WIRELESS VIDEO BROADCAST TO FIXED AND MOBILE RECEIVERS

By
Chee Siong, Lee
B.Eng.(Hons)

A thesis submitted for the degree of
Doctor of Philosophy

Supervisor: Prof. Lajos Hanzo
M.Sc., Ph.D, SMIEEE

Department of Electronics and Computer Science,
University of Southampton,
United Kingdom.

March 2001

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

FACULTY OF ENGINEERING

ELECTRONICS AND COMPUTER SCIENCE DEPARTMENT

Doctor of Philosophy

Wireless Video Broadcast to Fixed and Mobile Receivers

by Chee Siong, Lee

In this thesis, we explored the feasibility of providing broadcast video services to fixed and mobile users. The video codec used in our study was either the MPEG-2 or the MPEG-4 video codec. The MPEG-2 video bitstream was subjected to a rigorous bit error sensitivity investigation, in order to assist in contriving various error protection schemes for wireless broadcast video transmission. Turbo-coded performance enhancement of the terrestrial Digital Video Broadcast (DVB) system was proposed for transmission over mobile channels to receivers on the move. The turbo codec was shown to provide substantial performance advantages over conventional convolutional coding both in terms of bit error rate and video quality. We have also proposed similar enhancements for the satellite-based DVB system.

In order to increase the resilience of the video codec, we have investigated data partitioning in conjunction with multi-class source sensitivity matched error correction coding. However, our experiments suggested that multi-class data partitioning did not result in error resilience improvements, since a high proportion of relatively sensitive video bits had to be relegated to the lower integrity subchannel, when invoking a powerful low-rate channel codec in the high-integrity protection class. Nonetheless, terrestrial DVB transmission to mobile receivers is feasible, when using turbo-coded OFDM transceivers at realistic power-budget requirements under the investigated highly dispersive fading channel conditions.

We then introduced coded modulation schemes, namely Trellis Coded Modulation (TCM), Turbo Trellis Coded Modulation (TTCM), Bit-Interleaved Coded Modulation (BICM) and Iterative Decoded Bit-Interleaved Coded Modulation (BICM-ID). The associated coded modulation schemes provided performance improvements in terms of spectral efficiency and decoding complexity.

Finally, the performance of the MPEG-4 error resilience tools was characterised, which exhibited limited robustness, when operating in terrestrial DVB transmission scenarios. We have proposed a simple packetisation procedure based on an adaptive intra-coded macroblock update scheme, in order to enhance the resilience of the video codec. The proposed scheme is capable of tolerating up to 5 % packet loss ratio.

Contents

Acknowledgements v

List of Publications vi

Chapter 1 Overview 1
- 1.1 Motivation . . . 1
- 1.2 Overview of Block-Based Video Compression Techniques . . . 2
- 1.3 Video Error Resilience Issues . . . 7
- 1.4 Video Transmission to Mobile Receivers . . . 14
- 1.5 Organisation of Thesis . . . 15
- 1.6 Summary . . . 16

Chapter 2 Terrestrial Digital Video Broadcasting 17
- 2.1 Background and Motivation . . . 17
- 2.2 MPEG-2 Bit Error Sensitivity . . . 18
- 2.3 DVB Terrestrial Scheme . . . 29
- 2.4 Channel Model . . . 31
- 2.5 Data Partitioning Scheme . . . 33
- 2.6 Performance of the Data Partitioning Scheme . . . 39
- 2.7 Performance of the DVB Terrestrial Scheme Employing Non-hierarchical Transmission . . . 50
- 2.8 Performance of the DVB Terrestrial Scheme Employing Hierarchical Transmission . . . 56
- 2.9 Conclusions and Future Work . . . 61

Chapter 3 Satellite Single-Carrier Digital Video Broadcasting 63
- 3.1 Background and Motivation . . . 63
- 3.2 DVB Satellite Scheme . . . 64
- 3.3 Channel Model . . . 66
- 3.4 The Blind Equalisers . . . 67
- 3.5 Performance of the DVB-S Scheme . . . 70
  - 3.5.1 Transmission Over the Symbol-spaced Two-path Channel . . . 71
  - 3.5.2 Transmission Over the Two-symbol Delay Two-path Channel . . . 77
  - 3.5.3 Performance Summary of the DVB-S System . . . 77

3.6 Conclusions and Future Work 85

Chapter 4 Coded Modulation Assisted OFDM DVB Systems 86
4.1 Introduction 86
4.2 System Overview 87
4.3 Channel Model 89
4.4 Simulation Results and Discussions 90
4.5 Summary 93

Chapter 5 MPEG-4 Based Digital Video Broadcasting 98
5.1 Overview 98
5.2 Functionalities and Requirements 99
5.2.1 Content-Based Interactivity 100
5.2.2 Universal Access 101
5.2.3 Compression 101
5.3 MPEG-4 System Architecture 102
5.3.1 Binary Format for Scenes 102
5.3.2 Object Descriptor Framework 104
5.3.3 Elementary Stream Management 105
5.3.4 System Decoder Model 106
5.3.5 MPEG-4 Delivery Multimedia Integration Framework 108
5.3.6 Summary of MPEG-4 System Architecture 110
5.4 MPEG-4 Natural Video Coding 110
5.4.1 Shape Coding 114
5.4.1.1 Binary Shape Coding 115
5.4.1.2 Grey Scale Shape Coding 117
5.4.2 Motion Estimation and Compensation 117
5.4.3 Texture Coding 121
5.4.4 Scalable Coding 123
5.4.4.1 Spatial Scalability 125
5.4.4.2 Temporal Scalability 126
5.4.5 Summary of MPEG-4 Natural Video Coding 128
5.5 Performance of the MPEG-4 Codec 128
5.6 MPEG-4 Error Resilience Features 137
5.6.1 Resynchronisation 137
5.6.2 Data Partitioning 139
5.6.3 Reversible Variable Length Codes 139
5.6.4 Performance of the MPEG-4 Error Resilience Tools Within DVB Terrestrial Transmission Scenario 141
5.6.5 Packet Based MPEG-4 Video Data Transmission 147
5.6.6 Performance of the Packet Based MPEG-4 Video Data Transmission 151
5.7 Summary 153

Chapter 6 Conclusions and Suggestions for Future Work 159
6.1 Summary 159
6.2 Suggestions for Further Work 162

Glossary 164

Bibliography 167

Acknowledgements

Every endeavour that we have made is in general not a solitary process. This thesis is no exception. The following people has greatly assisted me in my research in Southampton:

- Prof. Lajos Hanzo, who is my PhD supervisor, has always provided the encouragement and guidance throughout my stay in Southampton.
- Thomas Keller has contributed the OFDM modem, as well as the channel model used in the terrestrial DVB experiments of Chapters 2 and as well as in Section 5.6.4.
- Spyros Vlahoyiannatos has contributed the blind equalisers employed in the satellite DVB system of Chapter 3. He has also contributed Section 3.4 of this thesis for the benefit of the readers.
- Lorenzo Piazzo has contributed the trellis coded modulation and turbo trellis coded modulation schemes for the study on coded modulation OFDM broadcast systems in Chapter 4.
- Michael S.X. Ng has contributed the bit-interleaved coded modulation and iterative decoded bit-interleaved coded modulation schemes for the study on coded modulation broadcast systems in Chapter 4.

I am also grateful to my other colleagues in the Communications Group for their help and friendship. Many thanks to Peter J. Cherriman for always sharing his Linux enthusiasm with me. Last but not least, I would like to thank my parents for their continual support during my studies. You have instilled in me the love for knowledge and this thesis is dedicated to you both.

List of Publications

1. C.S. Lee, T. Keller and L. Hanzo, "Turbo-coded digital video broadcasting for mobile environments", *European Conference on Multimedia Communications and Services (ECMCS'99)*, Kraków, Poland, 24–26 June, 1999.
2. C.S. Lee, T. Keller and L. Hanzo, "OFDM-based turbo-coded hierarchical and non-hierarchical terrestrial mobile digital video broadcasting", *IEEE Transactions on Broadcasting*, vol. 46, no. 1, pp. 1–22, March, 2000.
3. C.S. Lee, S. Vlahoyiannatos and L. Hanzo, "Satellite based turbo-coded, blind-equalized 4-QAM and 16-QAM digital video broadcasting", *IEEE Transactions on Broadcasting*, vol. 46, no. 1, pp. 23–33, March, 2000.
4. C.S. Lee, T. Keller and L. Hanzo, "Turbo-coded terrestrial mobile video broadcasting", International Conference on Telecommunications (ICT) 2000, Acapulco, Mexico, 22–25 May, 2000, vol. 1, pp. 359–363.
5. C.S. Lee, S. Vlahoyiannatos and L. Hanzo, "Blind-equalised and turbo-coded satellite based digital video broadcasting", International Conference on Telecommunications (ICT) 2000, Acapulco, Mexico, 22–25 May, 2000, vol. 1, pp. 562–566.
6. C.S. Lee, S.X. Ng, L. Piazzo and L. Hanzo, "TCM, TTCM, BICM and Iterative BICM assisted OFDM-based digital video broadcasting to mobile receivers", Proceedings of Vehicular Technology Conference (VTC) Spring 2001, Rhodes, Greece, 6–9 May, 2001.
7. C.S. Lee, L. Hanzo, T. Keller and S. Vlahoyiannatos, "QAM-based Video Broadcast Systems", Chapter 24 of *Single- and Multi-carrier Quadrature Amplitude Modulation: Principles and Applications for Personal Communications, WLANs and Broadcasting, L. Hanzo, W. Webb and T. Keller, John Wiley & Sons, 2000*, pp. 637–681.
8. C.S. Lee, T. Keller and L. Hanzo, "DVB-T for Mobile Receivers", Section 21.6 of *Wireless Video Communications: Second to Third Generation Systems and Beyond, L. Hanzo, P.J. Cherriman and J. Streit, IEEE Press, 2001*, pp. 950–995.
9. C.S. Lee, S. Vlahoyiannatos and L. Hanzo, "Satellite Based Video Broadcasting", Section 21.7 of *Wireless Video Communications: Second to Third Generation Systems and Beyond, L. Hanzo, P.J. Cherriman and J. Streit, IEEE Press, 2001*, pp. 996–1017.

Chapter 1

Overview

1.1 Motivation

In recent years, video communication has attracted widespread research, which resulted in the standardization of several audio-visual coding standards, such as the H.261 [1], H.263 [2], MPEG-1 [3], MPEG-2 [4] and MPEG-4 [5] schemes. The most recent member of this codec family, namely the MPEG-4 scheme was developed for multimedia communication, including the transmission of still images, full-motion video, sound, text and graphics information [6]. Furthermore, the user can access these services in an intuitive, self-explanatory manner, just like using the Plain Old Telephone Services (POTS). Cox *et al.* claimed [6] that in order for multimedia communications to become a reality, the following issues have to be addressed:

- compression and coding of multimedia signals, which includes algorithmic issues, standardisation issues and transmission issues
- synthesis and recognition of multimedia signals, which includes speech, images, handwriting and text
- organization, storage and retrieval of multimedia signals
- accessing the multimedia signal by matching the machine to the user's preference
- searching and browsing multimedia material with the aid of machine intelligence through text, speech or image queries.

In this thesis, we shall be concentrating on the transmission of video signals, in particular MPEG-2 and MPEG-4 video within a wireless environment. The main application area of MPEG-2 is the compression of entertainment television programmes for satellite transmission, terrestrial broadcasting and cable television over fibre optic cables. In Europe, the satellite- and terrestrial-based video broadcasting schemes were standardized under the auspices of the Digital Video Broadcasting (DVB) project [7–13]. Furthermore, video-on-demand services over fibre optic cables and digital subscriber line based applications are also envisaged [14–17]. Another application area is known as videoconferencing, which will be supported by the emerging Broadband Integrated Services Digital Networks (B-ISDN), which are often associated with the Asynchronous Transfer Mode (ATM) networks [18, 19]. Last but not least, multimedia communication has also captured the interest of the cellular communications community. With the forthcoming third generation

mobile communication system, also known as the Universal Mobile Telecommunication System (UMTS), the mobile terminal is expected to handle data transmission at rates up to 2 Mbit/s in indoor scenarios and 384 kbit/s in outdoor mobile scenarios [20–26].

The MPEG-4 standard is more ambitious in terms of its aimed applications, than the MPEG-2 standard. A significant paradigm shift in MPEG-4 with respect to other audio-visual coding standards, such as H.261, H.263, MPEG-1 and MPEG-2, is the introduction of so-called "audio-visual objects" that can be defined both spatially and temporally, instead of coding rectangular video frames in conjunction with their associated audio. When we refer to a video object, we consider objects such as humans, cars, books or even the background of a scene. These objects can then be individually manipulated, moved and/or tracked in relation to one another, when forming a complete scene. In fact, a complete scene can be viewed as a composite object. We can also define audio objects, such as a man's voice, the noise generated by a car's engine and music. Hence, the combination of a car and its related engine noise will produce an audio-visual object. The object-based coding approach was adopted in order to provide the following facilities:

- **content-based interactivity** – this capability enables the user to interact directly with the individual objects, which constitute an audio-visual scene. Different objects can also have different grade of error protection during transmission leading to different quality of service, depending on the subjective importance of the objects [27]. The MPEG-4 committee also envisaged the composition and amalgamation of natural information sources with synthetic sources, such as computer generated data [28]. This type of audio-visual experience can elicit cognitive satisfaction from the viewer in the form of telepresence [28, 29].
- **universal access** – the two main functionalities addressed in this context are robustness against error prone transmission environments and content-based scalability. Error resilience has to be incorporated into the system, in order to enable compressed and hence vulnerable multimedia source signals to traverse through heterogeneous, error prone communication networks. The content scalability mentioned encompasses spatial resolution, temporal resolution, quality and complexity [30], which has to be maintained for ensuring that the encoded multimedia source signals can be decoded and displayed on different types of hardware.
- **compression** – research has shown [29, 31–40] that if the reconstruction of a three-dimensional video scene is possible, then a substantial video bitrate reduction can be achieved. Striking the right balance in terms of video quality, error resilience, implementational complexity, coding delay, bitrate, etc. is particularly important in battery operated, low cost consumer wireless communications devices.

Video sequences contain spatial redundancy as well as temporal redundancy [41–46]. Efficient video compression schemes exploit both of these properties. This is the topic of discussion for our next section.

1.2 Overview of Block-Based Video Compression Techniques

The reduction of the bitrate requirement associated with representing video scenes, which is often referred to as compression is achieved by removing redundant information from

the video signal at the encoder. In video signals, two main types of redundancy can be observed, namely spatial and temporal redundancy.

Specifically, spatial and temporal redundancy manifests itself in that pixel values which are adjacent to one another are correlated both within the same frame, as well as in consecutive frames [47]. The correlations are hence three-dimensional in their nature. Therefore, we can predict the value of a pixel, if the neighbouring pixels are known.

