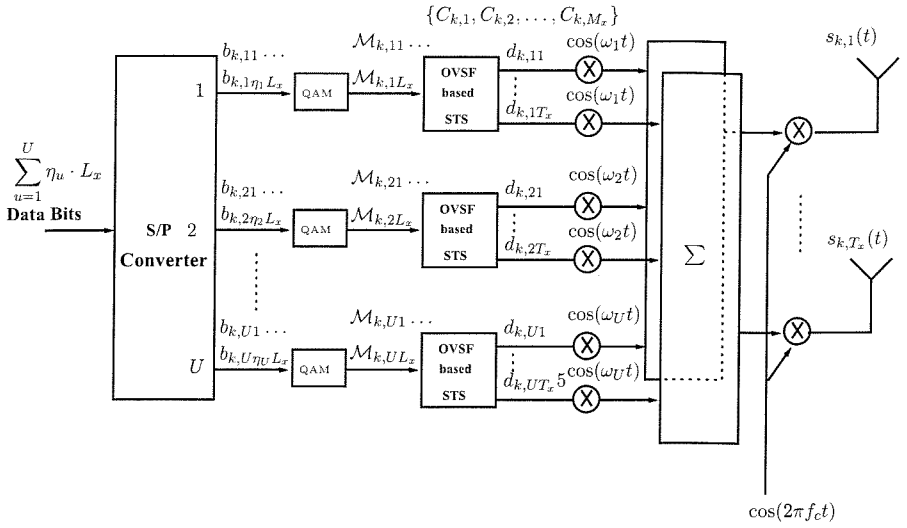


Figure 6.10: Schematic of the OVSF STS-aided MC-CDMA transmitter

Hence, each space-time spreading sub-block has $\eta_u L_x$ data bits, generating L_x modulated symbols given by $\{\mathcal{M}_{k,u1}, \ldots \mathcal{M}_{k,uL_x}\}$. These modulated symbols are space-time spread using the schemes of [214, 222] with the aid of M_x number of OVSF STS spreading codes, for example by the codes $\{c_{k,1}(t), c_{k,2}(t), \ldots, c_{k,M_x}(t)\}$, $k = 1, 2, \ldots, K$ and mapped to T_x number of transmitter antennas. Since the bits to be transmitted are mapped to the multi-bit QAM symbols as well as to T_x number of transmit antennas, the symbol duration of the STS signals is expanded to UL_xT_b, and the number of chips of the OVSF codes becomes $UL_xT_b/T_c = UL_xN$, where we have $N = T_b/T_c$, while T_c represents the chip-duration of the orthogonal spreading codes. The OVSF STS codes assume the form of

$$c_{k,i}(t) = \sum_{j=0}^{UL_xN-1} c_{k,i}[j] P_{T_c}(t - jT_c), \tag{6.7}$$

where we have $c_{k,i}[j] \in \{+1, -1\}$. The OVSF STS codes obey the relationship of $\sum_{l=0}^{UL_xN} c_{i,m}[l] c_{j,n}[l] = 0$, whenever we have $i \neq j$ or $m \neq n$. Furthermore, $P_{T_c}(t)$ represents the chip waveform defined over the interval of $[0, T_c)$. Since the maximum available number of the OVSF codes having UL_xN chips is UL_xN and each user requires M_x number of OVSF codes, the maximum number of users supported by these orthogonal codes is $\mathcal{K}_{max} = UL_xN/M_x$. As seen in Fig.6.10, following STS, each STS block generates T_x number of parallel signals to be mapped to the T_x number of transmitter antennas. Finally, the Inverse Fast Fourier Transform (IFFT) is invoked for carrying out multicarrier modulation [214, 223], and the IFFT block's output signal is transmitted using one of the transmitter antennas.

The operation of the STS scheme is described below. For example, for the case of $L_x = M_x =$

$T_x = 2$, the MC DS-CDMA signals transmitted by antenna 1 and 2 can be simply expressed as [214]:

$$\begin{aligned}\mathbf{s}_k(t) &= \begin{pmatrix} s_{k1}(t) \\ s_{k2}(t) \end{pmatrix} \\ &= \sqrt{\frac{2E_b}{4UT_b}} \begin{pmatrix} \sum_{u=1}^{U} [c_{k,1}\mathcal{M}_{k,u1} + c_{k,2}\mathcal{M}_{k,u2}] \cos[2\pi(f_c + f_u)t] \\ \sum_{u=1}^{U} [c_{k,1}\mathcal{M}_{k,u2} - c_{k,2}\mathcal{M}_{k,u1}] \cos[2\pi(f_c + f_u)t] \end{pmatrix}. \end{aligned} \tag{6.8}$$

By contrast, for the case of $L_x = M_x = T_x = 4$, the MC DS-CDMA signals transmitted by antenna 1, 2, 3 and 4, respectively, may be formulated as [214]:

$$\begin{aligned}\mathbf{s}_k(t) = \begin{pmatrix} s_{k1}(t) \\ s_{k2}(t) \\ s_{k3}(t) \\ s_{k4}(t) \end{pmatrix} &= \sqrt{\frac{2E_b}{16UT_b}} \\ \times \begin{pmatrix} \sum_{u=1}^{U} [c_{k,1}\mathcal{M}_{k,u1} + c_{k,2}\mathcal{M}_{k,u2} + c_{k,3}\mathcal{M}_{k,u3} + c_{k,4}\mathcal{M}_{k,u4}] \cos[2\pi(f_c + f_u)t] \\ \sum_{u=1}^{U} [c_{k,1}\mathcal{M}_{k,u2} - c_{k,2}\mathcal{M}_{k,u1} - c_{k,3}\mathcal{M}_{k,u4} + c_{k,4}\mathcal{M}_{k,u3}] \cos[2\pi(f_c + f_u)t] \\ \sum_{u=1}^{U} [c_{k,1}\mathcal{M}_{k,u3} + c_{k,2}\mathcal{M}_{k,u4} - c_{k,3}\mathcal{M}_{k,u1} - c_{k,4}\mathcal{M}_{k,u2}] \cos[2\pi(f_c + f_u)t] \\ \sum_{u=1}^{U} [c_{k,1}\mathcal{M}_{k,u4} - c_{k,2}\mathcal{M}_{k,u3} + c_{k,3}\mathcal{M}_{k,u2} - c_{k,4}\mathcal{M}_{k,u1}] \cos[2\pi(f_c + f_u)t] \end{pmatrix}. \end{aligned} \tag{6.9}$$

The specific spreading action of the STS scheme was illustrated in graphical terms in Figure 8.7 to 8.9 of [214, pp. 303-305]. The STS scheme constitutes a powerful fast-fading counter-measure, which is conveniently complemented by adaptive modulation techniques that excel in mitigating the effects slow shadow fading, rather than fast-fading [73]. Hence in the next section adaptive modulation is considered.

6.3.3 Adaptive QAM and STS Assisted MC DS-CDMA

6.3.3.1 Adaptive Mode-Switching and Learning for QAM

We considered a practically-motivated Adaptive QAM (AQAM) mode switching approach, which is based on the philosophy of *learning* the switching thresholds [224] in an on-line fashion so that the system becomes capable of maximising the achievable throughput expressed in terms of Bits Per Symbol (BPS). This learning schemes does not utilise the time-variant AQAM mode switching thresholds

designed by Torrance and Hanzo [225] using Powell's optimisation. The learning-based scheme also refrains from using analytical by computed switching thresholds, or any other assumptions concerning the operating environment. The only necessary side information required by this scheme [224] is the current estimated BER $p_e(\gamma)$ (or the Frame Error Rate (FER)) and the target BER P_{th} of the system. The basic philosophy is that when the current BER $p_e(\gamma)$ is lower than the target BER P_{th}, this indicates that the system is capable of improving the achievable BPS throughput and hence it will reduce the switching thresholds for the sake of maximizing the throughput. By the contrast, when the current BER $p_e(\gamma)$ is higher than the target BER P_{th}, then the system may be overwhelmed by transmission errors and hence the algorithm will increase the switching thresholds for the sake of reducing the system's throughput.

In our investigations we considered a five-mode AQAM scheme, associated with the AQAM mode switching threshold set of $s = \{s_1, s_2, s_3, s_4\}$ designed for BPSK, QPSK, 16QAM, 64QAM, respectively. For the sake of simplicity, we assumed that the AQAM threshold difference $(s_2 - s_1), (s_3 - s_1), (s_4 - s_1)$ between the different modulation schemes was the same as in Torrance's scheme [], and we denoted them by $\triangle_{21}, \triangle_{31}, \triangle_{41}$, respectively. In other words, once we obtained the switching threshold s_1, we knew all the switching thresholds. Similarly, once we adjusted the switching threshold s_1, all the other switching thresholds were considered to be adjusted. Controlling the threshold s_1 is based on the current BER performance $p_e(\gamma)$, where the switching thresholds can be adaptively updated as follows [224]:

$$\begin{aligned} s_1(t) &= s_1(t-1) + \mu \cdot \text{sign}[p_e(\gamma) - P_{th}], \\ s_2(t) &= s_1(t) + \triangle_{21}, \\ s_3(t) &= s_1(t) + \triangle_{31}, \\ s_4(t) &= s_1(t) + \triangle_{41}, \end{aligned} \tag{6.10}$$

where $0 < \mu < 1$ is the threshold control step-size. Perfect estimation of the current BER was assumed, although in practice the accurate estimation of the short-term 'in-burst-BER' of a transmission burst is a challenging task [73].

6.3.4 System Parameters and Channel Models

The simulation parameters used in our STS MC-DS CDMA system are listed in Table 6.2. Channel coding techniques are capable of substantially improving achievable performance of the communication systems, in particular the family of turbo codes. The basic configurations of the turbo codes used in conjunction with the various modems are summarised in Table 6.2. In our system, a burst-by-burst adaptive transmission scheme was employed, each transmission burst having a fixed number of symbols, namely $L = 240$. A half-rate turbo code having a memory of $v = 3$ was used in our

Parameter	Value
Channel type	Uncorrelated non-disperssive Rayleigh fading for each subcarrier
Spreading factor (OVSF)	32
No. of MC-CDMA subcarriers	16
Receiver type	RAKE/MF
Diversity	Space Time Spreading
No. of STS antennas	$T_x = 1, 2, 4$ and 8
Transmission burst length	120 bits
No. of users	32
Modulation type	BPSK, 4QAM, 16QAM, 64QAM
Channel code	1/2-rate turbo code
Channel coded bitrate	150kbps
Interleaver length	960 bit
No. of turbo decoder iterations	4
Video codec	MPEG-4
Video frame-rate	30 frame/sec
Video format	176 × 144 QCIF
Video bitrate	69kbps

Table 6.2: System parameters

proposed system. The QCIF MPEG-4 video coded bitrate was 69kbps at a frame scanning rate of 30 frames/sec., while the turbo-coded bitrate was 150 kbits/s. A two-path equal-weight symbol-spaced CIR was used. Again, all system parameters are summarised in Table 6.2.

6.3.5 Performance Results

Our experiments commence by investigating a fixed-mode scheme invoking a QPSK modem. In Figure 6.11, the BER performance of the half-rate turbo coded fix-mode QPSK system is presented using the various STS MC-CDMA transmitter schemes using $T_x = 1, 2, 4$ and 8 antennas for supporting 32 users. As expected, the scheme employing $T_x = 8$ transmitter antennas exhibits the highest BER performance, followed by the scheme having $T_x = 4$ transmitter antennas, etc.

For the sake of comparison, in the context of various video sequences having different motion activities, our simulations were carried out using the (174×144)-pixel Quarter Common Intermediate Format (QCIF) "Miss America", "Foreman" and "News" video sequences encoded at a near-constant video bitrate of 69 kbps. The corresponding video PSNR versus channel SNR performance is shown in Figure 6.12. Obverse that for the QPSK modulation scheme and a 1/2 coding rate, the $Tx = 1$ scheme is capable of achieving a perfect channel PSNR video performance at the channel SNR (CSNR) of

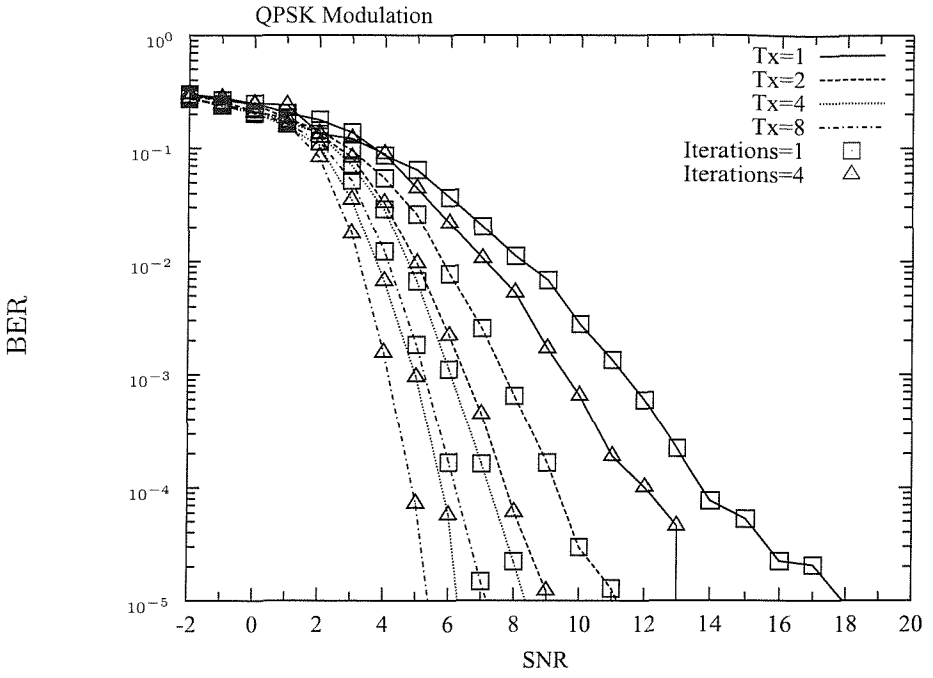


Figure 6.11: BER performance of STS assisted MC DS-CDMA when a QPSK scheme and half-rate turbo coding were employed. The number of antennas was $T_x = 1, 2, 4, 8$ and the number of turbo decoding iterations was I =1 and 4, respectively.

approximately 12 dB, while the $Tx = 2$ scheme required a CSNR of 9 dB. Furthermore, the $Tx = 4$ scheme required a CSNR of 6 dB and finally the $Tx = 8$ scheme necessitated a CSNR of 5 dB.

Let us now characterise the achievable performance of the MC DS-CDMA system considered in conjunction with AQAM. The AQAM-aided STS MC-CDMA system was configured for maintaining a near-constant BER. More explicitly, in our experiments the adaptive thresholds were adjusted for maintaining the target BER of $10^{-3}, 10^{-4}$and 10^{-5}. The MPEG-4 video PSNR versus CSNR performance of the proposed transceiver is portrayed in Figures 6.13 to 6.15 when configuring the AQAM modem to operate at target BERs of 10^{-3}, 10^{-4} and 10^{-5}, respectively. This alloawed us to maintain a near-constant video quality, regardless of the channel SNR. Observe for example that as shown in Figure 6.13, at the BER of 10^{-3} the lowest PSNR was 36.23 dB, while the highest PSNR was 39.82 dB. Similarly, in Figure 6.14 the PSNR ranges from 38.09 dB to 39.82 dB, while in Figure 6.15 from 39.41 dB to 39.82 dB. Furthermore, as also seen in Figure 6.13, there is a marked PSNR difference amongst the systems using a different number of antennas at the BER of 10^{-3}. This performance discrepancy becomes less dominant in Figures 6.14 and 6.15, where the target BER was lower, namely $BER = 10^{-4}$ and $BER = 10^{-5}$, respectively. These effects were further analysed and as an example Figures 6.16 to 6.18 show the PDF of the PSNR degradation for $Tx = 8$ at $CSNR = 2dB$ and at $BER = 10^{-3}, 10^{-4}$ as well as 10^{-5}.

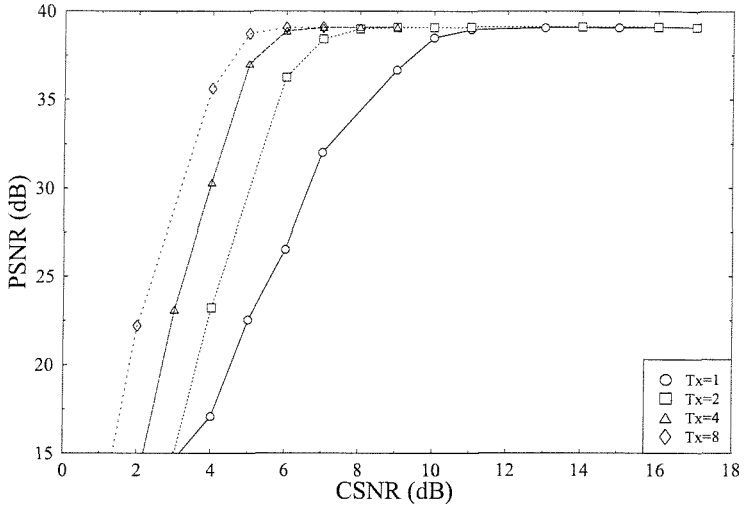


Figure 6.12: MPEG-4 video PSNR performance of STS assisted MC DS-CDMA when a QPSK scheme and half-rate turbo coding using four iterations were employed. The number of antennas was $T_x = 1, 2, 4, 8$.

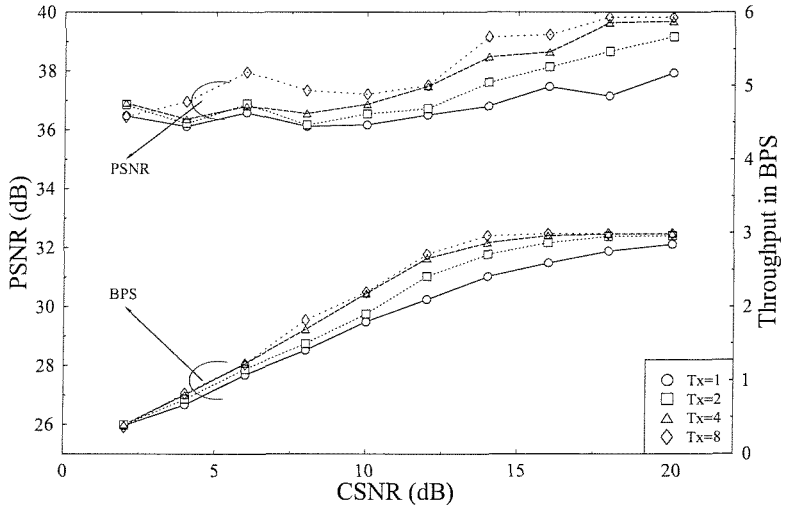


Figure 6.13: MPEG-4 video PSNR versus channel SNR performance of AQAM aided MC DS-CDMA. The switching thresholds were configured for maintaining a target **BER of 10^{-3}**.

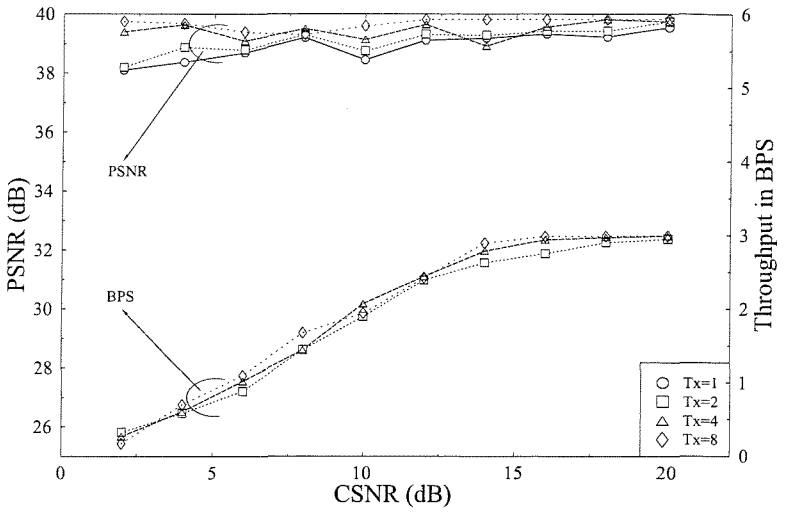


Figure 6.14: MPEG-4 video PSNR versus channel SNR performance of AQAM aided MC DS-CDMA. The switching thresholds were configured for maintaining a target **BER of 10^{-4}**.

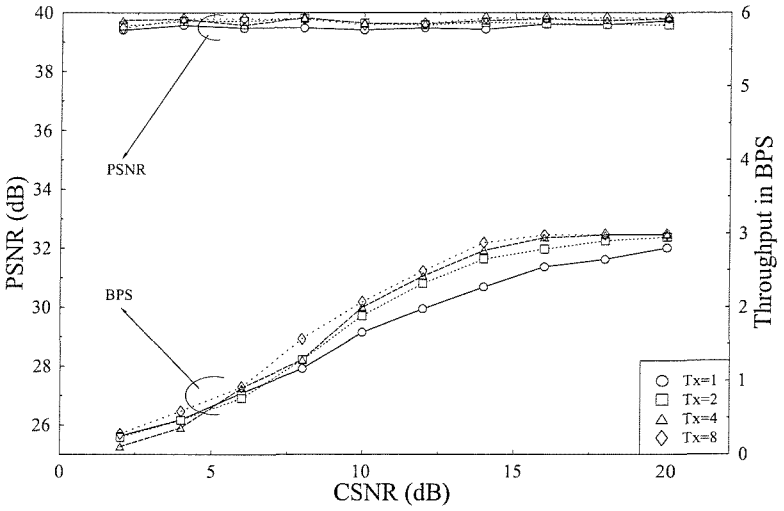


Figure 6.15: MPEG-4 video PSNR versus channel SNR performance of AQAM aided MC DS-CDMA. The switching thresholds were configured for maintaining a target **BER of 10^{-5}**.

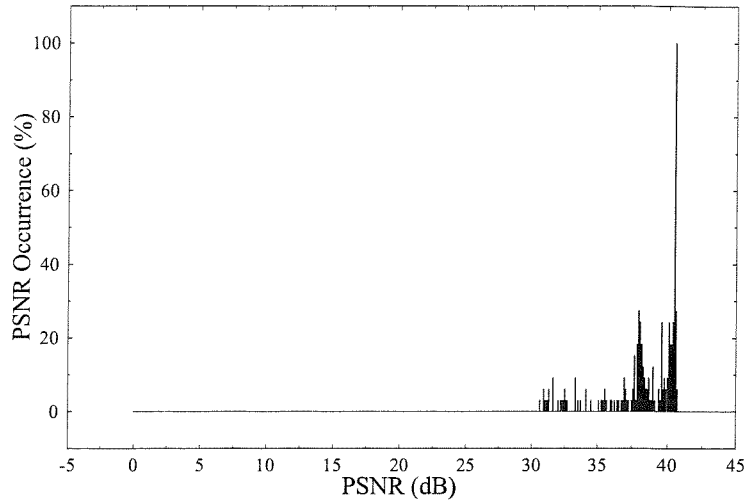


Figure 6.16: PDF of the MPEG-4 video PSNR of AQAM aided MC DS-CDMA. The switching thresholds were configured for maintaining a target **BER of 10^{-3}**. The number of antennas was $T_x = 8$ at CSNR = 2dB.

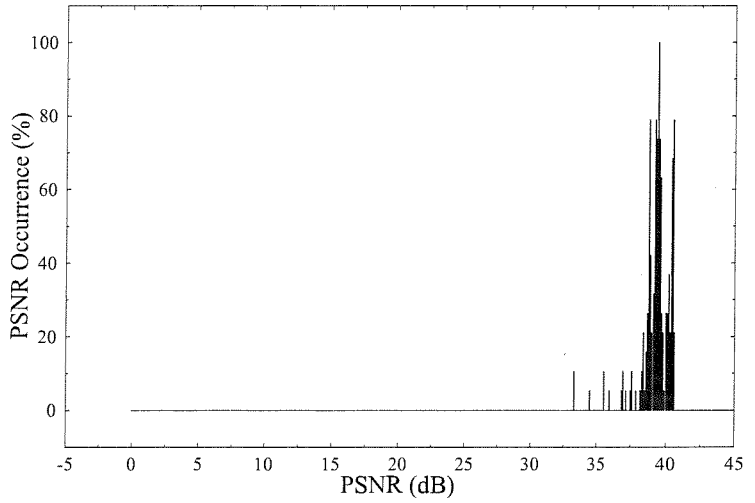


Figure 6.17: PDF of the MPEG-4 video PSNR of AQAM aided MC DS-CDMA. The switching thresholds were configured for maintaining a target **BER of 10^{-4}**. The number of antennas was $T_x = 8$ at CSNR = 2dB.

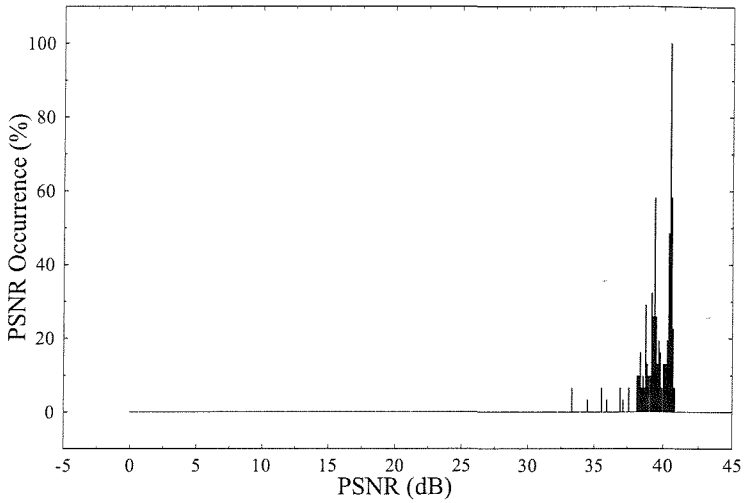


Figure 6.18: PDF of the MPEG-4 video PSNR of AQAM aided MC DS-CDMA. The switching thresholds were configured for maintaining a target **BER of 10^{-5}**. The number of antennas was $T_x = 8$ at CSNR = 2dB.

The associated BPS throughput performance curves were also plotted in Figures 6.13 to 6.15. We can observe from these figures that the throughput of our AQAM system increased gradually, as the CSNR increased. As the channel quality improved, the 64QAM mode was employed more often, reducing the probability of using BPSK, and as a result, the achievable throughput increased.

In conclusion, the AQAM-aided STS MC-CDMA system was capable of maintaining a near-constant target BER and hence a near-constant video PSNR across a wide range of channel qualities. This is particularly so in conjunction with $T_x = 4$ or 8 antennas and a target BER of 10^{-4} or 10^{-5}.

6.3.5.1 Subjective Testing

In order to complement the objective video performance evaluations provided in the previous section, subjective video assessments were also conducted. Since the objective PSNR video performance curves of Figures 6.13 to 6.15 were close to each other, their performance order was established by subjective testing. The aim of these investigations was to assess the performance of the AQAM scheme as a function of the number of STS transmitter antennas used. The video quality was assessed using pairwise comparison tests, where the panel viewers were asked to express a preference between two video sequences namely 'A' or 'B' or 'neither'. A total of 12 viewers took part in these subjective tests. Tables 6.3 to 6.6 detail some of the results of the viewing tests.

CSNR	Video Material A	Video Material B	A%	B%	Neither %
8dB	$10^{-3}(T_x = 1)$	$10^{-3}(T_x = 2)$	8.3%	50%	41.67%
	$10^{-3}(T_x = 2)$	$10^{-3}(T_x = 4)$	25%	58.3%	16.67%
	$10^{-3}(T_x = 4)$	$10^{-3}(T_x = 8)$	8.3%	66.67%	16.67%
14dB	$10^{-3}(T_x = 1)$	$10^{-3}(T_x = 2)$	41.67%	58.3%	0%
	$10^{-3}(T_x = 2)$	$10^{-3}(T_x = 4)$	8.3%	83.33%	8.3%
	$10^{-3}(T_x = 4)$	$10^{-3}(T_x = 8)$	0%	66.67%	33.33%

Table 6.3: Results of the subjective viewing tests conducted using pairwise comparisons, where the viewers were given a choice of preference between two video sequence coded in different transmission scenarios, both having $\mathbf{BER = 10^{-3}}$.

CSNR	Video Material A	Video Material B	A%	B%	Neither %
8dB	$10^{-4}(T_x = 1)$	$10^{-4}(T_x = 2)$	0%	58.33%	41.67%
	$10^{-4}(T_x = 2)$	$10^{-4}(T_x = 4)$	0%	16.67%	83.33%
	$10^{-4}(T_x = 4)$	$10^{-4}(T_x = 8)$	0%	16.67%	83.33%
14dB	$10^{-4}(T_x = 1)$	$10^{-4}(T_x = 2)$	50%	50%	0%
	$10^{-4}(T_x = 2)$	$10^{-4}(T_x = 4)$	58.33%	41.67%	0%
	$10^{-4}(T_x = 4)$	$10^{-4}(T_x = 8)$	8.3%	58.33%	33.33%

Table 6.4: Results of the subjective viewing tests conducted using pairwise comparisons, where the viewers were given a choice of preference between two video sequence coded in different transmission scenarios, both having $\mathbf{BER = 10^{-4}}$.

CSNR	Video Material A	Video Material B	A%	B%	Neither %
8dB	$10^{-5}(T_x = 1)$	$10^{-5}(T_x = 2)$	0%	0%	100%
	$10^{-5}(T_x = 2)$	$10^{-5}(T_x = 4)$	0%	0%	100%
	$10^{-5}(T_x = 4)$	$10^{-5}(T_x = 8)$	0%	0%	100%
14dB	$10^{-5}(T_x = 1)$	$10^{-5}(T_x = 2)$	0%	16.67%	83.33%
	$10^{-5}(T_x = 2)$	$10^{-5}(T_x = 4)$	0%	0%	100%
	$10^{-5}(T_x = 4)$	$10^{-5}(T_x = 8)$	8.3%	8.33%	83.33%

Table 6.5: Results of the subjective viewing tests conducted using pairwise comparisons, where the viewers were given a choice of preference between two video sequence coded in different transmission scenarios, both having $\mathbf{BER = 10^{-5}}$.

CSNR	Video Material A	Video Material B	A%	B%	Neither %
8dB	$10^{-3}(T_x = 8)$	$10^{-4}(T_x = 8)$	0%	100%	0%
	$10^{-4}(T_x = 8)$	$10^{-5}(T_x = 8)$	0%	83.33%	16.67%
	$10^{-3}(T_x = 8)$	$10^{-5}(T_x = 8)$	0%	100%	0%

Table 6.6: Results of the subjective viewing tests conducted using pairwise comparisons, where the viewers were given a choice of preference between two video sequence coded in different transmission scenarios, both having $\mathbf{BER = 10^{-3}}$, $\mathbf{BER = 10^{-4}}$ and $\mathbf{BER = 10^{-5}}$

Based on the subjective viewing tests, we found that at $BER = 10^{-3}$ the video quality recorded for different number of transmit antennas T_x was readily distinguishable. As can be seen from Table 6.3, more than 50% of the viewers preferred the scenarios having a higher number of transmit antennas. These results correlated with the corresponding objective preference results. When the systems's target BER was reduced to 10^{-4} or 10^{-5}, the subjective video quality differences associated with a different number of antennas became less noticeable which manifested itself in terms of none of the scenarios finding higher viewer preference. Finally, Table 6.6 suggests that the viewers were typically capable of unambiguously spotting the less corrupted, subjectively more pleasing video sequences having a lower BER.

6.3.6 Conclusions on AQAM/STS-Aided MC-CDMA Transceiver Scheme.

In this study, we focused our attention on the performance of AQAM as well as STS-assisted MC DS-CDMA and studied its ability to support wireless video communications. The simulated performance of this system was characterised. A range of system performance results was presented based on both fixed-modulation schemes as well as AQAM schemes. The high-efficiency MPEG-4 video codec was employed for compressing the video signal. The proposed system ensured robust video communications using the AQAM-aided MC DS-CDMA system in a highly dispersive Rayleigh-fading environment across a wide range of channel SNR values, which was also confirmed by subjective testing.

6.4 Unequal Error Protection MPEG-4 Based Turbo Transceiver

3

[3]This section is based on J. Y. Chung, S. X. Ng and L. Hanzo: Turbo-Detected Unequal Protection MPEG-4 Wireless Video Telephony using Trellis Coded Modulation and Space-Time Trellis Coding, submitted to and it was based on collaborative research with the co-authors.

6.4.1 Motivation and Background

The MPEG-4 standard [15, 212] offers a standardised framework for a whole range of multimedia applications. Examples include tele-shopping, interactive TV, internet games, or mobile video communication. With reference to Subsection 2.2.3, MPEG-4 integrates different types of multimedia services by the introduction of a so-called 'object-based' approach for the description and coding of multimedia contents. The key functionalities of MPEG-4 include independent coding of objects in a video frame, the ability to interactively embed these video objects into a scene shown on the screen, the transmission of 3D scene descriptions, quality versus bitrate based temporal and spatial scalability and improved error resilience [226].

As the MPEG-4 standard targets a broader range of different applications and bitrates than the previously defined hybrid video coding standards such as MPEG-1 [20], MPEG-2 [21] or H.263 [18], it employs a higher variety of different algorithms and coding modes. In the MPEG-4 coding algorithm a scene consists of one or more audio-visual objects potentially generated from multiple sources.