Another often used term is *psychovisual redundancy* which implies that our eyes have a limited ability to resolve fine spatial detail [43–45, 48]. As the visual angle subtended by the object that the eye is attempting to resolve becomes smaller, the intricate smaller detail becomes harder to perceive. Eventually, with increasing spatial complexity, the fine structure is no longer resolvable. In other words, the eye's spatial response falls to negligible values at a sufficiently high spatial frequency [46]. This fine spatial detail corresponds to high spatial frequencies components. Hence, by removing these higher spatial frequencies, we have removed those visual details, which the eye cannot detect. This phenomenon is exploited in transform coding, a technique to be exposed in more depth during our further discussions in this section.

Having discussed briefly the properties of the human eye which can be exploited for achieving bitrate compression, we shall now relate them to the design of a video codec. The block diagram of the MPEG-2 encoder and decoder structure is shown in Figure 1.2. We commence by concentrating on the encoder structure first. As mentioned earlier, pixel values which are adjacent to one another in time and space are correlated. The time domain correlation accrues from the fact that an object in a video sequence does not change significantly in consecutive video frames. Furthermore, adjacent pixels belonging to the same video frame typically have rather similar values, except for the edges of the video frame, where new objects may be introduced for example, due to camera panning. These features can be observed for example for Frames 0 and 5 of the uncoded "Miss America" video sequence shown in Figure 1.2. The noticeable difference is due to the facial movements around the mouth and the eyes of "Miss America". Other regions in the two images appear similar.

Since the previous image and the current image are similar, we can use the previous image as a direct prediction of the current image. Then, only the difference between the previous and the current images has to be coded. The amount of video information to be sent is significantly reduced. This technique is commonly referred to as frame differencing. However, objects in the image may move significantly, as time elapses. In order to increase the accuracy of our predictions, more sophisticated motion prediction techniques can be employed, as seen in Figure 1.2(a). Specifically, the previously coded image and the uncoded current image constitute the inputs of the motion prediction block. The motion prediction process will attempt to estimate the trajectory of the motion, which has occured during the time elapsed between the previous image and the current image. This motion trajectory is then described with the aid of the *motion vectors*, which describe the direction and extent of motion translation. The estimated motion vectors are then forwarded to the motion compensation process, which is based

(a) Frame 0 (b) Frame5

Figure 1.1: The original Frame 0 and Frame 5 of the "Miss America" video sequence.

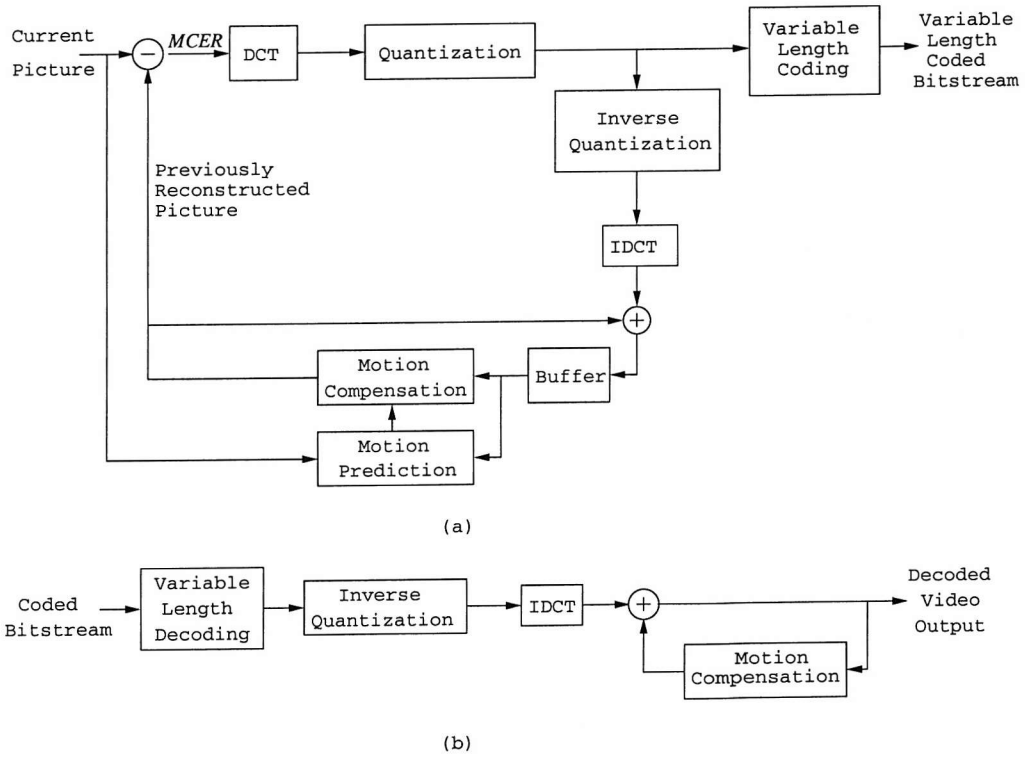


Figure 1.2: Block diagram of a MPEG-2 (a)encoder and (b)decoder employing motion compensation.

on the knowledge of this motion information and that of the previous image. Since this motion prediction is more accurate than simple frame differencing, the variance of the difference between the current image and the motion compensated predicted image will be smaller. This difference signal is also referred to as the Motion Compensated Error Residual (MCER), as shown in Figure 1.2. This process is responsible for removing the temporal correlation mentioned. The interested reader is referred to the literature [49–61] for a deeper discussion on various techniques employed in motion estimation.

The difference image contains the values of motion prediction errors at different pixel locations. These values will be typically similar for adjacent pixels. Thus, spatial correlation is exhibited. In order to remove this spatial redundancy, often transform coding [42, 45] is invoked. The transform coder transforms the representation of our data from the time domain into the frequency domain. In MPEG-2 and MPEG-4, this transformation is carried out by the so-called Discrete Cosine Transform (DCT) [62–64], which is labelled DCT in Figure 1.2. Since the difference signal is typically small and mostly zero, the DCT process can readily represent this information using mainly low-frequency information and limited energy high-frequency information, since the high-frequency component usually has a low intensity. Hence, in video compression parlance the DCT exhibits an efficient energy compaction property [45]. The DCT process does not in itself increase the achievable compression ratio. An efficient MCER quantisation process is required for removing the high-frequency components and hence to achieve compression. The quantised MCER is entropy encoded or, synonymously, variable-length encoded [46,47,65–69]. This method is used to reduce the number of bits required for encoding the signal, by assigning shorter codewords for representing frequently occuring values and longer codewords for rarely occuring values. In Figure 1.2 we observe that the quantised DCT values are also fed back to the encoder for inverse quantisation and Inverse Discrete Cosine Transformation (IDCT). The resultant reconstructed MCER is then added to the previous motion compensated prediction image and stored in a buffer to be used for future motion prediction.

In our brief discussion here, we have refrained from delving into so-called bitrate control issues. The bitrate control scheme is used to assist the video encoder in selecting the appropriate coding modes and quantiser step sizes based on a certain target video bitrate or video quality. The interested reader is referred to the literature [68–83] for further details on bitrate control issues.

The decoder, shown in Figure 1.2(b), performs the inverse of the encoder's tasks. First, the received bitstream is variable-length decoded. The output of this variable-length decoder is constituted by the quantised DCT coefficients, which are passed through the inverse quantiser. The resultant DCT coefficients are then inverse discrete cosine transformed for reconstructing the MCER in the time domain. The MCER is then added to the previous motion compensated image for forming the decoder's reconstructed image. It should be pointed out that the motion vector information estimated by the encoder is also encoded and incorporated into the transmitted bitstream. The decoder extracts this information and uses it for appropriately motion translating the previously reconstructed

image in order to form the motion compensated prediction of the current image. We also note that the reconstructed images at the decoder and the encoder's local decoder are identical, provided that no channel impairments have been inflicted on the encoded video bitstream.

In our discussions so far we have mentioned the employment of the previously reconstructed picture as prediction of the current picture, which relies on the estimated motion translation between the current picture and the previously reconstructed picture. However, this feedback process can only commence when a previously reconstructed picture becomes available. Since this coding mechanism makes use of the previous picture, it is termed as inter-frame coding. On the other hand, the first picture in a video sequence has to be encoded without reference to previous video frames, which is referred to as intra-frame coding.

The MPEG-2 standard defines three types of pictures. They are the so-called intra-coded pictures, forward predicted pictures and bidirectionally predicted pictures. They are summarised as below:

- **Intra-coded picture (I picture)**: Here, the picture is coded without any reference to any other pictures. Hence, without motion prediction, this method can only achieve moderate compression. Intra-coded pictures are necessary for providing 'random access' points into the video bitstream. This enables fast forward search features in video sequences as well as backward search. Since intra-frame pictures are not coded with respect to any other pictures, they also assist in preventing errors from propagating too far away from the corrupted picture, where they occured.
- **Predictive coded picture (P picture)**: This type of picture uses previous I or P pictures for prediction. This measure can increase the compression ratio of the coded picture as only the difference between the current picture and the reference picture is coded.
- **Bidirectionally predicted picture (B picture)**: This type of picture can use previous or future pictures for prediction. This type of picture offers the highest grade of compression. Since it requires future pictures as well as previous pictures for prediction, the encoder will rearrange the pictures with respect to their natural display order, so that the coded B pictures are only generated after the previous and future coded pictures have been transmitted.

Again, Figure 1.3 illustrates the relationship of the I, P and B pictures in terms of their prediction direction. In order to illustrate the order, in which the three different types of pictures are displayed, the following picture sequence is to be considered: IBBPBBPBBPBBPBBIBBPBBP... and so on. This order is the same as the order to be displayed to the viewer. Here, we display the first picture as an intra-coded picture (I picture). This is followed by bidirectionally predicted pictures (B pictures). In this context, the I picture is the previous picture and the P picture is the future picture. Hence, a B picture is predicted from these previous and future pictures. Since we know that the B picture depends on previous and future pictures, the order in which the picture sequence is transmitted now has to be rearranged to ensure that the previous and future pictures

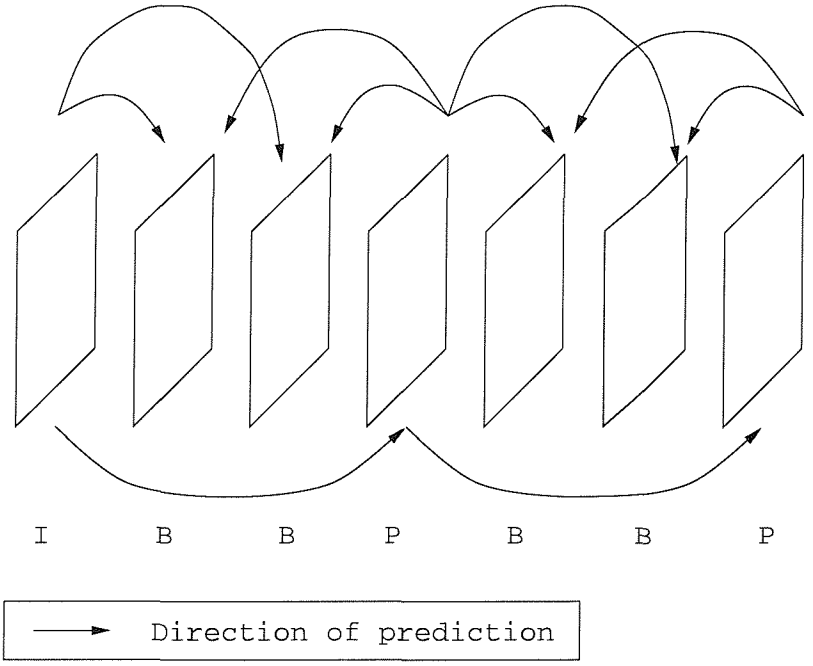


Figure 1.3: The relationship between I, P and B pictures in terms of prediction direction.

are received at the receiver, before the B picture to be reconstructed with their aid arrives. Hence, the picture sequence will be transmitted as IPBBPBBPBBPBBIPBBPBBPBB... and so on.

Lastly, before we conclude this section, there are several terms defined which need to be clarified, as they shall be encountered fairly often during our further discussion. They are the so-called block, macroblock and slice structures. A block is defined as a square containing 8×8 number of pixels. A macroblock comprises four blocks arranged as shown in Figure 1.4. Finally, a slice is constituted by a set of macroblocks. Figure 1.4 shows the different combination of macroblocks, forming slices that cover the whole picture.

An MPEG bitstream is then constituted by these DC DCT values, motion vectors, variable-length coded symbols and control information, which are required to successfully decode the bitstream. For more details on MPEG-2 video coding fundamentals, the reader is referred to the monographs by Mitchell *et al.* [69] and Haskell *et al.* [67]. The reader who is interested to delve deeper into the MPEG-4 standard and its various features can refer to the literatures [27, 28, 32, 84–99] for further information.

In summary, we have briefly discussed the basic techniques employed in the MPEG-2 and MPEG-4 video codec for compressing video signals. They are transform coding, quantisation, motion estimation, motion compensation and variable-length coding. In the following section, we will discuss the relationship between the video compression techniques highlighted in this section and the video error resilience issues.

1.3 Video Error Resilience Issues

The output bitstream of the MPEG encoder is packetised and transmitted. During their journey to the receiver, these packets may get lost or bits in the packet may become corrupted. In wireless communications the received signal may become corrupted by

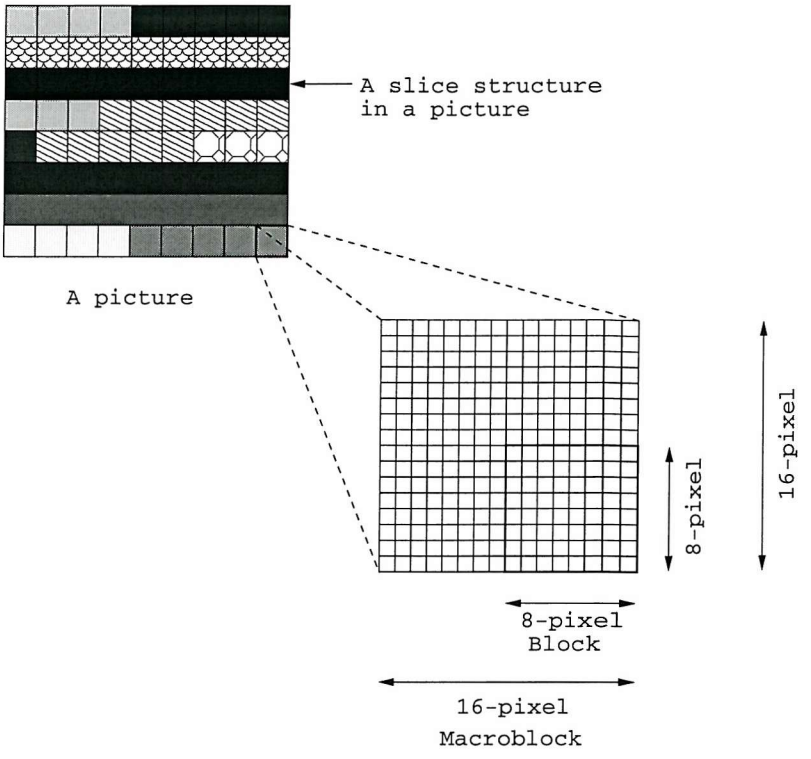


Figure 1.4: An example of the slice structure possible in a MPEG-2 coded picture. The dimensions of a macroblock and a block is also indicated.

bursts of errors, which is typical in dispersive Rayleigh fading channels. A Rayleigh fading channel model [23] is often used in wireless communications, where no line-of-sight (LOS) path exists between the base station and mobile station [100, 101]. Hence, waves that are transmitted by the base station and received at the mobile station may travel through different distances and experience different attenuations. The waves received from different paths will then have different amplitudes and phases, which may result in their constructive or destructive summation at the receiver antenna. When the summation of the incoming waves is constructive, we have a strong reception. However, if the summation is destructive, our received signal will be attenuated. As the mobile station travels, the signal received may move in and out of a fade. During the fade, the received signal will be riddled with bit errors, while when out of fade, the signal-strength will be high. This is the reason for the error bursts experienced in mobile radio transmissions. In satellite communications, the communication channel is typically less harsh compared to the mobile channel, as usually a line-of-sight path exists between the satellite and the receiver. In this case, the bit errors are a result of change in the property of the atmosphere, which may cause signal dispersion. Usually, the transmitted bitstream has parity bits attached to it in order to facilitate Forward Error Correction (FEC) and the resultant bitstream is channel interleaved for the sake of dispersing the bursts of errors.