The MPEG-4 algorithm employed for encoding natural video scenes is the classic block-based hybrid coding scheme [4], which is known from the well-established MPEG-1 [20], MPEG-2 [21] or H.263 [18] codecs. However, these codecs were further developed in order to allow the encoding of arbitrarily shaped video objects. For employment in error-prone environments, error resilient encoding features were introduced by several parts of the MPEG-4 standards, which were introduced in Section 4.5. This renders the MPEG-4 coding standard particularly suitable for wireless video telephony.

In this study the MPEG-4 video codec was incorporated in a sophisticated unequal-protection turbo transceiver using joint coding and modulation as inner coding, twin-class convolutional outer coding as well as space time coding based spatial diversity. Specifically, Trellis Coded Modulation (TCM) [74, 184] constitutes a bandwidth-efficient joint channel coding and modulation scheme, which was originally designed for transmission over Additive White Gaussian Noise (AWGN) channels. In an effort to mitigate the effects of Rayleigh fading channels and render them Gaussian-like for the sake of supporting the operation of the TCM scheme, Space-Time Trellis Coding (STTC) [227] employing multiple transmit and receive antennas was invoked for providing spatial diversity gain. Furthermore, maximal minimum distance Non-Systematic Convolutional codes (NSCs) [193, p. 331] having two different code-rates were used for providing unequal video protection.

Again, in this study, a novel unequal-protection joint source-coding, channel-coding, modulation and spatial diversity aided turbo-transceiver is proposed, which consists of a STTC, a TCM and two different-rate NSCs. This STTC-TCM-2NSC scheme was designed for MPEG-4 video telephony. We will demonstrate that significant iteration gains are attained with the aid of the proposed turbo transceiver. This section is structured as follows. In Subsection 6.4.3 we describe the proposed

system's architecture and highlight the interactions of its constituent elements. We elaborate further by characterising the achievable system performance in Subsection 6.4.5 and conclude with a range of system design guidelines in Subsection 6.4.6.

6.4.2 MPEG-4 Source Sensitivity and Error Protection

In Section 4.7, the MPEG-4 bistream's error sensitivity was quantified, in order to find the appropriate bitstream partitioning scheme suitable for the error-prone wireless channel. According to our findings related to the MPEG-4 video sensitivity investigations provided in Figures 4.11 to 4.16 of Section 4.7, we concluded that the PSNR degradations inflicted by the various MPEG-4 parameters followed the hierarchial order of VO, VOL, VOP, and MB. More specifically, the VO header, VOL header, VOP header or MB header exhibited the highest PSNR degradation, hence they constituted the most sensitive bits in the entire bitstream, despite the fact that these parameters exhausted only about 10 % of the total bitrate budget. More quantitatively, the proportion of header bits is illustrated in Figure 6.19.

This suggested invoking Unequal Error Protection (UEP) techniques for protecting the MPEG-4 parameters during transmission. In a low-complexity implementation having two protection classes may be deemed sufficient. The higher priority class (*Class One*) would contain all the important header information, synchronisation markers and some of the more important DC and low-frequency VLC DCT coefficients. By contrast, the low priority class (*Class Two*) would contain the remaining relatively less important, higher frequency VLC DCT coefficients.

The PSNR degradation inflicted by both the Class One and Class Two video bits as well as the average PSNR degradation of the unpartitioned (*Full Class*) video bitstream was evaluated for a range of BER values in Figure 6.20 using randomly distributed bit errors. These results showed that as expected, a lower BER was required by the Class One bits than by the Class Two bits. More explicitly, it can be seen that Class One, which consists of the more sensitive bits suffers a higher PSNR degradation. By contrast, Class Two is inherently more robust to errors. Note that in an unpartitioned Full Class scenario the associated PSNR degradation is between that of Class One and Class Two.

6.4.3 The Turbo Transceiver

The block diagram of the serially concatenated STTC-TCM-2NSC turbo scheme using a STTC, a TCM and two RSCs as its constituent codes is depicted in Figure 6.21. The MPEG-4 codec operated at R_f=30 frames per second using the (176 × 144)-pixel Quarter Common Intermediate Format Miss America video sequence, encoded at a near-constant bitrate of R_b=69 kbps. Hence, we have R_b/R_f =

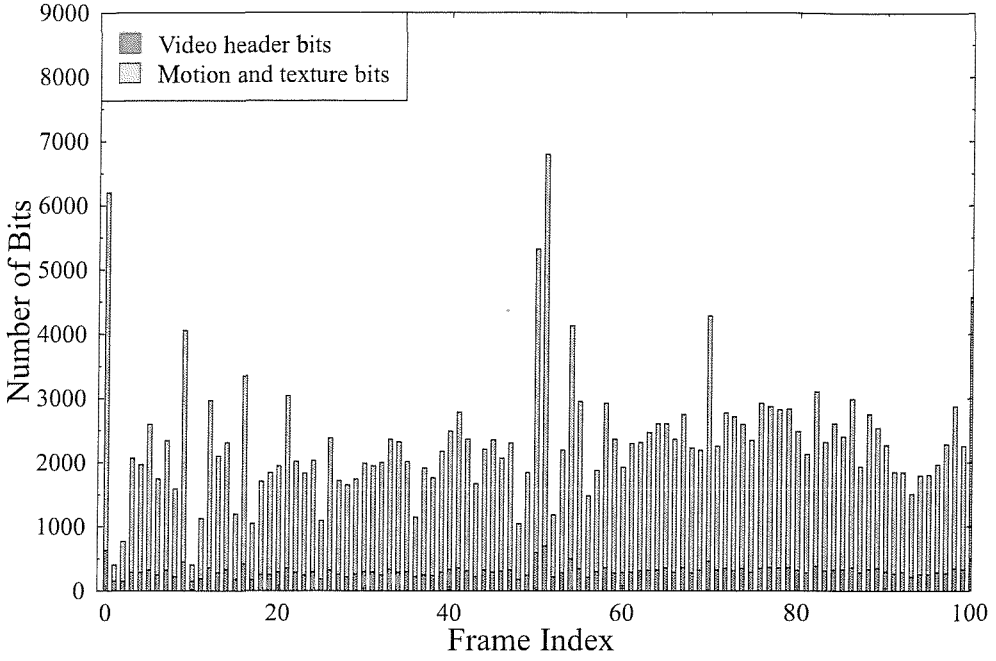


Figure 6.19: Probability of occurrence of the Class One and Class Two bits in the MPEG-4 codec in the context of the VOP shown on a frame by frame basis for the "Miss America" QCIF video sequence encoded at 30 frame/s and at a video bit rate of 69 kbps.

Parameter	**Value**
Channel type	Uncorrelated Rayleigh fading Channel
Transmission burst length	2300 bits
Modulation type	16QAM
Channel code	1/2-rate turbo code
Channel coded bitrate	150kbps
Interleaver length	960 bit
No. of turbo decoder iterations	4
Video codec	MPEG-4
Video frame-rate	30 frame/sec
Video format	176 × 144 QCIF
Video bitrate	69kbps

Table 6.7: System parameters

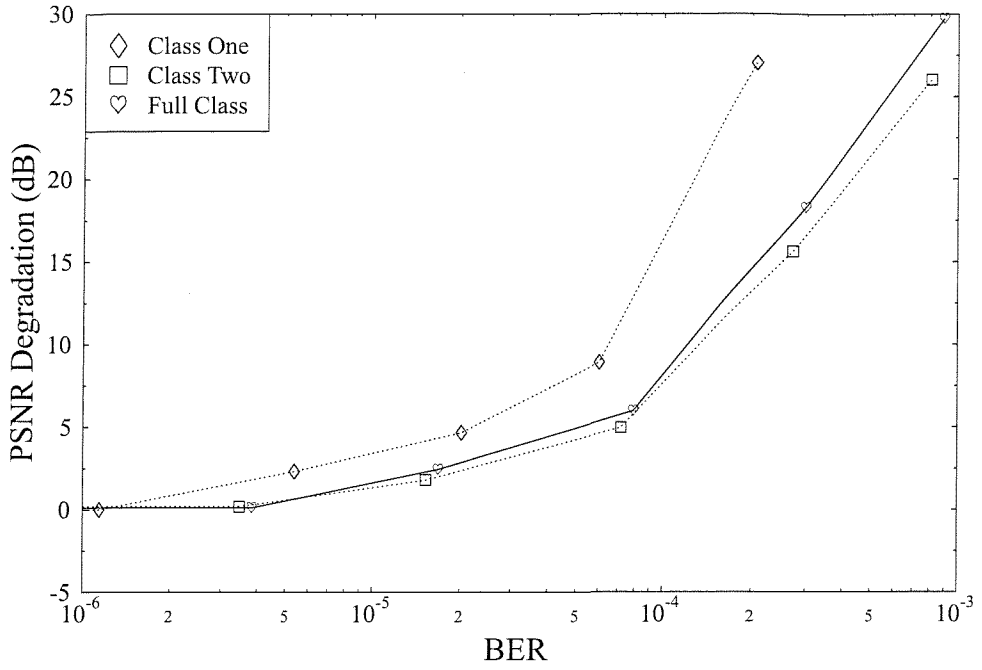


Figure 6.20: PSNR degradation versus average BER for the "Miss America" QCIF video sequence encoded at 30 frame/s and scanning at video frame rate of 69 kbps. When the specific class were corrupted, bits of the other classes were kept intact.

2300 bits per video frame. We partition the video bits into two unequal protection classes. Specifically, class-1 and class-2 consist of 25% (which is 575 bits) and 75% (which is 1725 bits) of the total number of video bits. The more sensitive video bits constituted mainly by the MPEG-4 framing and synchronisation bits are in class-1 and they are protected by a stronger binary NSC having a coding rate of $R_1 = k_1/n_1 = 1/2$ and a code memory of $L_1 = 3$. The less sensitive video bits predominantly signalling the MPEG-4 Discrete Cosine Transform (DCT) coefficients and motion vectors are in class-2 and they are protected by a weaker non-binary NSC having a coding rate of $R_2 = k_2/n_2 = 3/4$ and a code memory of $L_2 = 3$. Note that the number of MPEG-4 framing and synchronisation bits is only about 10% of the total video bits. Hence, about 25%-10%=15% of class-1 bits are constituted by the video bits signalling the most sensitive MPEG-4 DCT coefficients. We invoke code termination bits in both NSCs and hence the number of coded bits emerging from the $R_1 = 1/2$ binary NSC encoder is $(575 + k_1L_1)/R_1 = 1156$ bits, while that generated by the $R_2 = 3/4$ non-binary NSC encoder is $(1725 + k_2L_2)/R_1 = 2312$ bits. The class-1 and class-2 NSC coded bit sequences are interleaved by

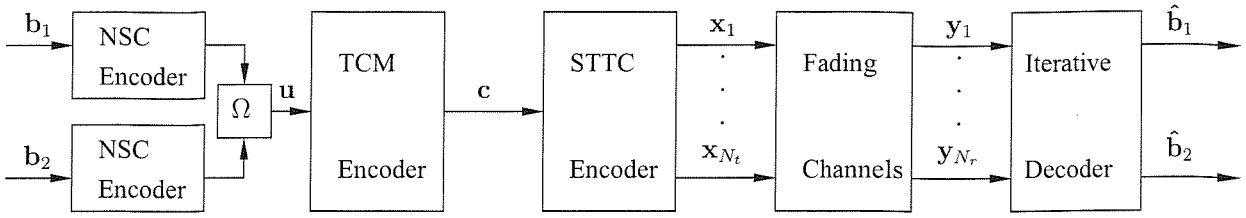


Figure 6.21: Block diagram of the serially concatenated STTC-TCM-2NSC scheme. The notations $\mathbf{b}_i$, $\hat{\mathbf{b}}_i$, $\mathbf{u}$, $\mathbf{c}$, $\mathbf{x}_j$ and $\mathbf{y}_k$ denote the vectors of the class-i video bits, the estimates of the class-i video bits, the resultant encoded bits of both NSC encoders, the TCM coded symbols, the STTC coded symbols for transmitter j and the received symbols at receiver k, respectively. Furthermore, Ω is a parallel-to-serial converter, while N_t and N_r denote the number of transmitters and receivers, respectively. The symbol-based channel interleaver between the STTC and TCM schemes as well as the two bit-based interleavers at the output of NSC encoders are not shown for simplicity. The iterative decoder seen at the right is detailed in Figure 6.22.

two separate bit interleavers of length 1156 and 2312 bits, respectively.

The two interleaved bit sequences are then concatenated in the parallel-to-serial converter block, denoted by Ω in Figure 6.21 to form a bit sequence of $1156 + 2312 = 3468$ bits. This bit sequence is then fed to the TCM encoder having a coding rate of $R_3 = k_3/n_3 = 3/4$ and a code memory of $L_4 = 3$. We employ code termination also in the TCM scheme and hence at the TCM encoder's output we have $(3468+k_3L_3)/R_3 = 4636$ bits or 4636/4=1159 TCM symbols. The TCM symbol sequence is then symbol-interleaved and fed to the STTC encoder. We invoke a 16-state STTC scheme having a code memory of $L_4 = 4$ and $N_t = 2$ transmit antennas, employing $M = 16$-level Quadrature Amplitude Modulation (16QAM). We terminate the STTC code by a 4-bit 16QAM symbol, since we have $N_t = 2$. Therefore, at each transmit antenna we have $1159 + 1 = 1160$ 16QAM symbols or $4 \times 1160 = 4640$ bits in a transmission frame. The overall coding rate is given by $R = 2300/4640 \approx 0.5$ and the effective throughput of the system is $\log_2(M)R \approx 2.0$ Bits Per Symbol (BPS).

At the receiver, we employ $N_r = 2$ receive antennas and the received signals are fed to the iterative decoders for the sake of estimating the video bit sequences in both class-1 and class-2, as seen in Figure 6.21. The STTC-TCM-2NSC scheme's turbo decoder structure is illustrated in Figure 6.22, where there are four constituent decoders, each labelled with a round-bracketed index. Symbol-based and bit-based MAP algorithms [74] operating in the logarithmic-domain are employed by the TCM as well as the rate $R_2 = 3/4$ NSC decoders and by the $R_1 = 1/2$ NSC decoder, respectively. The notations $P(.)$ and $L(.)$ in Figure 6.22 denote the logarithmic-domain symbol probabilities and the Logarithmic-Likelihood Ratio (LLR) of the bit probabilities, respectively. The notations c, u and b_i in the round brackets (.) in Figure 6.22 denote TCM coded symbols, TCM information symbols and

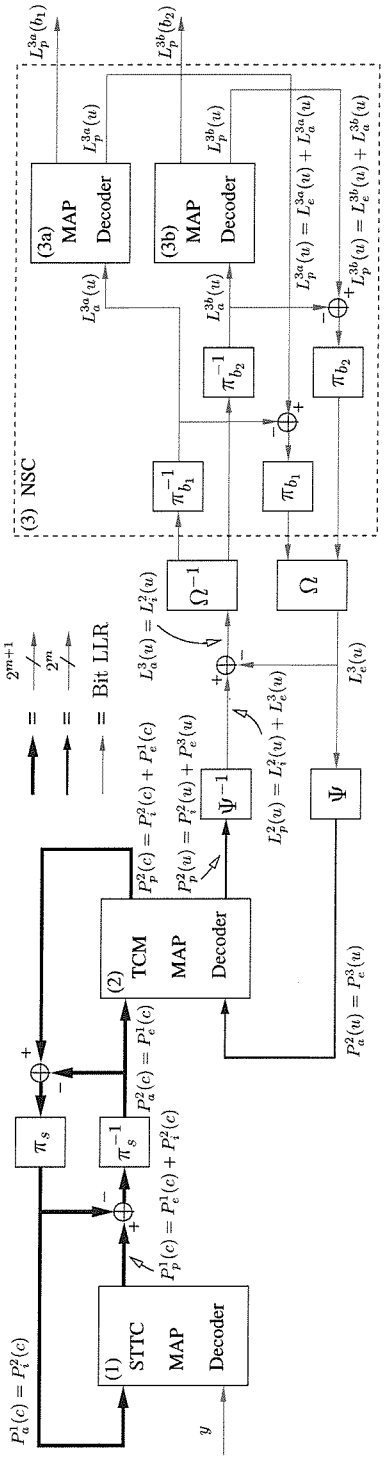


Figure 6.22: Block diagram of the STTC-TCM-2NSC turbo detection scheme seen at the right of Figure 6.21. The notations $\pi_{(s,b_i)}$ and $\pi^{-1}_{(s,b_i)}$ denote the interleaver and deinterleaver, while the subscript s denotes the symbol-based interlaver of TCM and the subscript b_i denotes the bit-based interleaver for class-i NSC. Furthermore, Ψ and Ψ^{-1} denote LLR-to-symbol probability and symbol probability-to-LLR conversion, while Ω and Ω^{-1} denote the parallel-to-serial and serial-to-parallel converter, respectively. The notation m denotes the number of information bits per TCM coded symbol. The thickness of the connecting lines indicates the number of non-binary symbol probabilities spanning from a single LLR per bit to 2^m and 2^{m+1} probabilities.

the class-i video bits, respectively. The specific nature of the probabilities and LLRs is represented by the subscripts a, p, e and i, which denote *a priori*, *a posteriori*, *extrinsic* and *intrinsic* information, respectively. The probabilities and LLRs associated with one of the four constituent decoders having a label of $\{1, 2, 3a, 3b\}$ are differentiated by the identical superscripts of $\{1, 2, 3a, 3b\}$. Note that the superscript 3 is used for representing the two NSC decoders of $3a$ and $3b$.