With the basic knowledge gained from our brief discussions on the video encoding process and the typical communication channels, which the encoded video bitstream has to travel through, the reader's attention is now directed to the consequences of bit errors on the video bitstream's decodability. Earlier we mentioned that the MPEG video bitstream

comprises motion vectors, variable-length codes, DC DCT values and other control information. Since the decoding of one video frame is dependent on the correct decoding of the previous video frame, it can be seen that the corruption of any of this data will result in the loss of synchronization between the encoder and the decoder. Due to the motion compensation related dependency between video frames, an error which occurs in one frame will propagate to future frames. This error can only be removed, when the next correctly decoded intra-coded frame is received correctly, assuming the absence of channel-induced impairments of the encoded video data. There are several types of errors that the video decoder may exhibit. A range of possible decoding errors are listed below:

- In intra-frame coding, the picture frame is coded without any reference to other frames. Prior to encoding, each picture frame is divided into 8×8-pixel blocks and each block will have its own DC DCT coefficient value. The DC DCT coefficients refer to the average values of the luminance or chrominance components of the block-partitioned picture frame. Since the DC DCT coefficients of adjacent blocks do not vary significantly with respect to their neighbouring DC DCT coefficients, a differential encoding method is employed. The first block will have its DC DCT coefficient coded non-differentially, but the subsequent ones will have only their differences with respect to that of the previous DC DCT coefficient coded. This is a form of redundancy removal. In the event of corruption to one of the DC DCT coefficients, the subsequent DC DCT coefficients will also be corrupted, until a non-differential value is received.
- Similar problems exist for inter-coded frames as well. In this case, the motion vectors are also coded differentially, since the adjacent motion vectors exhibit similar values. This property, again, leads to error propagation in decoded motion vectors.
- The extensive use of variable-length codewords in the video codec in order to achieve a high compression ratio is the main contributing factor to its low error resilience. Since the successful decoding of the next variable-length codeword is dependent on the successful decoding of the current codeword, a one-bit error of the current one is sufficient to render subsequent codewords undecodable, until a suitable resynchronisation point in the video bitstream is reached.

Since decoding errors may result in poor visual quality, numerous techniques have been proposed to alleviate this predicament. These techniques can be grouped into three main approaches:

- error resilient source coding methods
- combined source and channel coding methods
- concealment methods.

We shall now briefly consider each technique in turn.

In **error resilient source coding techniques**, the video encoder converts the video signals into an error resilient representation. However, this is achieved at the expense of coding efficiency, as we shall see in our ensuing discussions. As we have seen in the context of MPEG-2, the DC values are encoded differentially with respect to the previous DC values. This method, however, leads to extensive error propagation, when one of the DC

values becomes corrupted. Therefore, it is advantageous in error resilience terms to limit the prediction to smaller image regions. Indirectly, this will also lead to an increase in the relative frequency of resynchronization locations for the decoder. This approach was adopted in the MPEG-4 error resilient mode, where the video is encoded into a so-called packet structure [5,88,102,103]. Each packet is coded such that its decoding is independent of the other packets. This prevents error propagation, when a packet is received in error. Furthermore, the motion and texture data are separated by a so-called motion marker. This motion marker is unique with respect to any possible combination of motion-related variable-length codes [5, 103]. Secondly, the variable-length codes used to encode the motion vectors are known as reversible variable-length codes (RVLC) [85, 104, 105].

The video encoder may also employ a more resilient motion prediction model. A novel motion prediction approach was introduced by Budagavi and Gibson [106], where multiple previous reconstructed frames were used by the encoder for motion prediction. By using multiple frames for motion prediction, the accuracy of prediction can be increased, when the search for similar macroblocks in the previous frame is extended to other, slightly older previous frames, which may give a better motion estimate. Hence, the bitrate required for transmission may be further reduced. More importantly, by having the capability to predict from other frames, the probability for a previous reconstructed macroblock to be selected as the best match was reduced. Therefore, in the event of errors in this macroblock, the chances of error propagation were reduced. A similar approach is also adopted in the H.263+ standard [107–109] in a scheme known as Reference Picture Selection (RPS). When employed in conjunction with a feedback scheme, the encoder can select its prediction frame from a pool of previous video frames. The prediction frame is only selected from a video frame, which has been successfully decoded by the decoder. However, if all the video frames in the reconstruction buffer suffer from some form of decoding errors, then the ability to select the reference picture alone is inadequate in curtailing error propagation, since the erroneously decoded portion of the frame can still be used for prediction. In order to circumvent this problem, the reference picture selection feature can also be combined with another feature known as Independent Segment Decoding (ISD) [109], where the video encoder and decoder treat the picture segment boundaries, such as slices or group of blocks, as picture boundaries. Hence, this prevents the use of corrupted data from outside the boundary for prediction.

The RPS and ISD schemes mentioned above rely on the decoder informing the encoder, which pictures or segments of the picture are available for prediction. If a feedback channel from the decoder to the encoder is not available, which is characteristic of a broadcast scenario, these schemes will not be of use. A possible solution is to let the encoder make the decision on coding a particular macroblock in intra- or inter-coded mode. The intra-coded macroblock mode is selected for the current macroblock, if the choice of inter-coded macroblock coding would be expected to lead to significant error propagation due to the corruption of the predicted macroblock in the previous picture. Various other researchers have reported their work in this area [110–115]. In [116], the authors adopted a slightly different approach. Specifically, they did not rely on intra-coded macroblocks alone, in

order to curtail error propagation. Instead, several buffers were introduced in order to capture the n previous decoded pictures. In a feedback scenario, if any part of the picture is erroneously decoded, the encoder retransmits the packet related to the erroneous portion of the picture. Since there is an inherent delay in the retransmission process, again, the authors had introduced several buffers for storing the decoded pictures, while waiting for the retransmitted packet to arrive. The buffers are necessary, in order to ensure that no video frames are lost and that they are displayed in their original order. Once the retransmitted packet arrives, the data is decoded and incorporated into the relevant picture. All the pictures in the buffer that depend on this updated picture will also be updated accordingly. This approach is capable of mitigating the error propagation effects, while observing the associated delay constraints. If a feedback channel is unavailable, FEC techniques can be used to combat the packet dropping problem in the event of network congestion or erroneous packets.

An alternative approach is the employment of overlapped block motion compensation (OBMC) [2, 60]. In OBMC, the prediction of a pixel is based on a weighted sum of predictions generated by different motion vectors. This so-called 'leaky' predictive scheme enables the dependency on past pixel values to fade away over time, potentially enhancing the codec's error resilience.

We shall also mention the approach proposed by Swann and Kingsbury [117, 118], where a transcoder transcodes the bitstream into a more error resilient structure. At the receiver, the inverse transcoding takes place and the corresponding output is fed into the MPEG-2 decoder. In order to achieve bitstream resynchronization, a technique known as Error Resilient Entropy Coding (EREC) was advocated in [119]. However, we refrain from elaborating on the EREC technique here, referring the interested reader to the contributions by Redmill and Kingsbury [120] for an in-depth treatment of the subject. Several researchers have also utilised EREC to increase the resilience of wavelet video codec [121, 122].

Another form of resilient video coding is based on the employment of **scalability mechanisms**. Specifically, bitstream scalability refers to the property of a bitstream, which allows the decoding of a subset of the entire received bitstream, in order to generate a complete picture having a resolution or quality that is proportional to the fraction of the bitstream decoded [67]. There are four scalability modes defined in MPEG-2, namely the so-called data partitioning, spatial scalability, temporal scalability and Signal-to-Noise Ratio (SNR) scalability. Specifically, data partitioning scalability refers to breaking up the original bitstream into separate bitstreams. For example, in a twin-layer data partitioning process the first partition, which is also often referred to as the base layer, may contain the more important decoding parameters, such as the motion vectors as well as the DC DCT coefficients. The second partition, which is also termed as the enhancement layer, may contain the high-frequency DCT coefficients. For the sake of maintaining spatial scalability, the base layer may host the image encoded at a lower spatial resolution, while the enhancement layer may contain the image encoded at a higher spatial resolution. By contrast, in temporal resolution terms, the enhancement layer may have a higher

temporal resolution than the base layer. The monograph by Haskell *et al.* [67] offers an elaborate discussion on the various scalability modes of MPEG-2. The reader may also find the contribution by Chiang and Anastassiou [123] informative, which discussed the various scalability modes of High Definition Television (HDTV). Although the scalability mechanism was not designed to increase the codec's error resilience, it also contributes towards this cause. In ATM networks the information to be transmitted is packetised into fixed-size packets, known as ATM cells. These cells may have either a high-priority or a low-priority. During network congestion, the low-priority ATM cells will be dropped first. The high-priority cells will contain information from the base layer, while the low-priority cells will contain information from the enhancement layer. In the event of packet loss, the received bitstream may still be decoded, albeit at a lower quality. Aravind *et al.* [124] provided a study of packet loss in the context of ATM networks.

Research into wireless transmission of video images has also seen the adoption of the above scalability techniques for combating transmission errors. Gharavi *et al.* [125–127] have contributed extensively in this area. Although Gharavi's initial scheme [125,126] was contrived in the context of ATM networks, the idea had also been extended to wireless environments [127]. In this contribution subband coding was used for separating the input image into different frequency subbands, which were the so-called low-low, low-high, high-low and high-high horizontal-vertical frequency subbands. The low-low frequency subband was separated and coded as the base layer. The other three subbands were encoded as the enhancement layer. The base layer and the enhancement layer were encoded independently of each other. The base layer bitstream, which has a higher priority, is typically protected by stronger FEC codes than the enhancement layer. As the base layer information had a stronger FEC, it had a better chance of surviving the burst errors as compared to the enhancement layer. When the enhancement layer was lost, the picture quality degraded, but the system remained capable of sustaining the link. The scalability approach was also invoked by Benzler [128] and Horn *et al.* [129].

The second major category of error resilience enhancements video schemes is known as the family of **joint source-channel coding** techniques. For example, Steinbach *et al.* [130] proposed an amalgam of error compensation, FEC and Automatic Repeat Request (ARQ) techniques. The FEC scheme was used to correct any bit errors at the decoder, while the ARQ was used to transmit an acknowledgement flag to the encoder, in order to indicate whether the frame had been received correctly. In case of negative acknowledgements, the encoder attempted to track the error that had propagated due to the incorrectly received frame and attempted to compensate for this error during the next frame. Cherriman and Hanzo [131,132] proposed a different ARQ-assisted programmable video transceiver. In this work, the ARQ scheme was also used to inform the encoder of the success or failure of the decoder in reconstructing the frame. The difference compared to the scheme advocated by Steinbach *et al.* [130] was that until the acknowledgement flag was received, the encoder's reconstructed frame buffer was not updated. When a transmitted video packet was received incorrectly, the encoder was informed of the region of the frame, where the error had occured. The encoder then updated the corresponding region

of the reconstructed frame buffer according to the contents of the packet acknowledged, if and only if this packet was successfully received. Otherwise, the corresponding video frame segment remained not updated. In case of a corrupted packet, the end-user noticed a small artifact, lasting for only one video frame duration, which would be corrected during the next video frame provided that the corresponding packet was successfully received. As the application concerned was duplex or interactive videophony over wireless channels, the acknowledgement flag was superimposed on the reverse-direction transmitted packet from user A back to the remote transmitter of user B. Hence, the reconstructed frame buffer update delay imposed was half a Time Division Duplex (TDD) frame duration.

Khansari *et al.* [133] employed a hybrid automatic repeat request scheme in addition to invoking a twin-rate scheme. In this case, only one bitstream was transmitted at any one time. If the channel quality was good, a higher bitrate bitstream was transmitted. When the channel conditions deteriorated below a certain level, a lower bitrate bitstream was transmitted. The scheme also made use of FEC, which gave rise to the hybrid nature of the method. The lower bitrate was achieved through re-quantisation of the finely quantised DCT coefficients. The authors argued that in order for the local decoder to always be in synchronization with the remote receiver's decoder, the motion prediction process at the local decoder had to be carried out using information exclusively from the lower bitrate bitstream. If the channel conditions degraded and the system had to revert back to the transmission of the low bitrate information, the encoder could still operate without modifications. Since this was essentially an ARQ-based system, the transmitted video packets were kept in an ARQ buffer, until a positive acknowledgement (ACK) flag was received from the decoder. If a negative acknowledgement (NACK) flag was received, the frame was retransmitted, until the maximum tolerable end-to-end delay was exceeded, in which case the frame was deemed as lost. The channel quality estimate was inferred from the ARQ buffer's fullness. Specifically, a threshold value was preset, such that if the buffer was filled above this threshold, the channel quality was deemed low and the transmitter resorted to sending lower bitrate frames.

Recent developments in this area included the adoption of the so-called soft decision decoding method, where the video decoder and the channel decoder operate in unison by passing not only hard decision bits to each other, but also the reliability of the bits. The above soft decision decoding can be performed in either a single-pass scenario [134–136] or in an iterative manner, whereby the channel decoder and source decoder pass decoded information back to each other, in order to enhance the capability of the partner decoder to decode the input bitstream [137–140]. Due to the popularity of variable-length coding schemes in the current video coding standards such as H.263, MPEG-2 and MPEG-4, various authors have also investigated the employment of reliability information in assisting the decoding of variable-length codewords [141–148].