As we can observe from Figure 6.22, the STTC decoder of block 1 benefits from the *a priori* information provided by the TCM decoder of block 2, namely $P_a^1(c) = P_i^2(c)$ regarding the 2^{m+1}-ary TCM coded symbols, where m is the number of information bits per TCM coded symbol. More specifically, $P_i^2(c)$ is referred to as the intrinsic probability of the 2^{m+1}-ary TCM coded symbols, because it contains the inseparable extrinsic information provided by the TCM decoder itself as well as the *a priori* information regarding the uncoded 2^m-ary TCM input information symbols emerging from the NSC decoders of block 3, namely $P_a^2(u) = P_e^3(u)$. Hence, the STTC decoder indirectly also benefits from the *a priori* information $P_a^2(u) = P_e^3(u)$ provided by the NSC decoders of block 3, potentially enhanced by the TCM decoder of block 2. Similarly, the intrinsic probability of $P_i^2(u)$ provided by the TCM decoder for the sake of the NSC decoders' benefit consists of the inseparable extrinsic information generated by the TCM decoder itself as well as of the systematic information of the STTC decoder, namely $P_a^2(c) = P_e^1(c)$. Note that after the symbol probability-to-LLR conversion, $P_i^2(u)$ becomes $L_i^2(u)$. Therefore, the NSC decoders of block 3 benefit directly from the *a priori* information provided by the TCM decoder of block 2, namely from $L_a^3(u) = L_i^2(u)$ as well as indirectly from the *a priori* information provided by the STTC decoder of block 1, namely from $P_a^2(c) = P_e^1(c)$. On the other hand, the TCM decoder benefits directly from the STTC and NSC decoders through the *a priori* information of $P_a^2(c) = P_e^1(c)$ and $P_a^2(u) = P_e^3(u)$, respectively, as it is shown in Figure 6.22.

6.4.4 The Turbo Benchmarker

In order to evaluate the proposed scheme, we created a powerful benchmarker scheme by replacing the TCM and NSC encoders of Figure 6.21 by a single NSC codec having a coding rate of $R_0 = k_0/n_0 = 1/2$ and a code memory of $L_0 = 6$. We will refer to this benchmarker scheme as the STTC-NSC arrangement. All video bits are equally protected in the benchmarker scheme by a single NSC encoder and a STTC encoder. A bit-based channel interleaver is inserted between the NSC encoder and STTC encoder. Taking into account the bits required for code termination, the number of output bits of the NSC encoder is $(2300 + k_0 L_0)/R_0 = 4612$, which corresponds to 1153 16QAM symbols. Again, a 16-state STTC scheme having $N_t = 2$ transmit antennas is employed. After code termination, we have $1153 + 1 = 1154$ 16QAM symbols or $4(1154) = 4616$ bits in a transmission frame at each transmit antenna. The overall coding rate is given by $R = 2300/4616 \approx 0.5$ and the effective throughput is $\log_2(16)R \approx 2.0$ BPS, both of which are identical to the corresponding values of the

proposed scheme. The decoder structure of the STTC-NSC benchmarker scheme is similar to that of Figure 6.22, replacing the TCM and NSC decoders of block 2 and 3 by a single NSC decoder.

6.4.4.1 Systems Complexity

Let us define a single decoding iteration as a combination of a STTC decoding, a TCM decoding, a class-1 NSC decoding and a class-2 NSC decoding step for the proposed STTC-TCM-2NSC scheme. Similarly, a decoding iteration of the STTC-NSC benchmarker scheme is comprised of a STTC decoding and a NSC decoding step. We will quantify the decoding complexity of the proposed STTC-TCM-2NSC scheme and that of the benchmarker scheme using the number of decoding trellis states. The total number of decoding trellis states per iteration of the proposed scheme employing 2 NSC decoders having a code memory of $L_1 = L_2 = 3$, TCM having $L_3 = 3$ and STTC having $L_4 = 4$, is $S = 2^{L_1} + 2^{L_2} + 2^{L_3} + 2^{L_4} = 40$. By contrast, the total number of decoding trellis states per iteration for the benchmarker scheme having a code memory of $L_0 = 6$ and STTC having $L_4 = 4$, is given by $S = 2^{L_0} + 2^{L_4} = 80$. Therefore, the complexity of the proposed STTC-TCM-2NSC scheme having two iterations is equivalent to that of the benchmarker scheme having a single iteration, which corresponds to 80 decoding states. Similarly, the complexity of the STTC-TCM-2NSC scheme having four iterations is equivalent to that of the benchmarker scheme having two iterations, which corresponds to 160 decoding states.

6.4.5 Performance Results

In this section we evaluate the performance of the proposed MPEG-4 based video telephone schemes using both the Bit Error Ratio (BER) and the average video Peak Signal to Noise Ratio (PSNR) [4]. Figures 6.23 and 6.24 depict the BER versus Signal to Noise Ratio (SNR) per bit, namely E_b/N_0, performance of the proposed 16QAM-based STTC-TCM-2NSC scheme and that of the STTC-NSC benchmarker scheme, respectively, when communicating over uncorrelated Rayleigh fading channels. Furthermore, Figures 6.25 and 6.26 portray the PSNR versus E_b/N_0 performance of the proposed 16QAM-based STTC-TCM-2NSC scheme and that of the STTC-NSC benchmarker scheme, respectively, for transmissions over uncorrelated Rayleigh fading channels. Again, the overall throughput of both systems is 2 BPS.

It can be seen from Figure 6.23 that the class-1 video bits have a lower BER compared to the class-2 video bits. Specifically, at a BER of 10^{-4}, the class-1 bits can tolerate a 1.5 dB lower E_b/N_0 value at the first decoding iteration. Viewing this unequal-protection scheme from a different perspective, the class-1 bits benefit from more than an order of magnitude lower BER at a given SNR, than the class-2 bits. However, the E_b/N_0 difference of the two classes reduces to 0.5 dB at the fourth iteration. This is due to the different iteration gains attained by the two different bit classes. Explicitly, at

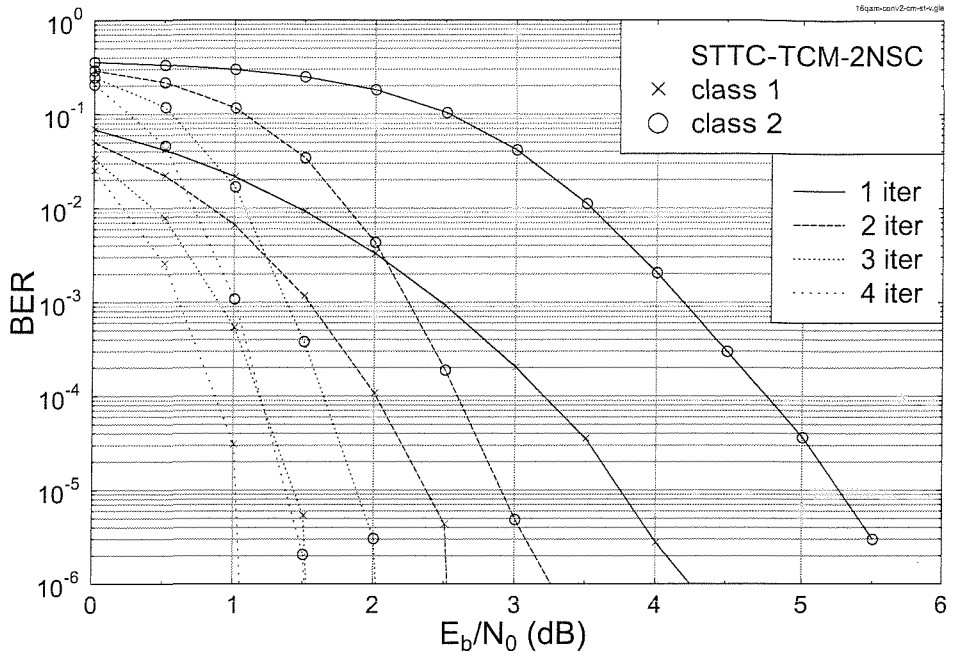


Figure 6.23: BER versus E_b/N_0 performance of the proposed 16QAM-based STTC-TCM-2NSC assisted MPEG-4 scheme, when communicating over uncorrelated Rayleigh fading channels. The effective throughput was **2 BPS**.

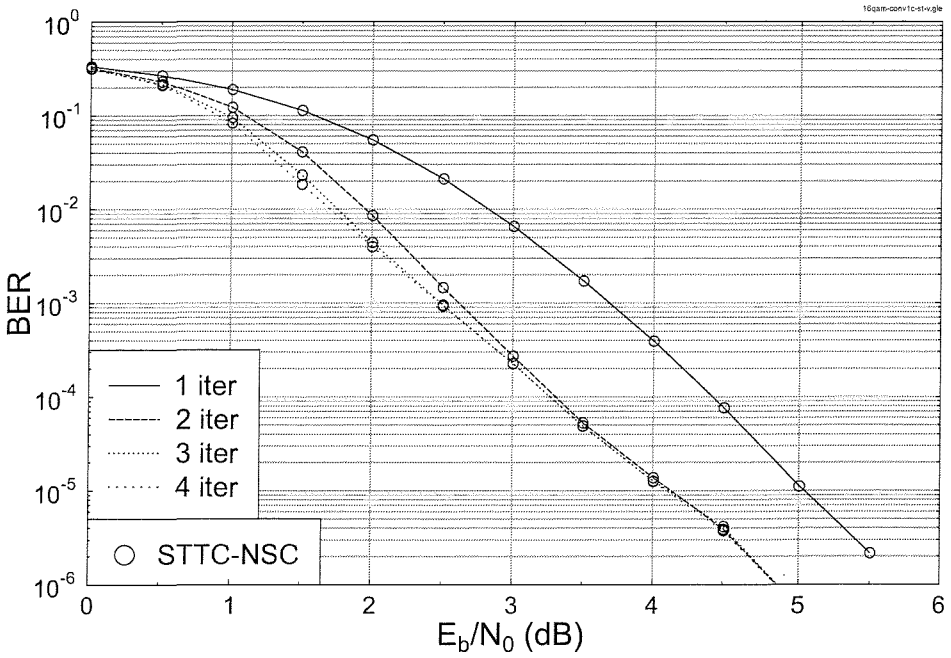


Figure 6.24: BER versus E_b/N_0 performance of the 16QAM-based STTC-NSC assisted MPEG-4 benchmarker scheme, when communicating over uncorrelated Rayleigh fading channels. The effective throughput was **2 BPS**.

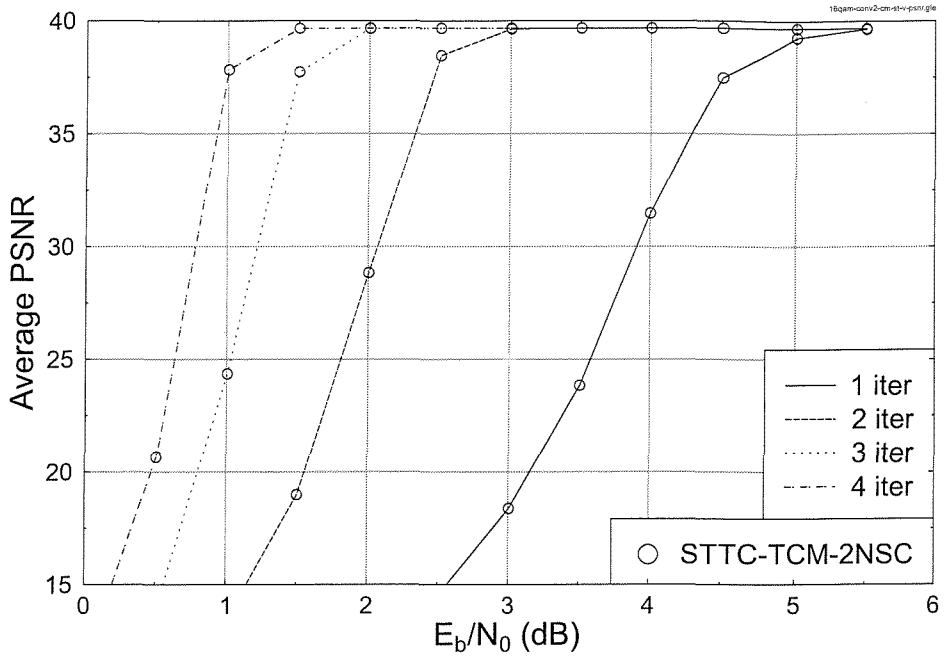


Figure 6.25: Average PSNR versus E_b/N_0 performance of the proposed 16QAM-based STTC-TCM-2NSC assisted MPEG-4 scheme, when communicating over uncorrelated Rayleigh fading channels. The effective throughput was **2 BPS**.

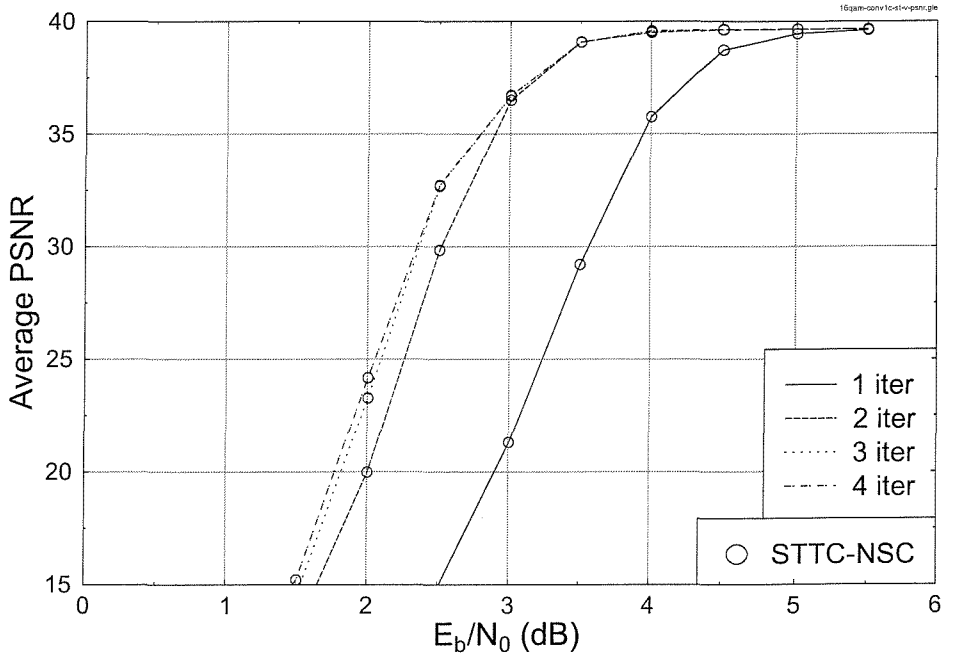


Figure 6.26: Average PSNR versus E_b/N_0 performance of the 16QAM-based STTC-NSC assisted MPEG-4 benchmarker scheme, when communicating over uncorrelated Rayleigh fading channels. The effective throughput was **2 BPS**.

a BER of 10^{-4} the iteration gains of the class-1 bits and class-2 bits are approximately 2.4 dB and 3.5 dB, respectively, when the number of iterations is increased from one to four. It is demonstrated in Figure 6.23 that the class-1 and class-2 bits required an extremely low E_b/N_0 of approximately 0.8 dB and 1.3 dB at BER=10^{-4} in order to attain a throughput of 2 BPS. By contrast, the STTC-NSC benchmarker scheme required an E_b/N_0 of approximately 4.4 dB and 3.3 dB at the first and second iteration, respectively. Further marginal gains of about 0.1 dB were attained, when the number of iterations was increased to three or four.