The third class of error resilience enhancement techniques is based on **error concealment**. The basic idea behind error concealment is the reliance on blocks from neighbouring video frame regions in order to conceal the loss of a block. The MPEG-2 standard specifies a concealment scheme known as 'concealment motion vectors for intra-coded

pictures' [4, 67]. The aim of this technique was to alleviate the visual quality degradation of the reconstructed frames. In order for the concealment process to be effective, the 'concealment motion vector' of a macroblock was encoded together with the macroblock immediately below this macroblock. Otherwise, if this macroblock was lost, its motion vector was lost as well and hence it could not be used for concealment. Even though the visual quality degradation may be somewhat mitigated, there is some residual error in the reconstruction process. This error can still propagate to other frames. Tsai and Huang [149] had proposed several concealment schemes, which could prevent the error propagation by adopting different concealment methods for intra-frame coded pictures, forward-predicted pictures and bidirectionally-predicted pictures.

Kwok and Sun [150] advocated a spatial concealment algorithm, which took into account the correlated edge information of the surrounding corrupted image region. This technique produced a pleasing post-processed image quality.

Hemami and Meng [151] as well as Hong *et al.* [152] made use of the so-called smoothness constraints in reconstructing lost DCT coefficients, so that blocking artifacts were avoided in the reconstructed image. In the latter case, the authors took into account the edge information from neighbouring image regions in order to improve on the interpolation algorithm used to recover lost DCT coefficients.

In fact, there have been numerous studies in this area, where Kaiser and Fazel [153], Wang and Zhu [154], Debrunner *et al.* [155] and Wang *et al.* [107] have provided excellent reviews.

Nevertheless, it should be observed that error concealment methods perform best, when there is local information, which can assist in interpolating the lost pixel values. It is generally better to employ error concealment in conjunction with the other two error resilience improvement mechanisms mentioned earlier.

1.4 Video Transmission to Mobile Receivers

Following the above rudimentary introduction to video source compression, protection and enhancement methods in this section we briefly allude to some of the motivating factors behind designing systems for video broadcast to mobile receivers. Whilst at the time of writing the DVB standards [12, 13, 16] can be considered mature and the corresponding systems have been widely rolled out, it is a desirable objective to enhance their performance with the advent of recent research results achieved for example in the field of turbo channel coding [156].

We will show that the achievable broadcast system performance can be substantially improved with the aid of turbo channel coding [156–158]. The associated power budget improvement can be exploited in various ways. Firstly, the coverage area of a given broadcast station can be increased, rendering the service more cost-efficient. Alternatively, due to the increased coding gain of turbo codecs more vulnerable, but higher throughput modulation schemes can be employed. This allows us to increase the achievable bitrate in the bandwidth available, which in turn supports a higher video bitrate and higher video quality. Furthermore, in mobile scenarios the system's power budget has to accommodate

a high shadow- as well as fast-fading margin [23], which potentially necessitates the employment of receivers that are more powerful than originally envisaged for stationary DVB systems. Another important issue in the context of mobile DVB receivers is that while stationary fixed receivers benefit from the employment of directional antennas, mobile receivers have to be equipped with lower gain omnidirectional antennas. Omnidirectional antennas degrade the system's power budget and hence the extra coding gain provided by turbo codecs is vital for the system's operation. Finally, with the proliferation of laptop personal computers, which inherently possess much of the necessary system components for supporting various broadcast services, the merger of mobile computing and interactive as well as distributive or broadcast-type wireless multimedia services is imminent. Against this background this thesis endeavours to innovate in the field of mobile video broadcast enabling technologies.

The organisation of this thesis is presented next.

1.5 Organisation of Thesis

We present the layout of this thesis below:

- Chapter 2 presents the results of our attempts to characterise the error sensitivity of the MPEG-2 video bitstream parameters to transmission errors. We have also provided a performance study of the DVB-Terrestrial (DVB-T) system in comparison to a turbo-coded DVB-like scheme in the context of both hierarchical and non-hierarchical transmissions. Furthermore, we have investigated the employment of data partitioning techniques in an effort to increase the resilience of the MPEG-2 video codec. However, the results of our experiments indicated that the hierarchical scheme was outperformed by the non-hierarchical arrangement.
- Chapter 3 presents results of the simulated system performance of the DVB-Satellite (DVB-S) system in comparison to an enhanced turbo-coded DVB-like scheme.
- In Chapter 4, we investigated the complexity versus performance trade-offs of transmission systems employing Trellis Coded Modulation (TCM), Bit-Interleaved Coded Modulation (BICM), Iterative Decoded Bit-Interleaved Coded Modulation (BICM-ID) and Turbo Trellis Coded Modulation (TTCM) schemes in conjunction with the DVB-T scheme.
- Chapter 5 presents an overview of the MPEG-4 standard. We provide a brief account of each of the components that constitute the MPEG-4 standard. A more detailed description of the MPEG-4 visual coding component is then provided. We have also characterised MPEG-4 error resilience features and their performance within the context of the MPEG-4 DVB-T transmission arrangement. In this chapter, an alternative scheme was also presented, where a packetizer was used to encapsulate the MPEG-4 video bitstream. The philosophy of the scheme is to minimise the loss of coded video data in the event of a transmission packet loss and also to provide the MPEG-4 video decoder with a video bitstream, which is syntactically correct.

The novel contributions of this thesis are as follows:

- The characterisation of the error sensitivity of the MPEG-2 video bitstream parameters in the event of transmission errors [157].
- The employment of turbo codes to enhance the performance of the DVB-T and DVB-S systems [157, 158].
- Investigation of the complexity versus performance trade-offs of trellis coded modulation, bit-interleaved coded modulation, iterative decoded bit-interleaved coded modulation and turbo trellis coded modulation based systems in the context of a DVB-T system [159].
- The introduction of a packetisation algorithm for the MPEG-4 video bitstream together with a so-called intra-coded macroblock update scheme, in order to increase the error resilience of the MPEG-4 video codec.

1.6 Summary

We have provided a brief account of the motivation behind the development of the MPEG-2 and MPEG-4 video coding standards in this introductory chapter. We have also presented an overview of block-based video compression techniques and their associated problems as well as their mitigation in the context of wireless video communications. There are other types of compression schemes, which have not been discussed here, notably vector quantisation techniques [42, 160–162], wavelet compression [163–167] and quad-tree based video compression schemes [42, 79, 168, 169]. Chapter 2 will present the results of our attempts to enhance the Pan-European DVB system for mobile transmission environments.

Chapter 2

OFDM-based Turbo-coded Hierarchical and Non-hierarchical Terrestrial Digital Video Broadcasting

2.1 Background and Motivation

Following the standardization of the Pan-European Digital Video Broadcasting (DVB) systems, we have begun to witness the arrival of digital television services to the home. However, for a high proportion of business and leisure travellers it is desirable to have access to DVB services while on the move. Although it is feasible to access these services with the aid of dedicated DVB receivers, these receivers may also find their way into the laptop computers of the near future. These intelligent laptops may also become the portable DVB receivers of wireless-in-home networks.

In recent years three DVB standards have emerged in Europe for terrestrial [13], cable-based [16] and satellite-oriented [12] delivery of DVB signals. The more hostile propagation environment of the terrestrial system requires concatenated Reed-Solomon [23, 170] (RS) and Rate Compatible Punctured Convolutional Coding [23,170] (RCPCC) combined with Orthogonal Frequency Division Multiplexing (OFDM) based modulation [100]. By contrast, the more benign cable- and satellite-based media facilitate the employment of multi-level modems using upto 256-level Quadrature Amplitude Modulation (QAM) [100]. These schemes are capable of delivering high-definition video at bitrates of up to 20 Mbits/s in stationary broadcast-mode distributive wireless scenarios.

Recently, there has been a range of DVB system performance studies in the literature [171–174]. Against this background, in this contribution we have proposed turbo-coding based improvements to the terrestrial DVB system [13] and investigated its performance under hostile mobile channel conditions. We have also studied various video bitstream partitioning and channel coding schemes both in the so-called hierarchical and non-hierarchical transceiver modes to be discussed during our further discourse and compared their performance.

The rest of the chapter is divided into the following sections. In Section 2.2 the bit error sensitivity of the MPEG-2 coding parameters [67] is characterised. A brief overview of the enhanced turbo-coded and standard DVB terrestrial scheme is presented in Section 2.3, while the channel model is described in Section 2.4. Following this, in Section 2.5 the reader is introduced to the MPEG-2 data partitioning scheme [4] used to split the input MPEG-2 video bitstream into two error protection classes, which can then be protected either equally or unequally. These two video bit protection classes can then be broadcasted to the receivers using the so-called DVB terrestrial hierarchical transmission format [13]. The performance of the data partitioning scheme was investigated by corrupting either the high or low sensitivity video bits using randomly distributed errors for a range of system configurations in Section 2.6 and their effects on the overall reconstructed video quality were evaluated. Following this, the performance of the improved DVB terrestrial system employing the so-called non-hierarchical and hierarchical format [13] is examined in a mobile environment in Sections 2.7 and 2.8, before our conclusions and future work areas are presented in Section 2.9. We note furthermore that readers mainly interested in the overall system performance may opt for directly proceeding to Section 2.3. Let us commence our discourse in the next section by describing an objective method of quantifying the sensitivity of the MPEG-2 video parameters.

2.2 MPEG-2 Bit Error Sensitivity

At this stage we have to note again that a number of different techniques can be used in order to quantify the bit error sensitivity of the MPEG-2 bits. The outcome of these investigations will depend to a degree on the video material used, the output bitrate of the video codec, the objective video quality measures used and the averaging algorithm employed. Perceptually motivated, subjective quality based sensitivity testing becomes simply infeasible due to the large number of associated test scenarios. Hence in this section a simplified objective video quality measure based bit-sensitivity evaluation procedure is proposed, which attempts to take into account all the major factors influencing the sensitivity of MPEG-2 bits. Specifically, the proposed procedure takes into account the position and the relative frequency of the MPEG-2 parameters in the bitstream, the number of the associated coding bits for each MPEG-2 parameter, the video bitrate and the effect of loss of synchronisation or error propagation due to corrupted bits. Nonetheless, we note that a range of similar bit sensitivity estimation techniques exhibiting different strengths and weaknesses can be devised and no doubt future research will produce a variety of similarly motivated techniques.

In this section we assume familiarity with the MPEG-2 standard [4, 67]. The aim of our MPEG-2 error resilience study was to quantify the average PSNR degradation inflicted by each erroneously decoded video codec parameter in the bitstream, so that appropriate protection can be assigned to each parameter. First, we will define three measures, namely the Peak Signal-to-Noise Ratio (PSNR), the PSNR degradation and the average PSNR degradation, which are to be used in our subsequent discussions. The

PSNR is defined as follows:

$$\text{PSNR} = 10\log_{10}\frac{\sum_{n=0}^{N}\sum_{m=0}^{M}255^2}{\sum_{n=0}^{N}\sum_{m=0}^{M}\Delta_{m,n}^2}, \tag{2.1}$$

where Δ is the difference between the uncoded pixel value and the reconstructed pixel value, while the variables M and N refer to the dimension of the image. The maximum possible 8-bit pixel luminance value of 255 was used in Equation 2.1 in order to mitigate the PSNR's dependence on the video material used. The PSNR degradation is the difference between the PSNR of the decoder's reconstructed image in the event of erroneous decoding and successful decoding. The average PSNR degradation is then the mean of the PSNR degradation values computed for all the image frames of the video test sequence.

Most MPEG-2 parameters are encoded by several bits and they may occur in different positions in the video sequence. In these different positions they typically affect the video quality differently, since corrupting a specific parameter of a frame close to the commencement of a new picture start code inflicts a lesser degradation, than corrupting an equivalent parameter further from the resynchronisation point. Hence the sensitivity of the MPEG-2 parameters is position-dependent. Furthermore, different encoded bits of the same specific MPEG-2 parameter may exhibit different sensitivity to channel errors. Figure 2.1 shows such an example for the parameter known as intra_dc_precision [67], which is coded under the so-called Picture Coding Extension [4]. In this example, the PSNR degradation profiles due to bit errors being inflicted on the parameter intra_dc_precision of Frame 28 showed that the degradation is dependent on the significance of the bit considered. Specifically, errors in the most significant bit (MSB) caused an approximately 3 dB higher PSNR degradation, than the least significant bit (LSB) errors. Furthermore, the PSNR degradation due to a MSB error of the intra_dc_precision parameter in Frame 73 is similar to the PSNR degradation profile for the MSB of the intra_dc_precision parameter of Frame 28. Due to the variation of the PSNR degradation profile for the bits of different significance of a particular parameter, as well as for the same parameter at its different occurrences in the bitstream, it is necessary to determine the *average* PSNR degradation for each parameter in the MPEG-2 bitstream.

Our approach in obtaining the average PSNR degradation was similar to that suggested in References [175] and [176]. Specifically, the average measure used here takes into account the significance of the bits corresponding to the MPEG-2 parameter concerned, as well as the occurrence of the same parameter at different locations in the encoded video bitstream. In order to find the average PSNR degradation for each MPEG-2 bitstream parameter, the different bits encoding a specific parameter, as well as the bits of the same parameter but occuring at different locations in the MPEG-2 bitstream were corrupted and the associated PSNR degradation profile versus frame index was registered. The observed PSNR degradation profile generated for different locations of a specific parameter was then used to compute the average PSNR degradation. As an example, we shall use the PSNR degradation profile shown in Figure 2.1. In this figure there are three degradation profiles. The average PSNR degradation for each profile is first computed in order

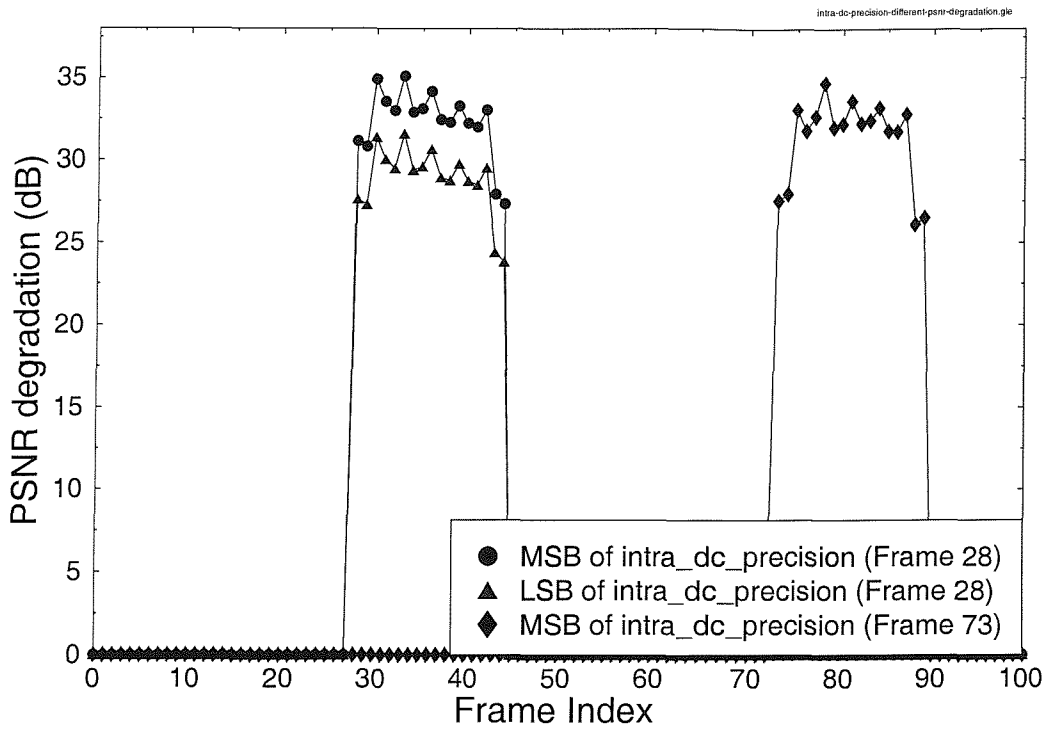
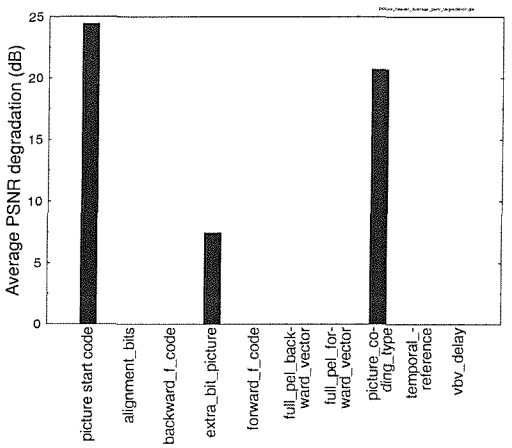


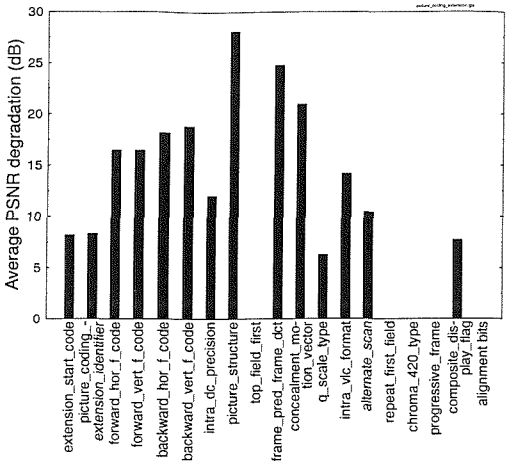
Figure 2.1: PSNR degradation profile for the different bits used to encode the so-called intra_dc_precision parameter [67] in different corrupted video frames for the "Miss America" QCIF video sequence encoded at 30 frame/s and 1.15 Mbit/s.

to produce three average PSNR degradation values corresponding to the three respective profiles. The mean of these three PSNR averages will then form the final average PSNR degradation for the intra_dc_precision parameter. The same process is repeated for all MPEG-2 parameters from the Picture Layer up to the Block Layer. The difference with respect to the approach adopted in [175, 176] was that while in [175, 176] the average PSNR degradation was acquired for each bit of the output bitstream, we have adopted a simpler approach in this contribution due to the large number of different parameters within the MPEG-2 bitstream. Figure 2.2 shows the typical average PSNR degradations of the various MPEG-2 parameters of the Picture Header Information, Picture Coding Extension, Slice Layer, Macroblock Layer and Block Layer [4], which were obtained using the 176×144 Quarter Common Intermediate Format (QCIF) "Miss America" (MA) video sequence at 30 frames/s and a high average bitrate of 1.15 Mbits/s.

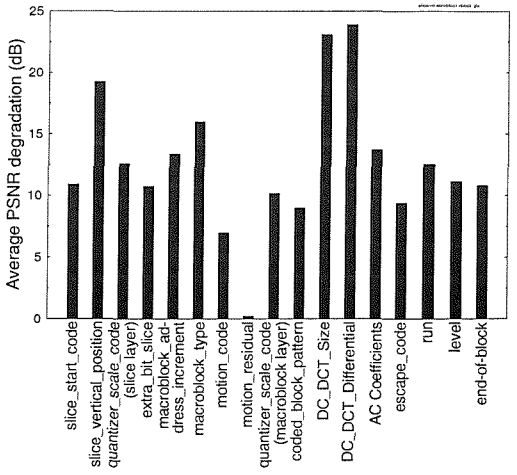
However, the different MPEG-2 parameters or codewords occur with different probabilities and they are allocated different numbers of bits. Therefore, the average PSNR degradation registered in Figure 2.2 for each MPEG-2 parameter was multiplied with the long-term probability of this MPEG-2 parameter occuring in the MPEG-2 bitstream and with the relative probability of bits being allocated to that MPEG-2 parameter. Figure 2.3 and Figure 2.4 show the probability of occcurence of the various MPEG-2



(a) Picture Header Information



(b) Picture Coding Extension

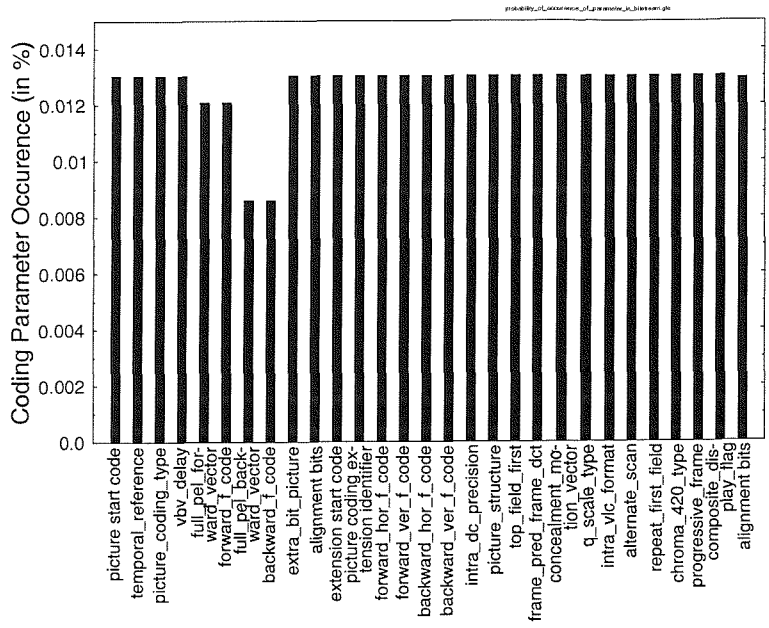


(c) Slice, Macroblock and Block Layers

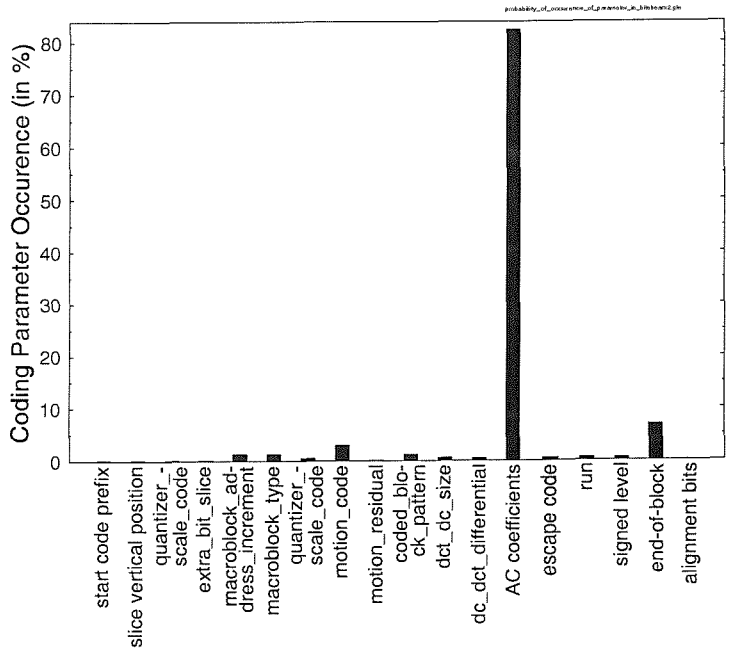
Figure 2.2: Average PSNR degradation for the various MPEG-2 parameters in (a) Picture Header Information (b) Picture Coding Extension and (c) Slice, Macroblock and Block Layers for the "Miss America" QCIF video sequence encoded at 30 frame/s and 1.15 Mbit/s.

parameters characterised in Figure 2.2 and the probability of bits allocated to the parameters in the Picture Header Information, Picture Coding Extension, as well as in the Slice, Macroblock and Block Layers [4], respectively, for the QCIF MA video sequence encoded at 1.15 Mbit/s.

We shall concentrate first on Figure 2.3(a). It is observed that all parameters – except for the full_pel_forward_vector, forward_f_code, full_pel_backward_vector and backward_f_code – have the same probability of occurrence, since they appear once for every coded video frame. The parameters full_pel_forward_vector and forward_f_code have a higher probability of occurrence than full_pel_backward_vector and backward_f_code, since the former two appear in both P-frames and B-frames, while the latter two only occur



(a) Picture Header Information and Picture Coding Extension



(b) Slice, Macroblock and Block Layers

Figure 2.3: Probability of occurrence for the various MPEG-2 parameters characterised in Figure 2.2 (a) Picture Header Information and Picture Coding Extension (b) Slice, Macroblock and Block Layers for the "Miss America" QCIF video sequence encoded at 30 frame/s and 1.15 Mbit/s.

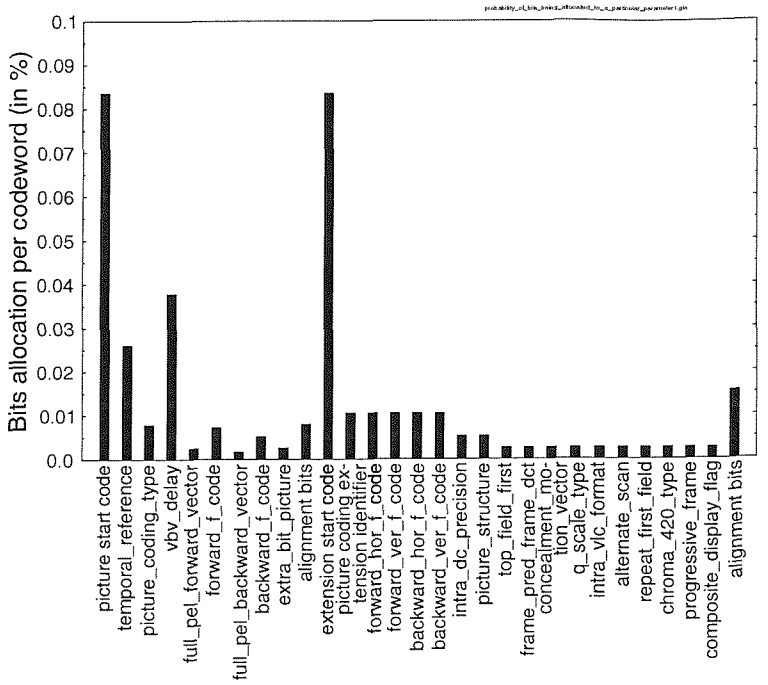
in B-frames. For our experiments, the MPEG-2 encoder was configured such that for every encoded P-frame, there were two encoded B-frames. However, when compared with the parameters from the Slice Layer, Macroblock Layer and Block Layer, which are characterised by the bar chart of Figure 2.3(b), the parameters of the Picture Header Information and Picture Coding Extension appeared significantly less frequently.

If we compare the frequency of occurrence of the parameters in the Slice Layer with those in the Macroblock and Block Layers, the former appeared less often, since there were 11 macroblocks and 44 blocks per video frame slice for the QCIF Miss America video sequence considered in our experiments. The AC Discrete Cosine Transform (DCT) [62] coefficient parameter had the highest probability of occurrence, exceeding eighty percent.

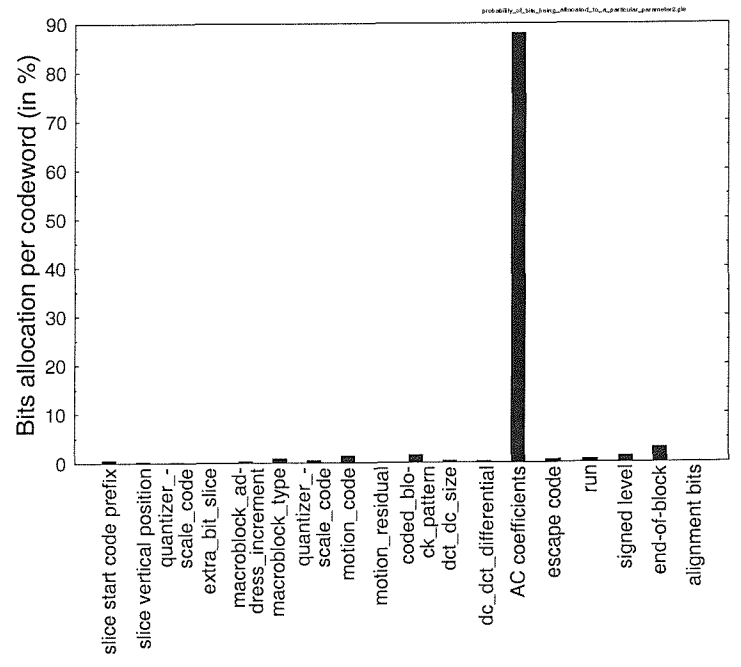
Figure 2.4 shows the probability of bits being allocated to the various MPEG-2 parameters in the Picture Header Information, Picture Coding Extension, Slice, Macroblock and Block Layers [4]. Figure 2.5 was included to more explicitly illustrate the probability of bit allocation seen in Figure 2.4(b), with the probability of allocation of bits to the AC DCT coefficients being omitted from the bar-chart. Considering Figure 2.4(a), the two dominant parameters, with the highest number of encoding bits requirement, are the Picture Start Code (PSC) and the Picture Coding Extension Start Code (PCESC). However, comparing these probabilities with the probability of bits being allocated to the various parameters in the Slice, Macroblock and Block Layers, the percentage of bits allocated can still be considered minimal due to their infrequent occurrence. In the Block Layer, the AC DCT coefficients require in excess of 85 percent of the bits available for the whole video sequence. However, at bitrates lower than 1.15 Mbit/s the proportion of AC-coefficient encoding bits was significantly reduced, as illustrated by Figure 2.6. Specifically, at 30 frames/s and 1.15 Mbit/s, the average number of bits per video frame is about 38 000 and a given proportion of these bits is allocated to the MPEG-2 control header information, motion information and to the DCT coefficients. Upon reducing the total bitrate budget – since the number of control header bits is more or less independent of the target bitrate – the proportion of bits allocated to the DCT coefficients is substantially reduced. This is explicitly demonstrated in Figure 2.6 for bitrates of 1.15 Mbit/s and 240 kbit/s for the "Miss America" QCIF video sequence.