Let us now consider the PSNR versus E_b/N_0 performance of the systems characterised in Figures 6.25 and 6.26. It is evidenced in Figure 6.25 that similar to our observations made in the context of the achievable BER results, an approximately 3.5 dB of iteration gain was attained by the proposed STTC-TCM-2NSC scheme, when the number iterations was increased from one to four at a PSNR of 37.5 dB. Again, the complexity of STTC-TCM-2NSC having four iterations corresponds to 160 trellis states, which is similar to that of the STTC-NSC scheme having two iterations. As shown in Figure 6.26, at PSNR=37.5 dB the STTC-NSC benchmarker scheme having two iterations required an E_b/N_0 value of 3.2 dB, which is about 2.2 dB higher than that required by the similar-complexity STTC-TCM-2NSC arrangement having four iterations, as shown in Figure 6.25. Observe in Figure 6.26 that the STTC-NSC scheme only attains a significant iteration gain during the second iteration. Further increasing the number of iterations results in a higher decoding complexity, but attains no noteworthy iteration gain. Hence, the best possible attainable performance is PSNR=37.5 dB at E_b/N_0=3.2 dB. Note that if we reduce the code memory of the NSC constituent code of the STTC-NSC benchmarker scheme from L_0=6 to 3, the best possible performance becomes poorer. If we increased L_0 from 6 to 7 (or higher), the decoding complexity increased significantly, while the attainable best possible performance is only marginally increased. Hence, the STTC-NSC scheme having L_0=6 constitutes a good benchmarker scheme in terms of its performance versus complexity tradeoffs.

6.4.6 Conclusions on the STTC-TCM-2NSC Turbo Transceiver

In conclusion, a jointly optimised source-coding, outer channel-coding, inner coded modulation and spatial diversity aided turbo transceiver was proposed for MPEG-4 wireless video telephony. With the aid of two different-rate NSCs the video bits were protected differently according to their sensitivity. The employment of TCM improved the bandwidth efficiency of the system and by utilising STTC spatial diversity was attained. The performance of the proposed STTC-TCM-2NSC scheme was enhanced with the advent of an efficient iterative decoding structure. It was shown in Figure 6.25 that the proposed STTC-TCM-2NSC scheme required E_b/N_0=1 dB in order to attain a PSNR of 37.5 dB. For the sake of explicit comparison with the PIC-aided and AQAM/STS MC DS-CDMA transceivers of Sections 6.2 and 6.3, the E_b/N_0 values have to be converted to SNR values. Since we are using

2BPS 16QAM/TCM transmissions over the channel, this corresponds to a 3 dB shift of the SNR values, requiring an SNR of about 4.5 dB. At the cost of a similar complexity, the STTC-TCM-2NSC scheme is approximately 2.2 dB more efficient in terms of the required E_b/N_0 than the STTC-NSC benchmarker scheme.

6.5 Chapter Conclusions

In this chapter, we have proposed three different video transceivers for providing video telephony services. Specifically, an iterative parallel interference cancellation aided CDMA video telephony scheme is studied in Section 6.2, while a more sophisticated space-time spreading assisted multi-carrier DS-CDMA arrangement is the subject of Section 6.3. Finally, turbo-detected unequal error protection MPEG-4 video telephony using a serially concatenated convolutional outer code, trellis coded modulation based inner code and space-time coding in Section 6.4. Corresponding system components are summarised in Table 6.8.

System	Source Codec	Channel Codec	Modulator (Transceiver)
A	MPEG-4	Turbo Code, $R = 1/2, L = 3$	CDMA QPSK CIR: Two-path, equal-weight, sample-spaced Effective throughput 1 BPS
B	MPEG-4	Turbo Code $R = 1/2, L = 3$	MC DS-CDMA Space Time Spreading AQAM Channel: Uncorrelated Rayleigh fading channel for all subcarriers. Effective throughput 0.5 to 3 BPS
C	MPEG-4	TCM, $r = 3/4, L = 3$ NSC1, $r = 1/2, L = 3$ NSC2, $r = 3/4, L = 3$ $R \approx 1/2$	Space Time Trellis Coding 16QAM Channel: Uncorrelated Rayleigh fading channel Effective throughput 2 BPS

Table 6.8: General overview of the $R = 1/2, L = 3$ systems used in Sections 6.2, 6.3 and 6.4. The remaining parameters of these systems were summarised in Tables 6.1, 6.2 and 6.7

In *System A*, we conducted research into PIC-assisted multi-user CDMA based MPEG-4 videophony transceivers. The number of iterations used by the PIC-based turbo receiver was four. The family of m-sequence based spreading codes having a length of $N = 15$ chips was employed in the system and the transmission burst length was $T = 120$ bits. A random channel interleaver having a memory

of eight transmission bursts was employed. The system considered supported $K = 7$ users and communicated over an asynchronous symbol-spaced, equal-weight two-path uncorrelated Rayleigh fading channel. Our simulation results showed that in this multi-user CDMA system, for the QCIF resolution "Miss America" sequence, a channel SNR value in excess of 20 dB was required by *System A* to achieve a near-error-free video quality. Hence in Section 6.3, we investigated a more sophisticated video transceiver by proposing an AQAM-aided STS MC-DSCDMA based MPEG-4 video transceiver. We have shown that the adaptive scheme maintains a near-constant video quality across a wide range of channel SNRs. Finally, *System C* was investigated in Section 6.4, where a jointly optimised turbo transceiver capable of providing unequal error protection was proposed for MPEG-4 aided wireless video telephony. The transceiver advocated amalgamated STTC, TCM and two different-rate NSCs. A benchmarker scheme combining STTC and NSC was used for comparison with the proposed scheme. The video performance of the both schemes was evaluated, when communicating over uncorrelated Rayleigh fading channels. It was found that the proposed scheme required about two dBs lower transmit power than the equal-protection benchmarker scheme in the context of the MPEG-4 videophone transceiver, when aiming for an effective throughput of 2 bits/symbol at a similar decoding complexity.

Finally, the PSNR degradation versus channel SNR performance of the systems studied in this chapter are summarised in Table 6.9 and plotted in Figure 6.27. In conclusion, the channel SNRs required by System A, B and C respectively, for the sake of achieving an error-free video performance associated with a 0dB PSNR degradation were 20, 7 and 4.5 dB, which indicated the superiority of System C.

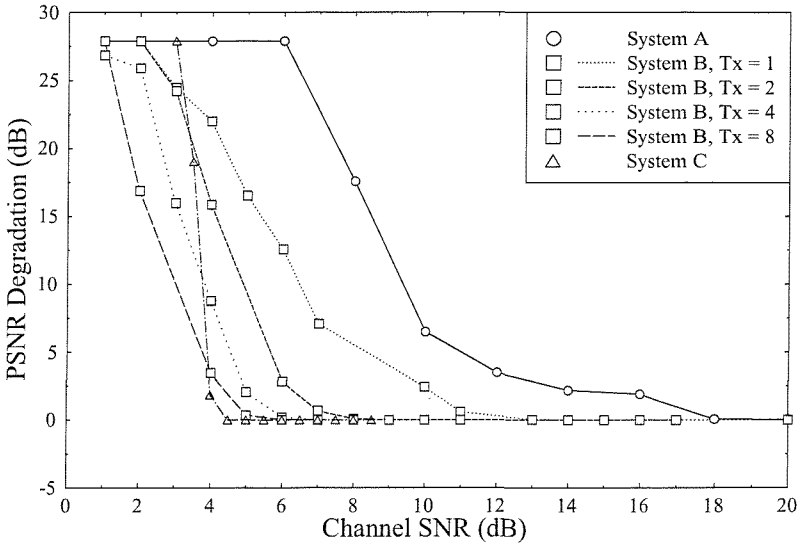


Figure 6.27: PSNR degradation of the various systems. The above results were extracted from Figures 6.4, 6.11 and 6.25. All the results shown above were derived using the QCIF resolution "Miss America" video sequence at the video encoding rate of 69 kbps.

CSNR (dB)	PSNR Degradation (PSNR)						
	System A	*System B*				*System C*	
		Tx=1	Tx=2	Tx=4	Tx=8	without TCM	with TCM
1.0	27.90	27.90	27.90	27.90	27.90	27.90	27.90
2.0	27.90	27.90	27.90	25.90	16.87	27.90	25.69
3.0	27.90	24.46	24.23	15.96	12.08	27.90	17.50
4.0	27.90	22.00	15.86	8.76	3.46	27.90	1.58
4.5	27.90	19.26	12.47	5.41	1.55	24.39	**0**
5.0	27.90	16.53	9.08	2.05	0.35	14.46	0
6.0	27.90	12.56	2.83	0.20	0.01	2.74	0
7.0	23.08	7.08	0.67	0.01	**0**	**0**	0
8.0	17.57	5.08	0.09	**0**	0	0	0
9.0	12.08	2.42	**0**	0	0	0	0
10.0	6.49	0.59	0	0	0	0	0
12.0	3.49	0.02	0	0	0	0	0
14.0	2.17	**0**	0	0	0	0	0
16.0	1.89	0	0	0	0	0	0
18.0	0.05	0	0	0	0	0	0
20.0	0.03	0	0	0	0	0	0

Table 6.9: PSNR degradation of the various systems. The above results were extracted from Figures 6.4, 6.11 and 6.25. All the results shown above were derived using the QCIF resolution "Miss America" video sequence at the video encoding rate of 69 kbps.

Chapter 7

Summary and Future Research Directions

7.1 Summary

In this thesis, a range of digital video compression standards have been discussed and a number of issues related to video transmissions over wireless fading channels have been investigated. Specifically, two video standards, namely the ISO MPEG-4 visual standard and the VCEG H.264 standard, were evaluated in the context of various transceivers.

This thesis commenced with a brief history and an overview of the current status of video compression techniques. The radio spectrum is a scarce resource. In the light of the enormous amount of video information to be transmitted, the employment of efficient coding techniques is of salient importance. For example, the transmission of a CIF video sequence at 15 frames/s over a 9.6kbits/s GSM channel would require a compression ratio of about 1900, which is unrealistic at the current state-of-art. Although some of the state-of-the-art video coding techniques, such as the ITU-T H.261, H.263 [22] and the well-established ISO MPEG-1 [142], MPEG-2 [143] standards are capable of providing reasonable compression ratios, there is a need for even higher coding efficiency in order to support the transmission of higher-definition picture formats, higher video frame rates and better visual quality video at low bit rates.

In Chapter 2, some of the functionalities of the ISO MPEG-4 video coding standard were introduced. The MPEG-4 visual standard consists of a 'core' video encoder/decoder model that invoke a number of additional coding tools. The core model is based on the well-established hybrid DPCM/DCT coding algorithm and the basic function of the core is extended by encoding 'tools' supporting enhanced-compression and reliable transmission. The MPEG-4 standard constitutes more

than just another compression scheme, since it evolved into a new concept of multi-media content manipulation, invoking powerful tools, such as *content-based interactivity*, which were discussed in Section 2.3. Furthermore, *shape coding* was discussed in Section 2.3.3, *scalability of video objects* in Section 2.4 and so on. These tools are capable of supporting various interactive functionalities and a vast range of other applications. The most significant departure of the MPEG-4 coding standard from the conventional video coding paradigms is that the concept of "objects" was introduced. Different parts of the final scene can be encoded and transmitted separately as one or more 'video objects', which may be re-composed by the decoder.

As described in Chapter 3, we have also studied the recently ratified H.264 video standard. The JVT's H.264 standard specifies another efficient video encoding scheme that may be applied in future mobile networking and communication systems. The JVT's objective was to create a single video coding standard that would simultaneously result in a new component in the MPEG-4 family of standards and in a new ITU-T H.264 Recommendation.The advanced functions of H.264, such as the 4×4-pixel block-based intra-frame prediction and the inter-frame prediction operations of Sections 3.4.1 to 3.4.2, respectively, have significantly increased the coding efficiency and the video quality. As it can be seen from Table 3.3, the employment of motion compensation in the context of the smallest affordable block size of 4×4-pixel sub block achieved a maximum of 0.44 dB PSNR gain compared to using the larger 16×16-pixel motion compensation block. We have also demonstrated in Subsection 3.4.3 that the integer DCT applied in conjunction with the scaling and rescaling quantisation processes yielded a high-quality reconstructed video block.

In Section 3.5 comparative studies of the MPEG-4 and H.264 codecs were carried out. Three demonstrative video sequences were selected for these investigations, namely the simple 'head and shoulders' "Miss America", the slightly higher-activity "Suzi" sequence and the high-texture, high-motion "Foreman" video sequences. The related comparative study indicated that the H.264 codec has a superior performance in comparison to the MPEG-4 codec.

The mobile channel constitutes a hostile propagation environment, which is associated with high bit error rates imposed by a number of propagation mechanisms, such as for example multi path fading, shadowing and co-channel interference. In the case of video, the effects of channel errors propagate due to the fact that the video bitstream is highly compressed with the aid of predictive techniques, in order to meet the stringent bandwidth limitations encountered. In fact, the higher the compression rate, the more sensitive the bitstream is to errors, since in case of high compression rates each encoded bit has an influence on a 'larger area' of the original video frame. We characterised the bitstream structure, the probability of the various parameters' occurrence and the bitstream sensitivity of the MPEG-4 codec in Chapter 4, indicating that the various video encoding parameters and the corresponding information bits have a different impact on the attainable video quality, when they are corrupted.

When transmitting video over a mobile channel, compressed video may suffer severe quality degradations and the employment of error-resilient techniques is vital. The effects of channel errors on the video quality are also augmented by the employment of variable-length coding (VLC) techniques, which are likely to lead to temporal and spatial error propagation. This situation is further aggravated in mobile channels, where bit error rates as high as 10^{-3} are common due to Rayleigh fading. We have characterised the sensitivity of the MPEG-4 bitstream parameters to channel errors. Our experiments have shown that the variable-length codes, which are used for encoding the DCT coefficients are the most susceptible to channel errors. This prompted us to use unequal error protection schemes for protecting the MPEG-4 bistream.

In Chapter 5, various burst-by-burst adaptive coded modulation aided joint detection based CDMA video transceivers have been studied, when communicating over the UTRA wideband vehicular fading channel. A burst-by-burst adaptive CDMA modem maximises the system's throughput by using the most appropriate modulation mode in response to the instantaneous channel and/or interference conditions. In this study, we have quantified the achievable video performance gains as a benefit of employing the proposed burst-by-burst adaptive CDMA modem.

The burst-by-burst adaptive modem guaranteed the same video performance, as the lowest- and highest-order fixed-mode modulation schemes at a range of low and high channel SNRs, respectively. However, between these extreme SNRs the effective video bitrate gracefully increased, as the channel SNR increased and/or the multi-user interference decreased, whilst maintaining a near-constant PLR. By controlling the AQAM switching thresholds a near-constant PLR can be maintained. The results portrayed in Figure 5.10 showed that the H.264 videophone system outperformed the corresponding H.263 codec based arrangement in video PSNR terms. Furthermore, we have also highlighted the employment of space-time coding in conjunction with an OFDM system in Section 5.4. Space-time coding in conjunction with a two-transmitter one-receiver configuration characterised in the context of fixed modulation based OFDM schemes outperformed the conventional one-transmitted one-receiver arrangement by 2.5 dB in channel SNR terms, when requiring an error-free video quality.

In Chapter 6, we conducted research into multi-user CDMA based MPEG-4 videophony transceivers. Explicitly, in Section 6.2 we proposed an iterative PIC aided CDMA MPEG-4 video phone scheme. The number of iterations used by the PIC-based turbo receiver was four. The family of m-sequence based spreading codes having a length of $N = 15$ chips was employed in our system and the transmission burst length was $T = 120$ bits. A random channel interleaver having a memory of eight transmission bursts was employed. The system considered supported $K = 7$ users and communicated over an asynchronous symbol-spaced, equal-weight two-path having a Doppler frequency of 100Hz Rayleigh fading channel. The simulation results of Figure 6.4 showed that this multiuser CDMA system provides an adequate video performance for channel SNRs in excess of about 20 dB. Hence in Section 6.3, we endeavoured to further enhance the transceiver's performance by proposing an AQAM

STS MC-DSCDMA-based MPEG-4 video transceiver. We have shown in Figures 6.13 to 6.15 that the adaptive scheme provides a fairly smooth PSNR evolution across the range of channel SNRs, corresponding to an attractive video performance. The improved performance of this adaptive system has also been confirmed by subjective viewing tests, as evidenced by Tables 6.3 to reftable:sub:comp

Based on our MPEG-4 bit sensitivity studies in Figures 4.11 to 4.16 of Section 4.7 we found that some of the encoded video parameters, such as the video header bits, are extremely sensitive to transmission errors. As a remedy of this problem, in Section 6.4.2 we have proposed a simple packetisation scheme, where we partitioned the MPEG-4 bitstream into two bit-sensitivity classes, namely a high-priority class and a low-priority class. Continuing our efforts in enhancing the transceiver' achievable performance further, we embarked on the study of an UEP scheme in conjunction with a novel turbo-detected trellis coded modulation and space-time coding scheme in Section 6.4. In our simulations we observed that the above scheme was capable of operating over uncorrelated Rayleigh-fading channels at an SNR as low as 4.5 dB, while transmitting at an effective throughput of 2BPS.