The next process, as discussed earlier, was to normalise the measured average PSNR degradation according to the probability of occurrence of the respective MPEG-2 parameters in the bitstream and the probability of bits being allocated to this parameter. The normalised average PSNR degradation caused by corrupting the parameters of the Picture Header Information and Picture Coding Extension [4] is portrayed in Figure 2.7(a). Similarly, the normalised average PSNR degradation for the parameters of the Slice, Macroblock and Block Layers is shown in Figure 2.7(b). In order to visually enhance Figure 2.7(b), the normalised average PSNR degradation for the AC DCT coefficients was omitted in the bar-chart shown in Figure 2.8.

The highest PSNR degradation was inflicted by the AC DCT coefficients, since these parameters occur most frequently and hence are allocated the highest number of bits. When a bit error occurs in the bitstream, the AC DCT coefficients have a high probability



(a) Picture Header Information and Picture Coding Extension



(b) Slice, Macroblock and Block Layers

Figure 2.4: Probability of bits being allocated to parameters in (a) Picture Header Information and Picture Coding Extension (b) Slice, Macroblock and Block Layers for the "Miss America" QCIF video sequence encoded at 30 frame/s and 1.15 Mbit/s.

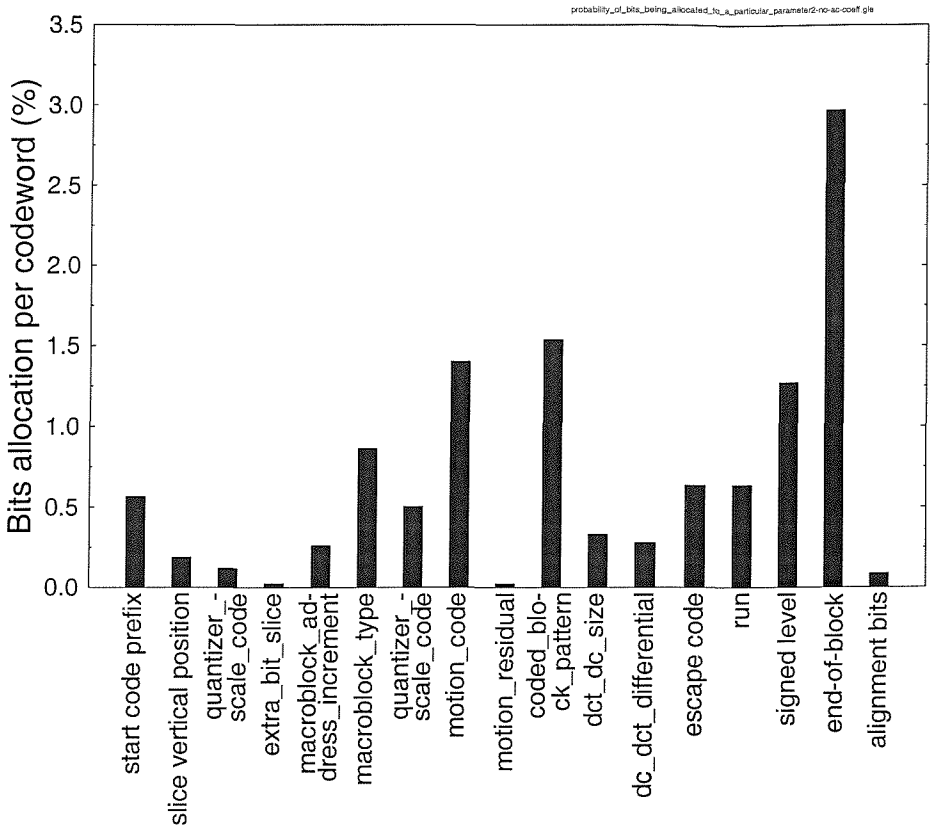


Figure 2.5: Probability of bits being allocated to the various MPEG-2 Slice, Macroblock and Block Layer parameters, as seen in Figure 2.4(b), where the probability of bits allocated to the AC DCT coefficients was omitted, in order to show the allocation of bits to the other parameters more clearly. This probability of bits allocation to the various MPEG-2 parameters is associated with the "Miss America" QCIF video sequence encoded at 30 frame/s and 1.15 Mbit/s.

of being corrupted. The other parameters, such as the DC_DCT_size and DC_DCT_differential, though exhibiting high average PSNR degradations when corrupted, registered low normalised average PSNR degradations since their occurence in the bitstream is confined to the infrequent intra-coded frames.

The end-of-block MPEG-2 parameter exhibited the second highest normalised average PSNR degradation in this study. Although the average number of bits used for the end-of-block is only approximately 2.17 bits, the probability of occurrence and the probability of bits being allocated to it is higher than those of other parameters, with the exception of the AC DCT coefficients. Furthermore, in general, the parameters of the Slice, Macroblock and Block Layers exhibit higher average normalised PSNR degradations due to their more frequent occurrence in the bitstream compared to the parameters, which belong to the Picture Header Information and to the Picture Coding Extension. This also implies that the percentage of bits allocated to these parameters is higher.

Comparing the normalised average PSNR degradations of the parameters in the Picture Header Information and Picture Coding Extension, the PSC exhibits the highest normalised average PSNR degradation. Although most of the parameters here occur

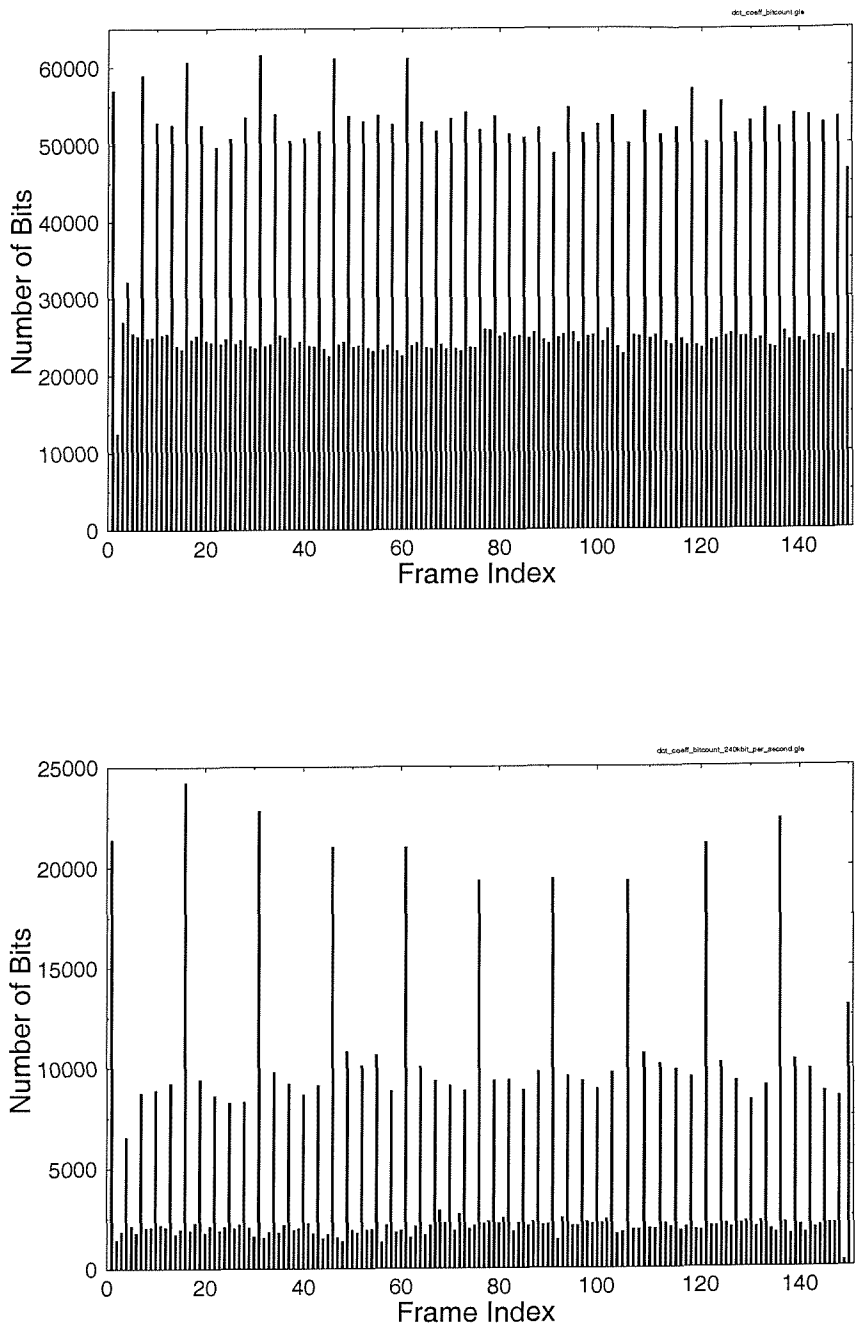
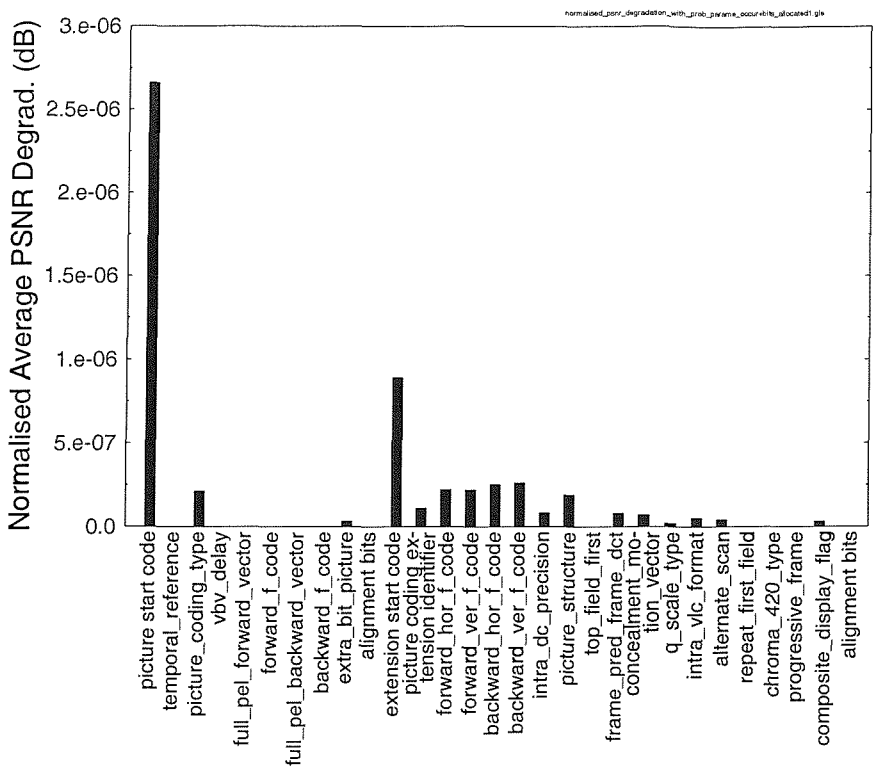
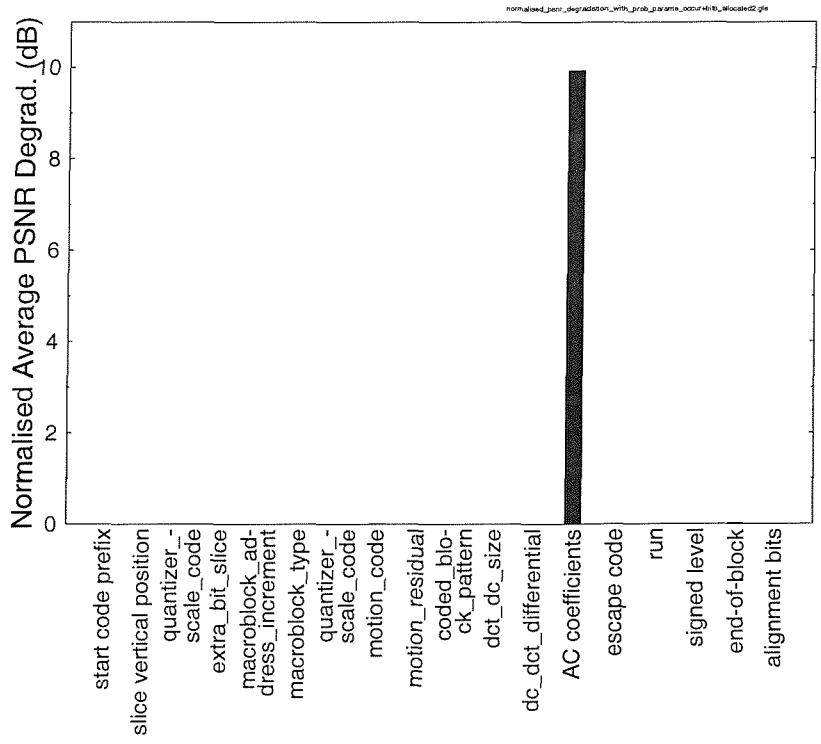


Figure 2.6: Profile of bits allocated to the DCT coefficients, when the 30 frame/s QCIF Miss America video sequence is coded at (a) 1.15 Mbit/s (top) and (b) 240 kbit/s (bottom). The sequence of frames is in the order I B B, P B B, P B B, P B B and so on.



(a) Picture Header Information and Picture Coding Extension



(b) Slice, Macroblock and Block Layers

Figure 2.7: Normalised average PSNR degradation for the various parameters in (a) Picture Header Information and Picture Coding Extension (b) Slice, Macroblock and Block Layers, normalised to the probability of occurrence of the respective parameters in the bitstream and the probability of bits being allocated to the parameter for the "Miss America" QCIF video sequence encoded at 30 frame/s and 1.15 Mbit/s.