7.2 Future Research

There is a range of further research problems in the area of join source and channel coding. In recent years iterative transceivers such as turbo codecs, equalisers and turbo multi user detectors have reached near-Shannonian performance limits. These transceivers substantially benefit from exploiting the so-called extrinsic information inherent in the source-coded multi-media signal, provided that the source signal exhibits some residual correlation or redundancy.

A particular manifestation of this work is, when an entirely non-binary system is designed, where the Variable-Length Coded (VLC) source information is decoded by a Viterbi decoder operating on the basis of VLC codewords, rather than bits [133, 228, 229]. This technique has numerous merits in the context of various wireless transceivers and video codecs. Initially two prototype systems are worth investigating.

In System 1 the H.264 and MPEG-4 codecs may be invoked and the standard-compliant bitstream may be partitioned into two streams namely the VLC codewords and all other bits. The VLC stream may be decoded as mentioned above, while the remaining bits may use a range of other advanced iterative detectors. In System 2 the H.264 and MPEG-4 bitstream would be re-encoded using VLCs and the above procedure would be repeated for the sake of comparatively studying the resultant systems.

Most of the techniques studied in this treatise were proposed for low-rate applications. However, they are also beneficial in high-quality, high-bitrate wireless video broadcast systems communicating over highly dispersive channels at bitrates between 1 and 10 Mbps using for example 4CIF resolution

video sequences and OFDM/MC-CDMA systems.

List of Symbols

General notation

- The superscript * is used to indicate complex conjugation. Therefore, a^* represents the complex conjugate of the variable a.
- The superscript ^{T} is used to indicate matrix transpose operation. Therefore, $\mathbf{a}^T$ represents the transpose of the matrix $\mathbf{a}$.
- The superscript ^{H} is used to indicate complex conjugate transpose operation. Therefore, $\mathbf{a}^H$ represents the complex conjugate transpose of the matrix $\mathbf{a}$.
- The notation $*$ denotes the convolutional process. Therefore, $a * b$ represents the convolution between variables a and b.
- The notation $\hat{x}$ represents the estimate of x.
- The notation $X(f)$ is the Fourier Transform of $x(t)$.

Special symbols

$\mathbf{A}$: The system matrix in a CDMA system or a ST system.

b_i: The coefficient taps of the backward filters of the Decision Feedback Equalizer.

$\boldsymbol{b}$: A symbol vector, or combination, representing one of the possible transmited symbol sequence.

d_{free}: The minimum free Euclidean distance.

c^W: Walsh spreading code

c^R: Random spreading code.

D: The diversity quantity.

D_I: The diversity quantity of I component.

D_Q: The diversity quantity of Q component.

$\bar{D}$: The signal set dimensionality.

c_k: The coded symbol of the encoder at instance k.

$\mathbf{c}_j^i$: The M-ary multilevel RBF network's centres at node-j and level-i.

C_i: The coefficient taps of the forward filters of the Decision Feedback Equalizer.

E_b: Bit energy.

E_s: Symbol energy.

$E[k]$: The expected value of k.

f_n: The adaptive modem mode switching thresholds at mode n.

f_{RBF}^i: The conditional PDF of the M-ary multilevel RBF equaliser, at level-i.

$G(D)$: The generator polynomials matrix for Convolutional codes.

$H^i(D)$: The coefficients of the generator polynomials for bit i in TCM codes.

h_i: The channel's impulse response.

I_n: The random TTCM symbol-interleaver size in terms of the number of bits for adaptive modem mode n.

K: The number of users.

$\bar{K}$: The Rician fading parameter.

L: The number of resolvable paths in wideband channels.

$\bar{L}$: The memory length of CIR, i.e. $L-1$.

L: The "length" of the shortest error event path or the Hamming distance of a code.

$L(j,m)$: The branch label for transition from state j when the imput symbol is m.

m: The decision feedback equaliser's feedforward order.

m_I: The branch metrics for I component.

m_Q: The branch metrics for Q component.

$\bar{\mathsf{m}}$: The number of information bits in a modulated symbol.

$\tilde{\mathsf{m}}$: The number of encoded information bits in a modulated symbol.

m: The number of bits in a modulated symbol.

M: The number of levels of a multi-level modulation scheme, PSK or QAM.

M: The number of possible values of a source symbol.

M_o: The number of modulation modes in an adaptive modulation scheme.

$\mathcal{M}$: The number of RBF centres or the number of independant basis functions of the RBF equaliser.

n: The decision feedback equaliser's feedback order.

$n_{s,i}$: The number of channel states or the number of hidden nodes, at level-i M-ary multilevel RBF equaliser.

$n_{s,f}$: The number of scalar centres or channel states, of the reduced complexity RBF equaliser.

N: The number of symbols produced in each transmission interval, or the number of symbols per JD block.

$n(t)$: AWGN added to the transmitted signal.

N_0: Single-sided power spectral density of white noise.

P: The population size of GA.

P_b: Bit error probability.

P_m: The probability of mutation in GA.

Q: The number of chips in each spreading sequence, spreading factor.

$\bar{\mathsf{Q}}$: The first spreading ratio of the DS-RR scheme.

$\tilde{\mathsf{Q}}$: The second spreading ratio of the DS-RR scheme.

R: Coding rate.

r'_I: The I component of the decoupled channel output, in the I/Q-TEQ scheme.

r'_Q: The Q component of the decoupled channel output, in the I/Q-TEQ scheme.

S: The number of coding states, which is equals to 2^ν.

S_n: The trellis state at time instant n.

T: The mating pool size of GA.

u_k: The input symbol to the encoder at instance k.

V_m: The mth modulation mode of the adaptive modulation scheme, which has a total of M_o different modulation modes.

w_j^i: The M-ary multilevel RBF equaliser's weight at node-j and level-i.

x_k: The transmitted symbol at instance k.

y_k: The received symbol at instance k.

Y: The number of generations of GA.

α: The forward variable of a MAP decoder.

β: The backward variable of a MAP decoder.

$\chi(i,b)$: The subset that contains all the phasors for which the position i of the phasor has the binary value b, $b \in \{0,1\}$.

γ: The branch transition metric of a MAP decoder.

γ_{dfe}: The SNR at the output of the DFE.

$\Gamma_o(k)$: The SINR of the user-k at the output of the JD-MMSE-DFE.

ν: The code memory.

$\Omega(\boldsymbol{b})$: The correlation metric of symbol vector $\boldsymbol{b}$.

σ_S^2: The complex signal's variance.

σ_N^2: The complex noise's variance.

π: Interleaver.

π^{-1}: Deinterleaver.

τ: The decision feedback equaliser's decision delay.

φ_i: The nonlinear activation function of multilevel RBF equaliser at level-i.

Bibliography

[1] R. M. Fano, "Transmission of Information," in *M.I.T. Press*, (Cambridge, Mass), 1949.

[2] J. B. Connell, "A Huffman-Shannon-Fano code," *Proceedings of the IEEE*, vol. 61, pp. 1046–1047, 1973.

[3] D. A. Huffman, "A method for the construction of minimum-redundancy codes," *Proceedings of IRE*, vol. 20, 9, pp. 1098–1101, September 1952.

[4] L. Hanzo, P.J. Cherriman and J. Street, *Wireless Video Communications: Second to Third Generation Systems and Beyond.* NJ, USA : IEEE Press., 2001.

[5] K.R. Rao and P. Yip, "Discrete Cosine Transform - Algorithms, Advantages, Applications," in *Academic Press*, (San Diego, CA), 1990.

[6] B. Ramamurthi and A. Gersho, "Classified vector quantization of images," *IEEE Transactions on communications*, vol. COM-34, pp. 1105–1115, November 1986.

[7] C. W. Rutledge, "Vector DPCM: Vector Predictive Coding of Color Images," *Proceedings of the IEEE Global Telecommunications Conference*, pp. 1158–1164, September 1986.

[8] H. R. W. M. Yuen, "A Survey of Hybrid MC/DPCM/DCT Video Coding Distortions," *Signal Processing*, vol. 70, pp. 247–278, July 1998.

[9] S.-T. Hsiang and J. W. Woods, "Invertible Three-Dimensional Analysis/Synthesis System for Video Coding with Half-Pixel-Accurate Motion Compensation," *Proceedomgs of SPIE 3653, Visual Communications and Image Processing'99*, January 1999.

[10] G.J.Sullivan, T.Wiegand and T. Stockhammer, "Draft H.26L Video Coding Standard for Mobile Applications," in *Proceedings of IEEE International Conference on Image Processing*, vol. 3, (Thessaloniki, Greece), pp. 573–576, October 2001.

[11] A. A. Alatan, L. Onural, M. Wollborn, R. Mech, E. Tuncel, and T. Sikora, "Image Sequence Analysis for Emerging Interactive Multimedia ServicesThe European COST211 Framework," *IEEE Communications Letters*, vol. 8, pp. 802–813, November 1998.

[12] CCITT/SG XV, "Codecs for videoconferencing using primary digital group transmission," in *Recommendation H.120, CCITT (currently ITU-T)*, (Geneva), 1989.

[13] G. K. Wallace, "The JPEG Still Picture Compression Standard," *Communications of the Association for Computing Machinery*, vol. 34, no. 4, pp. 30–44, 1991.

[14] ITU-T/SG16/Q15, "Video coding for low bitrate communication," in *ITU-T Recommendation H.263, Version 2 (H.263+), ITU-T*, (Geneva), 1998.

[15] ISO/IEC JTC1/SC29/WG11, "Information technology - Generic coding of audio-visual objects.," in *Part 2: Visual. Draft ISO/IEC 14496-2 (MPEG-4), version 1, ISO/IEC*, (Geneva), 1998.

[16] ITU-T/SG16/Q15, "Draft for "H.263++" annexes U, V, and W to recommendation H.263," in *Draft, ITU-T*, (Geneva), 2000.

[17] Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, "Joint Final Committee Draft (JFCD) of Joint Video Specification (ITU-T Rec. H.264 ISO/IEC 14496-10 AVC)," August 2002.

[18] ITU-T/SG15, "Video coding for low bitrate communication," in *ITU-T Recommendation H.263, Version 1, ITU-T*, (Geneva), 1996.

[19] CCITT H.261, "Video Codec for audiovisual services at px64 kbit/s," 1990.

[20] ISO/IEC JTC1/SC29/WG11, "Information technology - coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbits/s. ," in *Part 2: Video. Draft ISO/IEC 11172-2 (MPEG-1), ISO/IEC*, (Geneva), 1991.

[21] ISO/IEC JTC1/SC29/WG11, "Information technology - Generic coding of moving pictures and associated audio.," in *Part 2: Video. Draft ISO/IEC 13818-2 (MPEG-2) and ITU-T Recommendation H.262, ISO/IEC amd ITU-T*, (Geneva), 1994.

[22] ITU-T Experts Group on very low Bitrate Visual Telephony, "ITU-T Recommendation H.263:Video coding for low bitrate communication," December 1995.

[23] MPEG Video Group, "Report of ad-hoc group on the evaluation of tools for non-tested functionalities of video submissions to MPEG-4," *Munich meeting, document ISO/IEC/JTC1/SC29/WG11 N0679*, Jan. 1996.

[24] D. E. Pearson, "Developments in model-based video coding," *Proceedings of the IEEE*, vol. 83, pp. 892–906, 5–9 December 1995.

[25] B. Liu and A. Zaccarin, "New Fast Algorithms for the Estimation of Block Motion Vectors," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 3, pp. 148–157, April 1993.

[26] ITU-R Recommendation BT.601-5 (10/95), "Studio encoding parameters of digital television for standard 4:3 and wide-screen 16:9 aspect tatios," .

[27] AT&T, "History of AT&T: Milestone in AT&T History. Online document available at URL:http://www.att.com/history/milestones.html."

[28] D. Cohen, "Specifications for the Network Voice Protocol (NVP)," *Internet Engineering Task Force, RFC 741*, November 1977.

[29] R. Cole, "PVP - A Packet Video Protocol," *Internal Document, USC/ISI*, July 1981.

[30] E. M. Schooler, "A Distributed Architecture for Multimedia Conference Control," *ISI research report ISI/RR-91-289*, November 1991.

[31] ITU-T Recommendation H.320, "Narrowband ISDN Visual Telephone Systems and Terminal Equipment," 1995.

[32] V. JACOBSON, "DARTNET Planning and Review Meeting," December 1991.

[33] T. Dorcey, "CU-SeeMe Desktop VideoConferencing Software," *Connexions*, vol. 9, March 1995.

[34] Thierry Turletti, "H.261 software codec for videoconferencing over the Internet," in *Rapports de Recherche 1834, Institut National de Recherche en Informatique et en Automatique (INRIA)*, (Sophia-Antipolis, France), January 1993.

[35] T. Dorcey, "CU-SeeMe Desktop Videoconferencing Software," *CONNEXIONS*, pp. 42–45, March 1995.

[36] H. Schulzrinne, "RTP: The real-time transport protocol," in *MCNC 2nd Packet Video Workshop*, vol. 2, (Research Triangle Park, NC), December 1992.

[37] Cornell University, "The CU-SeeMe Home Page, URL: http://cu-seeme.cornell.edu/."

[38] G. Venditto, "Internet Phones - The Future is Calling," *Internet World*, pp. 40–52, 1996.

[39] Vocaltec Communications Ltd., "The Vocaltec Telephony Gateway. Online document available at URL:http://www.vocaltec.com/products/products.htm."

[40] ITU-T Recommendation H.324, "Terminal for Low Bitrate Multimedia Communication," 1995.

[41] ITU-T Recommendation T.120, "Data protocols for multimedia conferencing," July 1996.

[42] ITU-T Recommendation H.323, "Visual Telephone Systems and Equipment for LAN which Provide a Non-guaranteed Quality of Service," November 1996.

[43] CERN, "Caltech and HP open scientific datacenter," November 1997.

[44] R. Braden et al., "Resource Reservation Protocol (RSVP)," September 1997.

[45] ITU-T Recommendation H.323 Version 2, "Packet-based multimedia communication systems," January 1998.

[46] J. Lennox, J. Rosenberg, H. Schulzrinne, "Common Gateway Interface for SIP," June 2000.

[47] ITU-T Recommendation H.450.4, "Call hold supplementary service for H.323, Series H: Audiovisual and Multimedia Sytems," May 1999.

[48] ITU-T Recommendation H.323 Version 4, "Packet-based multimedia communication systems," November 2000.

[49] CCITT/SG XV, "Video codec for audiovisual services at $p \times 64$ kbit/s," in *Recommendation H.120, CCITT (currently ITU-T)*, (Geneva), 1993.

[50] ISO/IEC JTC1, "Coding of audio-visual objects - Part2: Visual," April 1999.

[51] ITU-T Recommendation H.263, Version 2., "Video coding for low bitrate communication. International Telecommunications Union, Geneva," , January 1998.

[52] M.R. Civanlar, A. Luthra, S. Wenger and W. Zhu, "Special issue on streaming video," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, March 2001.

[53] C.W. Chen, P. Cosman, N. Kingsbury, J. Liang and J.W. Modestino, "Special issue on error resilient image and video transmission," *IEEE Journal on Selected Area in Communications*, vol. 18, June 2001.

[54] P. List, A. Joch, J. Lainema, G. Bjontegaard and M. Karczewicz, "Adaptive Deblocking Filter," *IEEE Transaction on Circuits and Systems for Video Technology*, vol. 13, pp. 614–619, July 2003.

[55] Y. Wang and Q.-F. Zhu, "Error control and concealment for video communications: A review," *Proceedings of the IEEE*, vol. 86, pp. 974–997, May 1998.

[56] S. Aign and K. Fazel, "Temporal and spatial error concealment techniques for hierarchical MPEG-2 video codec," in *in Proceedings IEEE International Conference on Communications ICC 95*, (Seattle, WA), pp. 1778–1783, June 1995.

[57] Y. Wang, Q.-F. Zhu and L.Shaw, "Maximally smooth image recovery in transform coding," *IEEE Transactions on Communications*, vol. 41, pp. 1544–1551, October 1993.

[58] S.S. Hemami and T.H.-Y. Meng, "Transform coded image reconstruction exploiting interblock correlation," *IEEE Transactions on Image Processing*, vol. 4, pp. 1023–1027, July 1995.

[59] W.-M. Lam and A.R. Reibman, "An error concealment algorithm for images subject to channel errors," *IEEE Transactions on Image Processing*, vol. 4, pp. 533–542, May 1995.

[60] H. Sun, K. Challapali and J. Zdepski, "Error concealment in digital simulcast AD-HDTV decoder," *IEEE Transactions on Consumer Electronics*, vol. 38, pp. 108–118, August 1992.

[61] R. Aravind, M.R. Civanlar and A.R. Reibman, "Packet loss resilience of MPEG-2 scalable video coding algorithms," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, pp. 426–435, October 1996.