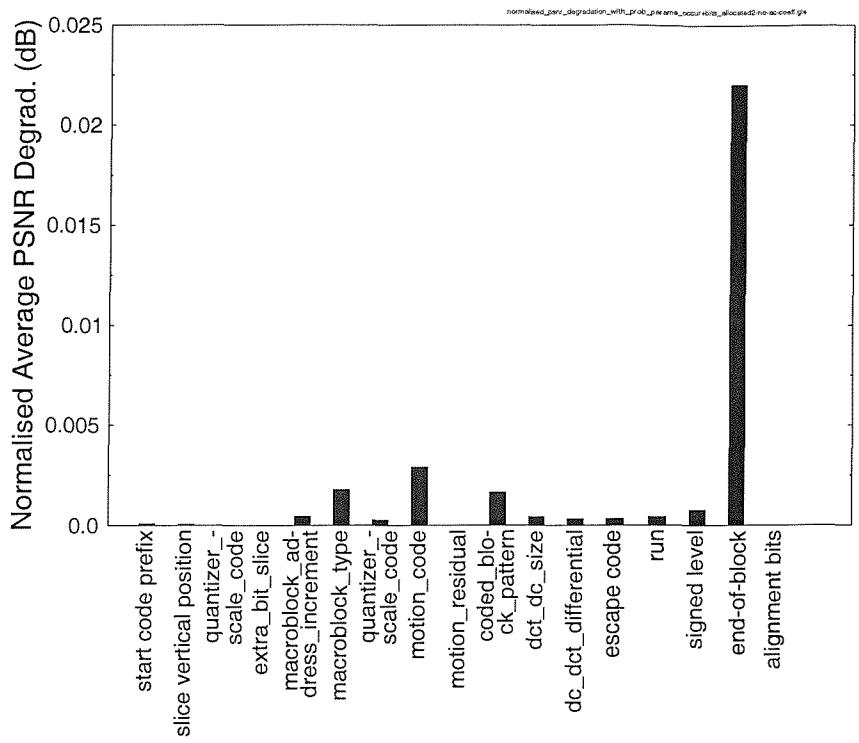


Figure 2.8: This bar chart is the same as Figure 2.7(b), although the normalised average PSNR degradation for the AC DCT coefficients was omitted in order to show the average PSNR degradation of the other parameters. This bar chart is presented for the "Miss America" QCIF video sequence encoded at 30 frame/s and 1.15 Mbit/s case.

with equal probability as seen in Figure 2.3(a), the picture start code requires a higher portion of the bits compared to the other parameters, with the exception of the extension start code. Despite having the same probability of occurrence and the same allocation of bits, the extension start code exhibits a lower normalised PSNR degradation than the picture start code, since its average un-normalised degradation is lower, as shown in Figure 2.2.

From Figures 2.7 and 2.8, we observe that the video PSNR degradation was dominated by the erroneous decoding of the AC DCT coefficients, which appeared in the MPEG-2 video bitstream in the form of variable-length codewords. This suggests invoking unequal error protection techniques for protecting the MPEG-2 parameters during transmission. In a low complexity implementation, two protection classes may be envisaged. The higher priority class would contain all the important header information and some of the more important low-frequency variable-length coded DCT coefficients. The lower priority class would then contain the remaining less important, higher frequency variable-length coded DCT coefficients. This partitioning process will be detailed in Section 2.5 together with its associated performance in the context of the hierarchical DVB [13] transmission scheme in Section 2.8. Let us, however, first consider the architecture of the investigated DVB system in the next section.

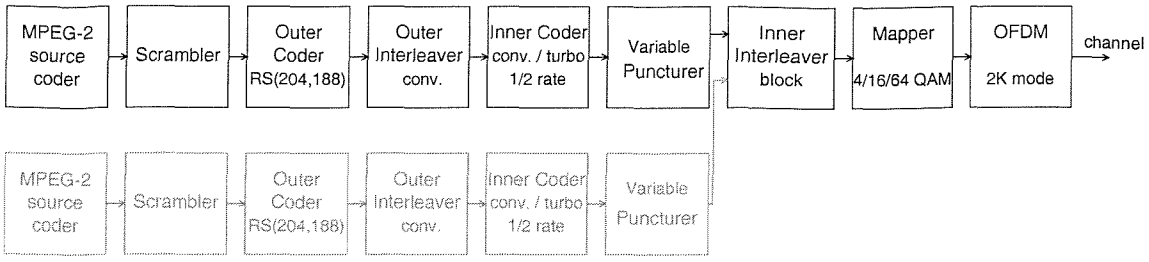


Figure 2.9: Schematic of the DVB terrestrial transmitter functions.

2.3 DVB Terrestrial Scheme

The block diagram of the DVB terrestrial (DVB-T) transmitter [13] is shown in Figure 2.9, which consists of an MPEG-2 video encoder, channel coding modules and an Orthogonal Frequency Division Multiplex (OFDM) modem [100, 177]. The bitstream generated by the MPEG-2 encoder is packetised into frames of 188-byte long. The video data in each packet is then randomized by the scrambler of Figure 2.9. The specific details concerning the scrambler have not been included in this paper, since these may be obtained from the DVB-T standard [13].

Due to the poor error resilience of the MPEG-2 video codec, powerful concatenated channel coding is employed. The concatenated channel codec of Figure 2.9 comprises a shortened Reed-Solomon (RS) outer code and an inner convolutional encoder. The 188-byte MPEG-2 video packet is extended by the Reed-Solomon encoder [23, 170] with parity information to facilitate error recovery in order to form a 204-byte packet. The Reed-Solomon decoder can then correct up to eight erroneous bytes for each 204-byte packet. Following this, the RS-coded packet is interleaved by a convolutional interleaver and further protected by a half-rate inner convolutional encoder using a constraint length of 7 [23, 170].

Furthermore, the overall code rate of the concatenated coding scheme can be adapted by variable puncturing which supports code rates of 1/2 (no puncturing) as well as 2/3, 3/4, 5/6 and 7/8. The parameters of the convolutional encoder are summarised in Table 2.1.

Convolutional Coder Parameters	
Code Rate	1/2
Constraint Length	7
n	2
k	1
Generator Polynomials (octal format)	171, 133

Table 2.1: Parameters of the CC(n, k, K) convolutional inner encoder of the DVB-T modem.

If only one of the two branches of the transmitter in Figure 2.9 is utilised, the DVB-T modem is said to be operating in its non-hierarchical mode. In this mode, the modem can have a choice of QPSK, 16-QAM or 64-QAM modulation constellations [100].

A second video bitstream can also be multiplexed with the first one by the inner interleaver, when the DVB modem is in its so-called hierarchical mode [13]. The choice of modulation constellations in this mode is between 16-QAM and 64-QAM. We shall be employing this transmission mode, when the so-called data partitioning scheme, of Section 2.5, is used to split the incoming MPEG-2 video bitstream into two video bit-protection classes with one class having a higher grade of protection or priority than the other one. The higher priority video bits will be mapped to the MSBs of the modulation constellation points and the lower priority video bits to the LSBs of the QAM-constellation [100]. For 16-QAM and 64-QAM, the two MSBs of each 4- or 6-bit QAM symbol will contain the more important video data. The lower priority video bits will then be mapped to the lower significance 2 bits and 4 bits of 16-QAM and 64-QAM, respectively [100].

These QPSK, 16-QAM or 64-QAM symbols are then distributed over the OFDM carriers [100]. The parameters of the OFDM system are presented in Table 2.2.

OFDM Parameters	
Total number of subcarriers	2048 (2K mode)
Number of effective subcarriers	1705
OFDM symbol duration T_s	224 μs
Guard Interval	$T_s/4 = 56\mu$s
Total symbol duration (inc. Guard Interval)	280 μs
Consecutive subcarrier spacing $1/T_s$	4464 Hz
DVB channel spacing	7.61 MHz
QPSK and QAM symbol period	7/64 μs

Table 2.2: Parameters of the OFDM module used in the DVB-T modem [13].

Beside implementing the standard DVB-T system as a benchmark, we have improved the system by replacing the convolutional coder by a turbo codec [156, 178]. The turbo codec's parameters used in our investigations are displayed in Table 2.3. The block diagram of the turbo encoder is shown in Figure 2.10. The turbo encoder is constructed of two component encoders. Each component encoder is a half-rate convolutional encoder, whose parameters are listed in Table 2.3. The two component encoders are used to encode the same input bits, although the input bits of the second component encoder are interleaved before encoding. The output bits of the two component codes are punctured and multiplexed, in order to form a single output bitstream. The component encoder used here is known as a half-rate Recursive Systematic Convolutional (RSC) encoder [179]. It generates one parity bit and one systematic output bit for every input bit. In order to provide an overall coding rate of $R = 1/2$, half the output bits from the two encoders must be punctured. The puncturing arrangement used in our work is to transmit all the systematic bits from the first encoder and every other parity bit from both encoders [180]. We note here that one iteration of the turbo decoder involves two so-called Logarithmic Maximum A-Posteriori (LogMAP) [181] decoding operations, which we repeated for the 8 iterations. Hence, the total turbo decoding complexity is about 16 times higher than a

constraint length $K = 3$ constituent convolutional decoding. Therefore the turbo decoder exhibits a similar complexity to the $K = 7$ convolutional decoder.

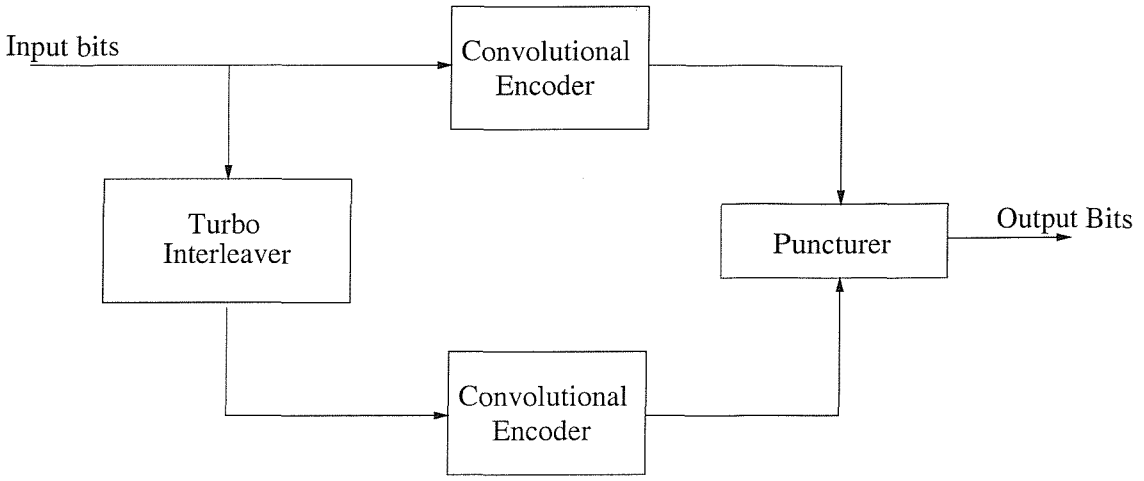


Figure 2.10: Block diagram of turbo encoder.

Turbo Coder Parameters	
Turbo Code Rate	1/2
Input Block Length	17952 bits
Interleaver Type	Random
Number of Turbo Decoder Iterations	8
Turbo Encoder Component Code Parameters	
Component Code Encoder Type	Recursive Systematic Convolutional (RSC)
Component Code Decoder Type	Log-MAP [181]
Constraint Length	3
n	2
k	1
Generator Polynomials (octal format)	7, 5

Table 2.3: Parameters of the inner turbo encoder used to replace the DVB-S system's $K = 7$ convolutional encoder of Table 2.1 (RSC: recursive systematic code).

In this section, we have given an overview of the standard and enhanced DVB-T system, which we have used in our experiments. Readers interested in further details of the DVB-T system are referred to the DVB-T standard [13]. The performance of the standard DVB-T system and the turbo coded system is characterised in Section 2.7 and 2.8 for non-hierarchical and hierarchical transmissions, respectively. Let us now briefly consider the multipath channel model used in our investigations.

2.4 Channel Model

The channel model employed in this study was the twelve-path COST 207 [182] Hilly Terrain (HT) type impulse response, with a maximal relative path delay of 19.9 μs. This strongly dispersive channel was selected in order to provide a scenario, which can be

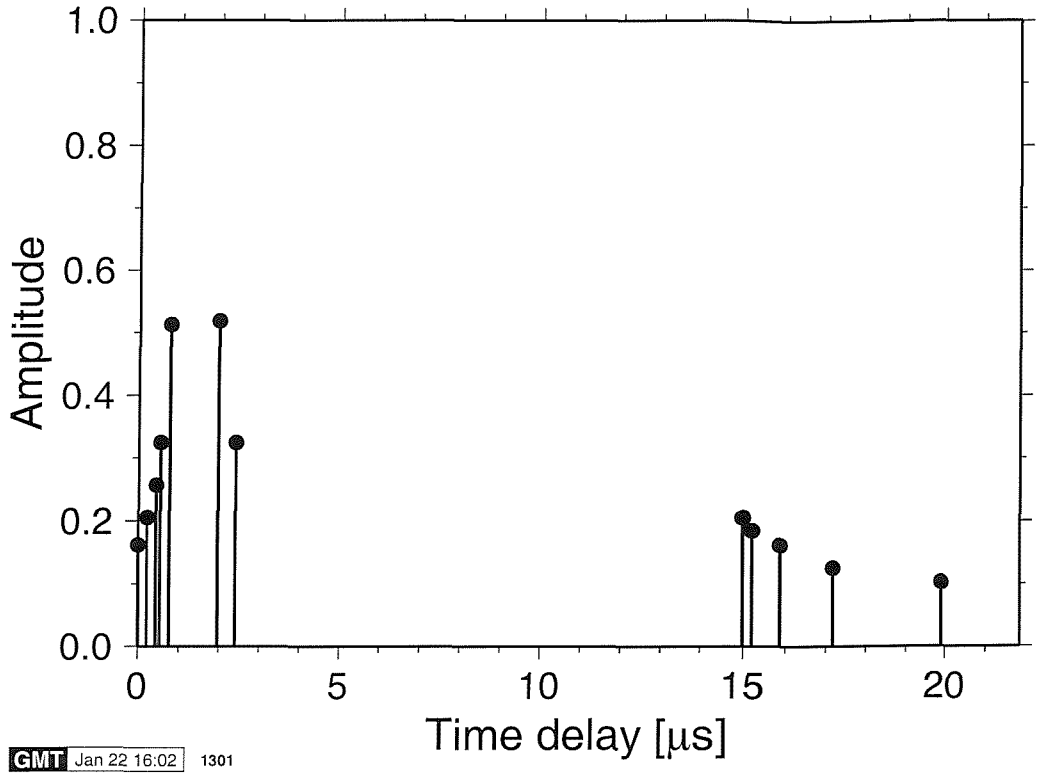


Figure 2.11: COST 207 Hilly Terrain (HT) type impulse response.

viewed as the worst-case propagation scenario, again, associated with propagation excess delays up to 19.9 μs. However, if the number of OFDM subcarriers is sufficiently high, the individual OFDM subchannels can be viewed as a multiplicity of parallel narrowband channels.

In the system characterised here, we have used a carrier frequency of 500MHz and a sampling rate of 7/64μs. Each of the channel paths was faded independently, obeying a Rayleigh fading distribution, according to a normalised Doppler frequency of 10^{-5} [23]. This corresponds to a worst-case vehicular velocity of about 200 km/h. The unfaded impulse response is depicted in Figure 2.11. For the sake of completeness we note that the standard COST 207 channel model was defined in order to facilitate the comparison of different GSM implementations [23] under identical conditions. The associated bitrate was 271 kbit/s, while in our investigations the bitrate of DVB-quality transmissions can be as high as 20 Mbit/s, where there is a higher number of resolvable multipath components within the dispersion-range considered. However, the performance of various wireless tranceivers is well understood by the research community over this standard COST 207 channel and hence its employment is beneficial in benchmarking terms. Furthermore, since the OFDM modem has 2048 subcarriers, the subcarrier signalling rate is effectively 2000-times lower than our maximum DVB-rate of 20 Mbit/s, corresponding to 10 kbit/s. At this subchannel rate, the individual subchannels can be considered nearly frequency-flat. In summary, in conjunction with the 200 km/h vehicular speed used the investigated channel conditions constitute a pessimistic scenario.