[62] W.-J. Chu and J.-J. Leou, "Detection and concealment of transmission errors in H.261 images," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 8, pp. 78–84, February 1998.

[63] J. Apostolopoulos, "Error-resilient video compression via multiple state streams," *IProceedings IEEE International Workshop on Very Low Bit Rate Video Coding*, pp. 168–171, October 1999.

[64] Q.-F. Zhu, Y. Wang and L. Shaw, "Coding and cell-loss recovery in DCTbased packet video," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 3, pp. 248–258, June 1993.

[65] B. Haskell and D. Messerschmitt, "Resynchronization of motion compensated video affected by ATM cell loss," *Proceedings IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 93*, pp. 545–548, March 1992.

[66] A. Narula and J.S. Lim, "Error concealment techniques for an all-digital highdefinition television system," *Proceedings of the SPIE*, vol. 2094, pp. 304–318, November 1993.

[67] W.M. Lam, A.R. Reibman and B. Liu, "Recovery of lost or erroneously received motion vectors," *Proceedings IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 93*, vol. 5, pp. 417–420, April 1993.

[68] J. Lu, M.L. Lieu, K.B. Letaief and J.C.-I. Chuang, "Error resilient transmission of H.263 coded video over mobile networks," *Proceedings IEEE International Symposium on Circuits and Systems*, vol. 4, pp. 502–505, June 1998.

[69] M. Ghanbari and V. Seferidis, "Cell-loss concealment in ATM video codecs," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 3, pp. 238–247, June 1993.

[70] L.H. Kieu and K.N. Ngan, "Cell-loss concealment techniques for layered video codecs in an ATM network," *IEEE Transactions on Image Processing*, vol. 3, pp. 666–677, September 1994.

[71] L. Hanzo, L.L. Yang, E.L. Kuan and K. Yen, *Single- and Multi-Carrier CDMA*. Chichester, UK: John Wiley-IEEE Press, 2003.

[72] L. Hanzo, W. Webb and T. Keller, *Single- and Multi-Carrier Quadrature Amplitude Modulation: Principles and Applications for Personal Communications, WLANs and Broadcasting.* Piscataway, NJ, USA: IEEE Press, 2000.

[73] L. Hanzo, C.H. Wong, and M.S. Yee, *Adaptive Wireless Tranceivers.* Chichester, UK: John Wiley-IEEE Press, 2002.

[74] L. Hanzo, T.H. Liew and B.L. Yeap, *Turbo Coding, Turbo Equalisation and Space Time Coding for Transmission over Wireless channels.* New York, USA: John Wiley-IEEE Press, 2002.

[75] P. Jung and J. Blanz, "Joint detection with coherent receiver antenna diversity in CDMA mobile radio systems," *IEEE Transactions on Vehicular Technology*, vol. 44, pp. 76–88, Feb. 1995.

[76] L. Hanzo, Müenster ,Choi and Keller, *OFDM and MC-CDMA*. Chichester, UK: John Wiley and IEEE Press., 2003.

[77] E. Steinbach, N. Färber and B. Girod, "Adaptive playout for low-latency video streaming," in *IEEE International Conference on Image Processsing ICIP-01*, (Thessaloniki, Greece), pp. 962–965, October 2001.

[78] M. Kalman, E. Steinbach and B. Girod, "Adaptive playout for real-time media streaming," in *IEEE International Symposium on Circuits and Systems*, vol. 1, (Scottsdale, AZ), pp. 45–48, May 2002.

[79] S. Wenger, G.D. Knorr, J. Ott and F. Kossentini, "Error resilience support in H.263+," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 8, pp. 867–877, November 1998.

[80] R. Talluri, "Error-resilient video coding in the ISO MPEG-4 standard," *IEEE Communications Magazine*, vol. 2, pp. 112–119, June 1998.

[81] N. Färber, B. Girod and J. Villasenor, "Extension of ITU-T Recommendation H.324 for error-resilient video transmission," *IEEE Communications Magazine*, vol. 2, pp. 120–128, June 1998.

[82] E. Steinbach, N. Färber and B. Girod, "Standard compatible extension of H.263 for robust video transmission in mobile environments," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 7, pp. 872–881, December 1997.

[83] B. Girod and N. Färber, "Feedback-based error control for mobile video transmission," *Proceedings of the IEEE*, vol. 87, pp. 1707–1723, October 1999.

[84] G.J. Sullivan and T. Wiegand, "Rate-distortion optimization for video compression," *IEEE Signal Processing Magazine*, vol. 15, pp. 74–90, November 1998.

[85] A. Ortega and K. Ramchandran, "From rate-distortion theory to commercial image and video compression technology," *IEEE Signal Processing Magazine*, vol. 15, pp. 20–22, November 1998.

[86] T. Wiegand, X. Zhang and B. Girod, "Long-term memory motion-compensated prediction," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, pp. 70–84, February 1999.

[87] T. Wiegand, N. Färber and B. Girod, "Error-resilient video transmission using long-term memory motion-compensated prediction," *IEEE Journal on Selected Areas in Communications*, vol. 18, pp. 1050–1062, June 2000.

[88] P.A. Chou, A.E. Mohr, A. Wang and S. Mehrotra, "Error control for receiver-driven layered multicast of audio and video," *IEEE Transactions on Multimedia*, vol. 3, pp. 108–122, March 2001.

[89] G. Cote and F. Kossentini, "Optimal intra coding of blocks for robust video communication over the Internet," *Signal Processing: Image Communication*, vol. 15, pp. 25–34, September 1999.

[90] R. Zhang, S.L. Regunathan and K. Rose, "Video coding with optimal inter/intra-mode switching for packet loss resilience," *IEEE Journal on Selected Areas in Communications*, vol. 18, pp. 966–976, June 2000.

[91] R. Zhang, S.L. Regunathan and K. Rose, "Optimal estimation for error concealment in scalable video coding," in *Proceedings of Thirty-Fourth Asilomar Conference on Signals, Systems and Computers*, vol. 2, (Pacific Grove, CA), pp. 1974–1978, 2000.

[92] R. Zhang, S.L. Regunathan and K. Rose, "Robust video coding for packet networks with feedback," in *Proceedings Thirty-Fourth Asilomar Conference on Signals, Systems and Computers*, (Snowbird, UT), pp. 450–459, 2000.

[93] W. Tan and A. Zakhor, "Video multicast using layered FEC and scalable compression," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, pp. 373–387, March 2001.

[94] P.C. Cosman, J.K. Rogers, P.G. Sherwood and K. Zeger, "Image transmission over channels with bit errors and packet erasures," in *Proceedings Thirty-Fourth Asilomar Conference on Signals, Systems and Computers*, (Pacific Grove, CA), pp. 1621–1625, 1998.

[95] S. Lee and P. Lee, "Cell loss and error recovery in variable rate video," *Journal of Visual Communication and Image Representation*, vol. 4, pp. 39–45, March 1993.

[96] B. Girod, K. Stuhlmüller, M. Link and U. Horn, "Packet loss resilient Internet video streaming," in *Proceedings of Visual Communications and Image Processing VCIP-99*, (San Jose, CA), pp. 833–844, January 1999.

[97] M. Khansari and M. Vetterli, "Layered transmission of signals over powerconstrained wireless channels," in *Proc. of the IEEE International Conference on Image Processing (ICIP)*, vol. 3, (Washington, DC), pp. 380–383, October 1995.

[98] R. Puri and K. Ramchandran, "Multiple description source coding using forward error correction codes," in *Proc. Asilomar Conference on Signals, Systems and Computers*, (Pacific Grove, CA,), pp. 342–346, November 1999.

[99] R. Puri, K. Ramchandran, K.W. Lee and V. Bharghavan, "Forward error correction (FEC) codes based multiple description coding for Internet video streaming and multicast," *Signal Processing: Image Communicationn*, vol. 16, pp. 745–762, May 2001.

[100] P.A. Chou and K. Ramchandran, "Clustering source/channel rate allocations for receiver-driven multicast under a limited number of streams," in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, vol. 3, (New York, NY), pp. 1221–1224, July 2000.

[101] K. Stuhlmüller, N. Farber, M. Link and B. Girod, "Analysis of video transmission over lossy channels," *IEEE Journal on Selected Areas in Communications*, vol. 18, pp. 1012–1032, June 2000.

[102] Y.J. Liang, J.G. Apostolopoulos and B. Girod, "Model-based delay-distortion optimization for video streaming using packet interleaving," in *Proceedings of the 36th Asilomar Conference on Signals, Systems and Computers*, (Pacific Grove, CA), November 2002.

[103] S. Wicker, *Error Control Systems for Digital Communication and Storage.* Prentice-Hall, 1995.

[104] B. Dempsey, J. Liebeherr and A. Weaver, "On retransmission-based error control for continuous media traffic in packet-switching networks," *Computer Networks and ISDN Systems Journal*, vol. 28, pp. 719–736, March 1996.

[105] H. Liu and M. E. Zarki, "Performance of H.263 video transmission over wireless channels using hybrid ARQ," *IEEE Journal on Selected Areas in Communications*, vol. 15, pp. 1775–1786, December 1999.

[106] C. Papadopoulos and G.M. Parulkar, "Retransmission-based error control for continuous media applications," in *Proc. Networkand Operating System Support for Digital Audio and Video (NOSSDAV)*, (Zushi, Japan), pp. 5–12, April 1996.

[107] G.J. Conklin, G.S. Greenbaum, K.O. Lillevold, A.F. Lippman and Y.A. Reznik, "Video coding for streaming media delivery on the Internet," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, pp. 269–281, March 2001.

[108] Y.-Q. Zhang, Y.-J. Liu and R.L. Pickholtz, "Layered image transmission over cellular radio channels," *IEEE Transactions on Vehicular Technology*, vol. 43, pp. 786–794, August 1994.

[109] B. Girod, N. Färber and U. Horn, "Scalable codec architectures for Internet video-on-demand," in *Proceedings of the Thirty-First Asilomar Conference on Signals, Systems and Computers*, (Pacific Grove, CA), pp. 357–361, November 1997.

[110] A. Puri, L. Yan and B.G. Haskell, "Temporal resolution scalable video coding," in *Proc. of the IEEE International Conference on Image Processing (ICIP)*, vol. 2, (Austin, TX), pp. 947–951, November 1994.

[111] K.M. Uz, M. Vetterli, and D.J. LeGall, "Interpolative multiresolution coding of advance television with compatible subchannels," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 1, pp. 88–99, March 1991.

[112] S. Zafar, Y.-Q. Zhang and B. Jabbari, "Multiscale video representation using multiresolution motion compensation and wavelet decomposition," *IEEE Journal on Selected Areas in Communications*, vol. 11, pp. 24–35, January 1993.

[113] U. Horn, K. Stuhlmüller, M. Link and B. Girod, "Robust Internet video transmission based on scalable coding and unequal error protection," *Signal Processing: Image Communication*, vol. 15, pp. 77–94, September 1999.

[114] B. Girod and U. Horn, "Scalable codec for Internet video streaming," in *Proceedings of the International Conference on Digital Signal Processing*, (Piscataway, NJ), pp. 221–224, July 1997.

[115] M. Khansari, A. Zakauddin, W.-Y. Chan, E. Dubois and P. Mermelstein, "Approaches to layered coding for dual-rate wireless video transmission," *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, pp. 285–262, November 1994.

[116] G. Karlsson and M. Vetterli, "Subband coding of video for packet networks," *Optical Engineering*, vol. 27, pp. 574–586, July 1998.

[117] D. Quaglia and J.C. De Martin, "Delivery of MPEG video streams with constant perceptual quality of service," *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, vol. 2, pp. 85–88, August 2002.

[118] E. Masala, D. Quaglia and J.C. De Martin, "Adaptive picture slicing for distortion-based classification of video packets," *Proceedings of the IEEE Fourth Workshop on Multimedia Signal Processing*, pp. 111–116, October 2001.

[119] J. Shin, J.W. Kim and C.C.J. Kuo, "Quality-of-service mapping mechanism for packet video in differentiated services network," *IEEE Transactions on Multimedia*, vol. 3, pp. 219–231, June 2001.

[120] J. Shin, J. Kim and C.-C.J. Kuo, "Relative priority based QoS interaction between video applications and differentiated service networks," in *Proceedings of the IEEE International Conference on Image Processing*, (Vancouver, BC, Canada), pp. 536–539, September 2000.

[121] S. Regunathan, R. Zhang and K. Rose, "Scalable video coding with robust mode selection," *Signal Processing: Image Communication*, vol. 16, pp. 725–732, May 2001.

[122] H. Yang, R. Zhang and K. Rose, "Drift management and adaptive bit rate allocation in scalable video coding," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, vol. 2, (Rochester, NY), pp. 49–52, September 2002.

[123] S. Dogan, A. Cellatoglu, M. Uyguroglu, A. H. Sadka and A. M. Kondoz, "Error-Resilient Video Transcoding for Robust Internetwork Communications Using GPRS," *IEEE Transactions on Circuits and Systems for VideoTechnology - Special Issue on Wireless Video*, vol. 12, pp. 453–464, July 2002.

[124] H.G. Musmann, P. Pirsch and H.J. Grallert, "Advances in picture coding," *Proceedings of the IEEE*, vol. 73, pp. 523–548, April 1985.

[125] M. Flierl and B. Girod, "Generalized B Pictures and the Draft H.264/AVC Video-Compression Standard," *IEEE Transaction on Circuits and Systems for Video Technology*, vol. 13, pp. 587–597, July 2003.

[126] T. Shanableh and M. Ghanbari, "Loss Concealment Using B-Pictures Motion Information," *IEEE Transaction on Multimedia*, vol. 5, pp. 257–266, June 2003.

[127] M. Al-Mualla, N. Canagarajah and D. Bull, "Simplex Minimization for Single- and MultipleReference Motion Estimation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, pp. 1029–1220, December 2001.

[128] G. S. A. Luthra and T. Wiegand, "special issue on the H.264/AVC video coding standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, July 2003.

[129] S. Wenger, "H.264/AVC Over IP," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, pp. 587–597, July 2003.

[130] M. H. T. Stockhammer and T. Wiegand, "H.264/AVC in Wireless Environments," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, pp. 587–597, July 2003.

[131] A. Arumugam, A. Doufexi, A. Nix and P. Fletcher, "An investigation of the coexistence of 802.11g WLAN and high data rate Bluetooth enabled consumer electronic devices in indoor home and office environments," *IEEE Transactions on Consumer Electronics*, vol. 49, pp. 587–596, August 2003.

[132] R. Thobaben and J. Kliewer, "Robust Decoding of Variable-Length Encoded Markov Sources Using a Three-Dimensional Trellis," *IEEE communications letters*, vol. 7, pp. 320–322, July 2003.

[133] A. Murad and T. Fuja, "Joint Source-Channel Decoding of Variable-Length Encoded Sources," in *IEEE Information Theory Workshop*, (Killarney, Ireland), pp. 94–95, June 1998.

[134] L. Chiariglione, "The development of an integrated audiovisual coding standard: MPEG," *Proceedings of the IEEE*, vol. 83, pp. 151–157, February 1995.

[135] R. Schäfer and T. Sikora, "Digital video coding standards and their role in video communications," *Procedings of the IEEE*, vol. 83(10), pp. 907–924, June 1995.

[136] D. J. Le Gall, "The MPEG video compression algorithm," *Signal Processing: Image Communication*, vol. 4, pp. 129–140, 1992.

[137] T. Sikora, "MPEG-4 very low bit rate video," *Proceedings of IEEE ISCAS Conference, Hong Kong*, pp. 1440–1443, February 1997.

[138] T. Sikora, "The MPEG-4 video standard verification model," *IEEE Transactions on Circuit and Systems for Video Technology*, vol. 7, pp. 19–31, February 1997.

[139] ISO/IEC JTC1/SC29/WG11 N0702 Rev., "Information technology - Generic coding of moving pictures and associated audio, Recommendation H.262. Draft International Standard," vol. 83, March 1994.

[140] ISO/IEC 11172-2 Information technology, "Coding of moving pictures and associated audio for digital storage media at up to about 1.5Mbit/s - Video. Standards Organization/International Electrotechnical (in German). International Commission," 1993.

[141] MPEG AOE Group, "Proposal package description (PPD)-Revision 3," July 1995.

[142] ISO/IEC JTC1/SC29/WG11, "Information technology - Coding of moving pictures and associated audio for digital storage media at up to1.5Mbits/s. Part 2: Video. Draft ISO/IEC 11172-2 (MPEG-1)," *ISO/IEC*, 1991.

[143] ISO/IEC JTC1/SC29/WG11, "Information technology - Generic coding of moving pictures and associated audio. Part 2: Video. Draft ISO/IEC 13818-2 (MPEG-2) and ITU-T Recommendation H.262, ISO/IEC and ITU-T," *ISO/IEC*, 1994.

[144] ISO/IEC JTC1/SC29/WG11, "Information technology - Generic coding of audio-visual objects. Part 2: Visual. Draft ISO/IEC 14496-2 (MPEG-4), version 1," 1998.

[145] A. Jain, *Fundamentals of Digital Image Processing.* Wnglewood Cliffs, NJ: Prentice-Hall, 1989.

[146] O. Avaro, A. Eleftheriadis, C. Herpel, G. Rajan, L. Ward, "MPEG-4 Systems: Overview," *Signal Processing: Image Communication*, vol. 15, pp. 281–298, 2000.

[147] G. Franceschini, "The delivery layer in MPEG-4," in *Signal Processing: Image Communication*, vol. 15, pp. 347–363, 2000.

[148] C. Herpel, "Architectural Considerations for Carriage of MPEG-4 over IP Network," *ISO/IEC JTC1/SC29/WG11 N2615*, December 1998.

[149] C. Herpel, A. Eleftheriadis, "MPEG-4 Systems: Elementary stream management," *Signal Processing: Image Communication*, vol. 15, pp. 299–320, 2000.

[150] R. Talluri, "Error-Resilient Video Coding in the ISO MEPG-4 Standard," *IEEE Communications Magazine*, pp. 112–119, June 1998.

[151] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications," *RFC 1889*, January 1996.

[152] L. Chiariglione, "MPEG and Multimedia Communications," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 7, pp. 5–18, February 1997.

[153] K.N. Ngan, T. Sikora, M-T. Sun and S. Pamchanathan, "Segmentation, description and retrival of video content," *IEEE Transactions on Circuits and Systems for Video Technology, special issue*, vol. 8(5), pp. 521–524, September 1998.

[154] K.N. Ngan, T. Sikora, M-T. Sun and S. Pamchanathan, "Representation and coding of images and video," *IEEE Transactions on Circuits and Systems for Video Technology, special issue*, vol. 8, pp. 797–801, November 1998.

[155] ISO/IEC 13818-2 MPEG-2 Video Coding Standard, "Information technology - Generic coding of moving pictures and associated audio information: Video," March 1995.

[156] T. Sikora and L. Chiariglione, "MPEG-4 Video and its potential for future multimedia services," *Proceedings of IEEE ISCAS Conference, Hong Kong*, vol. 2, pp. 1468–1471, June 1997.

[157] F. Bossen, T. Ebrahimi, "A simple and efficient binary shape coding technique based on bitmap representation," *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP'97), Munich, Germany*, vol. 4, pp. 3129–3132, April 1997.

[158] T. Ebrahimi, C. Horne, "MPEG-4 natural video coding - An overview," *Signal Proceesing: Image Communication*, vol. 15, no. 4, pp. 365–385, 2000.

[159] ISO/IEC JTC1/SC29/WG11 N1902, "Information Technology - Coding of Audio Visual Objects: Visual," October 1997.

[160] Recommendation ITU-T BT.500-11, "Methodology for the subjective assessment of the quality of television pictures," in *ITU-T*, 2002.

[161] ITU-T/SG 16/VCEG(formerly Q.15 now Q.6), "H.26L Test Model Long Term Number 7 (TML-7), Doc. VCEG-M81," April 2001.

[162] W. Choi and B. Jeon, "Dynamic UVLC codeword remapping with fixed re-association table for H.26L," in *Picture Coding Symposium(PCS)*, (Seoul, Korea), pp. 167–170, April 2001.

[163] D. Marpe, G. Blattermann, G. Heising, and T. Wiegand, "Further results for CABAC entropy coding scheme," Document VCEG-M59, ITU-T Video Coding Experts Group, Apr. 2001, http:// standards.pictel.com/ ftp/ video-site/ 0104 Aus/ VCEG-M59.doc."

[164] R.J. Clarke, "Transform Coding of Images," in *Microelectronics and Signal Processing. Academic Press*, (London), 1985.

[165] T. N. N. Ahmed and K. Rao, "Discrete Cosine Transform," *IEEE Transactions on Computers*, pp. 90–93, January 1974.

[166] ITU-T Rec. H.26L/ISO/IEC 11496-10, "Advanced Video Coding," *Final Committee Draft, Document JVT-E022*, September 2002.

[167] D. Marpe, G. Blättermann, G. Heising, and T. Wiegand, "Adaptive Codes for H.26L," *ITU-T SG16/Q.6 VCEG-L-13*, January 2001.

[168] ITU-T, "Video coding for low bitrate communication," *ITU-T Recommendation H.263; version 1*, November 1995.

[169] J. Signes, Y. Fisher, and A. Eleftheriadis, "MPEG-4's Binary Format for Scene Description," in *Signal Processing: Image Communication, Special issue on MPEG-4*, vol. 15, pp. 312–345, Jannuary 2000.

[170] A.M. Tekalp and J. Ostermann, "Face and 2-D mesh animation in MPEG-4," in *Signal Processing: Image Communication, Special issue on MPEG-4*, vol. 15, pp. 387–421, Jannuary 2000.

[171] ISO/IEC JTC1/SC29/WG11 N1902, "Information Technology - Coding of Audio Visual Objects: Visual," November 1998.

[172] ISO/IEC JTC1/SC29/WG11, "Adhoc Group on Core Experiments on Error Resilience aspects of MPEG-4 video, Description of Error Resilience aspects of MPEG-4 video," *Description of Error Resilience Core Experiments*, November 1996.

[173] J.G. Proakis, *Digital Communication. 3rd ed.* McGraw-Hill, New York, 1995.

[174] S. B. Wicker, *Error Control Systems for Digital Communication and Storage.* Englewood Cliffs, NJ: Prentice Hall, 1994.

[175] A. Andreadis, G. Benelli, A. Garzelli, S. Susini, "FEC coding for H.263 compatible video transmission," *Proceesings of International Conference on Image Processing, Santa Barbara, CA*, pp. 579–581, October 1997.

[176] J. Wen, J.D. Villasenor, "A class of reversible variable length codes for robust image and video coding," *Proceedings 1997 IEEE International Conference on Image Processing*, vol. 2, pp. 65–68, October 1997.

[177] H. Sun, J. W. Zdepski, W. Kwok, and D. Raychaudhuri, "Error Concealment for Robust Decoding of MPEG Compressed Video," in *Signal Processing: Image Communication*, vol. 10(4), pp. 249–268, September 1997.

[178] A. Li, S. Kittitornkun, Y.H. Hu, D.S. Park, and J. Villasenor, "Data Partitioning and Reversible Variable Length Codes for Robust Video Communications," in *IEEE Data Compression Conference Preceeding*, (Snowbird, Utah), pp. 460–469, March 2000.

[179] B. L. Montgomery, J. Abrahams, "Synchronization of binary source codes," in *IEEE Transactions on Information Theory*, vol. 32, pp. 849–854, November 1996.

[180] Y. Takishima, M. Wada, H. Murakami, "Reversible variable length codes," *IEEE Transections on Communications*, vol. 43, pp. 158–162, February 1995.

[181] R. Talluri, I. Moccagatta, Y. Nag, G. Cheung, "Error concelment by data partitioning," *Signal Processing Magazine*, vol. 14, pp. 505–518, May 1999.

[182] C. Bormann, L. Cline, G. Deisher, T. Gardos, C. Maciocco, D. Newell, J. Ott, G. Sullimendation, S. Wenger, C. Zhu, "RTP Payload format for the 1998 version of ITU-T recommendation H.263 video (H.263+); Request for Comments 2429l," , May 1998.

[183] J. Y. Chung, S. X. Ng and L. Hanzo, "Burst-by-burst adaptive coded modulation-aided joint detection-based CDMA for wireless video telephony," in *IEEE Vehicular Technology Conference*, (Birmingham, USA), pp. 1317–1321, May 2002.

[184] G. Ungerböck, "Channel coding with multilevel/phase signals," *IEEE Transactions on Information Theory*, vol. 28, pp. 55–67, January 1982.

[185] P. Robertson, T. Wörz, "Bandwidth-Efficient Turbo Trellis-Coded Modulation Using Punctured Component Codes," *IEEE Journal on Selected Areas in Communications*, vol. 16, pp. 206–218, February 1998.

[186] C. Berrou, A. Glavieux and P. Thitimajshima, "Near Shannon Limit Error-Correcting Coding and Decoding : Turbo Codes," in *Proceedings of IEEE International Conference on Communications*, (Geneva, Switzerland), pp. 1064–1070, May 1993.

[187] E. Zehavi, "8-PSK trellis codes for a Rayleigh fading channel," *IEEE Transactions on Communications*, vol. 40, pp. 873–883, May 1992.

[188] X. Li, J. A. Ritcey, "Bit-interleaved coded modulation with iterative decoding using soft feedback," *IEE Electronics Letters*, vol. 34, pp. 942–943, May 1998.

[189] C.H. Wong and L. Hanzo, "Upper-bound performance of a wideband burst-by-burst adaptive modem," *IEEE Transactions on Communications*, vol. 48, pp. 367–369, March 2000.

[190] A.J. Goldsmith and S. Chua, "Adaptive coded modulation for fading channels," *IEEE Transactions on Communications*, vol. 46, pp. 595–602, May 1998.

[191] A. Klein, G. K. Kaleh, and P. W. Baier, "Zero forcing and minimum mean square error equalization for multiuser detection in code division multiple access channels," *IEEE Transactions on Vehicular Technology*, vol. 45, pp. 276–287, May 1996.

[192] G. H. Golub and C. F. van Loan, *Matrix Computations.* North Oxford Academic, 1983.

[193] S. Lin and D. J. Costello, Jr, *Error Control Coding: Fundamentals and Applications.* Inc. Englewood Cliffs, New Jersey 07632: Prentice-Hall, 1983.

[194] J. Hagenauer, "Rate-Compatible Punctured Convolutional Codes and their Applications," *IEEE Transactions on Communication*, vol. 36, pp. 389–400, April 1988.

[195] L. H. C. Lee, "New Rate-Compatible Puncture Convolutional Codes for Viterbi Decoding," *IEEE Transactions on Communication*, vol. 42, pp. 3073–3079, December 1994.

[196] P. Robertson, E. Villebrun and P. Höher, "A Comparison of Optimal and Sub-Optimal MAP Decoding Algorithms Operating in Log Domain," in *Proceedings of the International Conference on Communications*, pp. 1009–1013, June 1995.

[197] L. R. Bahl, J. Cocke, F. Jelinek and J. Raviv, "Optimal Decoding of Linear Codes for Minimizing Symbol Error Rate," *IEEE Transactions on Information Theory*, vol. 20, pp. 284–287, March 1974.

[198] ITU-T, "Recommendation H.263: Video Coding for Low Bitrate communication," , March 1998.

[199] Special Mobile Group of ETSI, "UMTS: Selection procedures for the choice of radio transmission technologies of the UMTS," tech. rep., European Telecommunications Standard Institute (ETSI), France, 1998.

[200] C. Berrou, A. Glavieux and P. Thitimajshima, "Near Shannon Limit Error-Correcting Coding and Decoding: Turbo Codes," in *Proceedings of the International Conference on Communications*, (Geneva, Switzerland), pp. 1064–1070, May 1993.

[201] R. Gallager, "Low Density Parity Check Codes," *IRE Transactions On Information Theory*, vol. IT-8, pp. 21–28, January 1962.

[202] F. Guo, S. X. Ng and L. Hanzo, "LDPC assisted Block Coded Modulation for Transmission over Rayleigh Fading Channels," in *IEEE Vehicular Technology Conference*, (Jeju, Korea), April 2003.

[203] M. Y. Alias, F. Guo, S. X. Ng, T. H. Liew and L. Hanzo, "LDPC and Turbo Coding Assisted Space-Time Block Coded OFDM," in *IEEE Vehicular Technology Conference*, (Jeju, Korea), April 2003.

[204] M.C. Davey, "Error-correction using low density parity check codes," *Ph.D thesis, University of Cambridge, UK*, 1999.

[205] G. Bjontegaard, T. Wiegand, "H.26L Test Model Long Term Number 8 (TML-8) draft 0," June 2001.

[206] R. van Nee and R. Prasad, *OFDM For Wireless Multimedia Communications*. Boston: Artech House Publishers, 2000.

[207] B. Lu, X. Wang and K. R. Narayanan, "LDPC-Based Space-Time Coded OFDM Systems Over Correlated Fading Channels: Performance Analysis and Receiver Design," *IEEE Transactions On Communications*, vol. 50(1), pp. 74–88, January 2002.

[208] H. Futaki and T. Ohtsuki, "Low-Density Parity-Check (LDPC) Coded OFDM Systems," in *Proceedings of IEEE VTC Fall 2001*, (Atlantic City, USA), pp. 82–86, May 2001.

[209] S. M. Alamouti, "A Simple Transmit Diversity Technique for Wireless Communications," *IEEE Journal on Selected Areas in Communications*, vol. 16(8), pp. 1451–1458, October 1998.

[210] D. J. C. MacKay and R. M. Neal, "Near Shannon Limit Performance of Low Density Parity Check Co des," *Electronics Letters*, vol. 32, pp. 1645–1646, August 1996.

[211] L. Ping and W. K. Leung, "Decoding Low Density Parity Check Codes wih Finite Quantization Bits," *IEEE Communications Letters*, vol. 4, pp. 62–64, February 2000.

[212] ISO/IEC JTC1/SC29/WG11 W2502, "ISO/IEC 14496-2.," in *Final Draft International Standard. Part 2: Visual*, (Atlantic City), 1998.

[213] M. C. Reed, *Iterative Receiver Techniques for Coded Multiple Access Communication System.* PhD thesis, The University of South Australia, 1999.

[214] L. Hanzo, L. L. Yang, E. L. Kuan, and K. Yen, *Single- and Multi-Carrier DS-CDMA.* John Wiley and IEEE Press, 2003, 1060 pages.

[215] Y. Zhang, "Reduced complexity iterative multiuser detection for DS/CDMA with FEC," *International Conference on Universal Personal Communications*, pp. 10–14, October 1997.

[216] A. J. G. P. D. Alexander and M. C. Reed, "Performance analysis of an iterative decoder for code-division mulitple-access," *European Transactions on Telecommunication*, vol. 9, pp. 419–426, Sept/Oct 1998.

[217] L. Hanzo, T. H. Liew, and B. L. Yeap, *Turbo Coding, Turbo Equalisation and Space-Time Coding for Transmission over Fading Channels.* John Wiley-IEEE Press, 2002.

[218] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and encoding: Turbo-codes (1)," in *IEEE International Conference on Communications*, (Geneva, Switzerland), pp. 1064–1070, May 1993.

[219] S. Verdu, *Multiuser Detection.* Cambridge Press, 1998.

[220] S. M. Alamouti, "A simple transmit diversity technique for wireless communications," *IEEE Journal on Selected Areas in Communications*, vol. 16, pp. 1451–1458, October 1998.

[221] L.-L. Yang and L. Hanzo, "Space-Time Spreading Assisted Broadband MC DS-CDMA," in *Proceedings of IEEE VTC'2002, Spring*, (Birmingham, Alabama, USA), May 2002.

[222] B. Hochwald, T. L. Marzetta, and C. B. Papadias, "A Transmitter diversity scheme for wideband CDMA systems based on Space-time spreading," *IEEE Journal on Selected Areas in Communications*, vol. 19, pp. 48–60, January 2001.

[223] L. Hanzo, M. Münster, B. J. Choi, and T. Keller, *OFDM and MC-CDMA*. John Wiley and IEEE Press, 2003, 970 pages.

[224] C. Tang, "An Intelligent Learning Scheme for Adaptive Modulation," *In Proceedings of the IEEE Vehicular Technology Conference*, pp. 718–719, Oct 2001.

[225] J. Torrance and L. Hanzo, "Optimisation of switching levels for adaptive modulation in a slow Rayleigh fading channel," *Electronics Letters*, vol. 32, pp. 1167–1169, 20 June 1996.

[226] B. Haskell, A. Puri, and L. Rabiner, "Image and Video Coding-Emerging Standards and beyond," vol. 8, pp. 814–837, November 1998.

[227] V. Tarokh, N. Seshadri and A. R. Calderbank, "Space-time codes for high rate wireless communication: Performance analysis and code construction," *IEEE Transactions on Information Theory*, vol. 44, pp. 744–765, March 1998.

[228] T. Hindelang, J. Hagenauer, and S. Heinen, "Source-controlled channel decoding: estimation of correlated parameters," in *ITG-Fachbericht, Conference Source and Channel Coding*, vol. 159, (Munich, Germany), pp. 251–258, January 2000.

[229] R. Bauer and J. Hagenauer, "Symbol by symbol map decoding of variable length codes," in *3rd ITG Conference Source and Channel Coding*, vol. 159, (Munich, Germany), pp. 111–116, January 2000.

Index

Author Index

M

N

O

P

Q

R

S

T