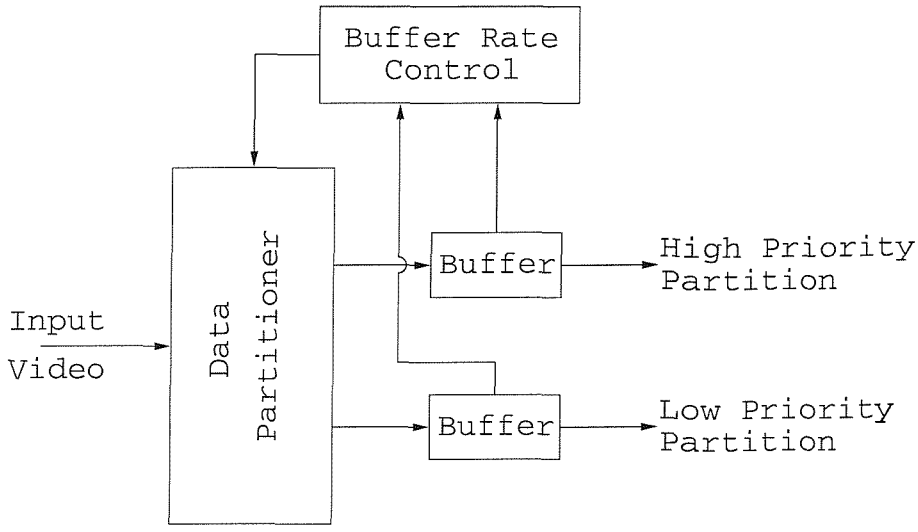


Figure 2.12: Block diagram of the data partitioner and rate controller.

In order to facilitate unequal error protection, the data partitioning procedure of the MPEG-2 video bitstream is considered next.

2.5 Data Partitioning Scheme

Efficient bitstream partitioning schemes for H.263-coded video were proposed for example by Gharavi and Alamouti [183], which were evaluated in the context of the third-generation mobile radio standard proposal known as IMT2000 [23]. As portrayed in Figures 2.7 and 2.8, the corrupted variable-length coded DCT coefficients inflict a high video PSNR degradation. Assuming that all MPEG-2 header information is received correctly, the fidelity of the reconstructed images at the receiver is dependent on the number of correctly decoded DCT coefficients. However, the subjective effects of the loss of higher spatial frequency DCT coefficients are less dramatic compared to that of the lower spatial frequency DCT coefficients. The splitting of the MPEG-2 video bitstream into two different integrity bitstreams is referred to as data partitioning [4]. Recall from Section 2.3 that the hierarchical 16- and 64-QAM DVB-T transmission scheme enables us to multiplex two unequally protected MPEG-2 video bitstreams for transmission. This section describes the details of the MPEG-2 data partitioning scheme [4].

Figure 2.12 shows the block diagram of the data partitioning scheme, which splits an MPEG-2 video bitstream into two resultant bitstreams. The position at which the MPEG-2 bitstream is split is based on a variable referred to as the Priority Breakpoint (PBP) [4]. The PBP can be adjusted at the beginning of the encoding of every MPEG-2 image slice, based on the buffer "occupancy" or "fullness" of the two output buffers. For example, if the high-priority buffer is 80 % full and the low-priority buffer is only 40 % full, the rate control module would have to adjust the PBP such that more data is directed to the low-priority partition. This measure is taken to avoid high-priority buffer overflow and low-priority buffer underflow events. The valid values for the MPEG-2 PBP are summarised in Table 2.4 [4].

PBP	Syntax elements in high-priority partition
0	Low-priority partition always has its PBP set to 0.
1	Sequence, GOP, Picture and Slice layer information upto extra bit slice.
2	Same as above and upto macroblock address increment.
3	Same as above plus including macroblock syntax elements, but excluding coded block pattern.
4 ... 63	Reserved for future use.
64	Same as above plus including DC DCT coefficient and the first runlength coded DCT coefficient.
65	Same as above and up to the second runlength coded DCT coefficient.
$64 + x$	Same as above and up to x runlength coded DCT coefficient.
127	Same as above and up to 64 runlength coded DCT coefficient.

Table 2.4: Priority breakpoint values and the associated MPEG-2 parameters that will be directed to the high-priority partition [4]. A higher PBP directs more parameters to the high-priority partition. By contrast, for the low-priority partition a higher PBP implies obtaining less data.

There are two main stages in updating the PBP. The first stage involves the rate control module of Figure 2.12 in order to decide on the preferred new PBP value for each partition based on its individual buffer occupancy and on the current value of the PBP. The second stage then combines the two desired PBPs based on the buffer occupancy of both buffers in order to produce a new PBP.

The updating of the PBP in the first stage of the rate control module is based on a heuristic approach, similar to that suggested by Aravind *et al.* [124]. The update procedure is detailed in Algorithm 1, which is discussed below and augmented by a numerical example at the end of this section.

The variable 'sign' is used in Algorithm 1, in order to indicate how the PBP has to be adjusted in the high- and low-priority MPEG-2 partitions, so as to arrive at the required target buffer occupancy. More explicitly, the variable 'sign' in Algorithm 1 is necessary, because the MPEG-2 PBP values [4] shown in Table 2.4 indicate the amount of information which should be directed to the high-priority partition. Therefore, if the low-priority partition requires more data, then the new PBP must be lower than the current PBP. By contrast, for the high-priority partition a higher PBP implies obtaining more data.

Once the desired PBPs for both partitions have been acquired with the aid of Algorithm 1, Algorithm 2 is invoked to compute the final PBP for the current MPEG-2 image slice. The inner working of these algorithms will be augmented by a numerical example at the end of this section. There are two main cases to consider in Algorithm 2. The first one occurs, when both partitions have a buffer occupancy of less than 50%. By using the reciprocal of the buffer occupancy in Algorithm 2 as a weighting factor during the computation of the PBP adjustment value "delta", the algorithm will favour the new PBP decision of the less occupied buffer in order to fill the buffer with more data in the current image slice. This is simply because the buffer is closer to underflow and hence increasing the PBP according to its instructions will assist in preventing the particular buffer from

Algorithm 1 Computes the desired PBP update for the high- and low-priority partitions which is then passed to Algorithm 2, in order to determine the PBP to be set for the current image slice.

Step 1: Initialize parameters
  if High Priority Partition **then**
    sign := +1
  else
    sign := −1
  end if

Step 2:
  if buffer occupancy ≥ 80% **then**
    diff := 64 − PBP
  end if

  if buffer occupancy ≥ 70% **and** buffer occupancy < 80% **then**
    if PBP ≥ 100 **then**
      diff := −9
    end if
    if PBP ≥ 80 **and** PBP < 100 **then**
      diff := −5
    end if
    if PBP ≥ 64 **and** PBP < 80 **then**
      diff := −2
    end if
  end if

  if buffer occupancy ≥ 50% **and** buffer occupancy < 70% **then**
    diff := +1
  end if

  if buffer occupancy < 50% **then**
    if PBP ≥ 80 **then**
      diff := +1
    end if
    if PBP ≥ 70 **and** PBP < 80 **then**
      diff := +2
    end if
    if PBP ≥ 2 **and** PBP < 70 **then**
      diff := +3
    end if
  end if

Step 3:
  diff := sign × diff
  Return diff

Algorithm 2 Computes the new PBP for the current image slice based on the current buffer occupancy of both partitions

Step 1:

if $\text{Occupancy}_{HighPriority} < 50\%$ **and** $\text{Occupancy}_{LowPriority} < 50\%$
or $\text{Occupancy}_{HighPriority} = 50\%$ **and** $\text{Occupancy}_{LowPriority} < 50\%$
or $\text{Occupancy}_{HighPriority} < 50\%$ **and** $\text{Occupancy}_{LowPriority} = 50\%$
or $\text{Occupancy}_{HighPriority} < 25\%$ **and** $50\% < \text{Occupancy}_{LowPriority} < 70\%$
or $50\% < \text{Occupancy}_{HighPriority} < 70\%$ **and** $\text{Occupancy}_{LowPriority} < 25\%$

then

$$\text{delta} := \frac{\text{Occupancy}_{HighPriority}^{-1} \times \text{diff}_{HighPriority} + \text{Occupancy}_{LowPriority}^{-1} \times \text{diff}_{LowPriority}}{\text{Occupancy}_{HighPriority}^{-1} + \text{Occupancy}_{LowPriority}^{-1}}$$

else

$$\text{delta} := \frac{\text{Occupancy}_{HighPriority} \times \text{diff}_{HighPriority} + \text{Occupancy}_{LowPriority} \times \text{diff}_{LowPriority}}{\text{Occupancy}_{HighPriority} + \text{Occupancy}_{LowPriority}}$$

end if

Step 2:

New_PBP := Previous_PBP + $\lceil$delta$\rceil$ where $\lceil\rceil$ means rounding up to the nearest integer
Return New_PBP

underflowing. On the other hand, when both buffers experience a buffer occupancy of more than 50%, the buffer occupancy itself is used as a weighting factor instead. Now, the algorithm will instruct the buffer having a higher occupancy to have its desired PBP adjusted such that less data is inserted into it in the current MPEG-2 image slice. Hence, buffer overflow problems are alleviated with the aid of Algorithm 1 and Algorithm 2.

The new PBP value is then compared to its legitimate range tabulated in Table 2.4. Furthermore, we restricted the minimum PBP value such that I-, P- and B-pictures have minimum PBP values of 64, 3 and 2, respectively. Since B-pictures are not used for future predictions, it was decided that their data need not be protected as strongly as that of the I- and P-pictures. As for P-pictures, Ghanbari and Seferidis [184] showed that correctly decoded motion vectors alone can still provide a subjectively pleasing reconstruction of the image, even if the DCT coefficients were discarded. Hence, the minimum MPEG-2 bitstream splitting point or PBP for P-pictures has been set to be just before the coded block pattern parameter, which would then ensure that the motion vectors would be mapped to the high-priority partition. Upon receiving corrupted DCT coefficients they would be set to zero, which corresponds to setting the motion-compensated error residual of the macroblock concerned to zero. For I-pictures the fidelity of the reconstructed image is dependent on the number of DCT coefficients that can be decoded successfully. Therefore, the minimum MPEG-2 bitstream splitting point or PBP was set to include at least the first runlength coded DCT coefficient.

Below we demonstrate the operation of Algorithm 1 and Algorithm 2 with the aid of a simple numerical example. We shall assume that the PBP prior to the update is 75 and the buffer occupancy for the high- and low-priority partition buffers is 40% and 10%, respectively. Considering the high-priority partition, according to the buffer occupancy

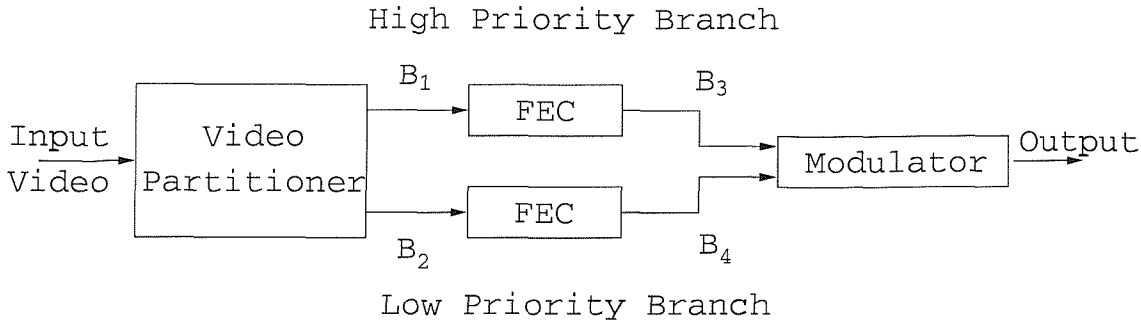


Figure 2.13: Video partitioning scheme for the DVB-T system operating in hierarchical mode.

of 40% Algorithm 1 will set the desired PBP update difference denoted by "diff" for the PBP to +2 and this desired update is referred to as $\text{diff}_{HighPriority}$ in Algorithm 2. For the low-priority partition, according to the buffer occupancy of 10% Algorithm 1 will set the desired update for the PBP to −2, since the sign of diff is changed by Algorithm 1. The desired PBP update for the low-priority partition is referred to as $\text{diff}_{LowPriority}$ in Algorithm 2. Since both partition buffers' occupancy is less than 50%, Algorithm 2 will use the reciprocal of the buffer occupancy as the weighting factor, which will then favour the desired update of the low-priority partition due to its 10 % occupancy. The final update value – which is denoted by delta in Algorithm 2 – is equal to -2 (after being rounded up). Hence, according to Step 2 of Algorithm 2 the new PBP is 73. This means that for the current MPEG-2 image slice more data will be directed into the low-priority partition, in order to prevent buffer underflow since PBP was reduced from 75 to 73 according to Table 2.4.

Apart from adjusting the PBP values from one MPEG-2 image slice to another to avoid buffer underflow or overflow, the output bitrate of each partition buffer must be adjusted such that the input bitrate of the inner interleaver and modulator in Figure 2.9 is properly matched between the two partitions. Specifically, in the 16-QAM mode the two modem subchannels have an identical throughput of 2 bits per 4-bit symbol. By contrast, in the 64-QAM mode there are three 2-bit subchannels per 6-bit 64-QAM symbol, although the standard [13] recommends using a higher-priority 2-bit and a lower-priority 4-bit subchannels. Hence, it is imperative to take into account the redundancy added by Forward Error Correction (FEC), especially when the two partition's FECs operate at different code rates. Figure 2.13 shows a block diagram of the DVB-T system operating in the hierarchical mode and receiving its input from the video partitioner. The FEC module represents the concatenated coding system, consisting of a Reed-Solomon codec [23] and a convolutional codec [23]. The modulator can invoke both 16-QAM and 64-QAM [100]. We shall now use an example to illustrate the choice of the various partitioning ratios summarised in Table 2.5.

We shall assume that 64-QAM is selected and the high- and low-priority video partitions employ rate 1/2 and 3/4 convolutional codes, respectively. This scenario is portrayed

Modulation	Conv. Code Rate (High Priority)	Conv. Code Rate (Low Priority)	Partition Bit Rate Ratio (High Priority : Low Priority) (B1 : B2)
16-QAM	1/2	1/2	1 : 1
	1/2	2/3	3 : 4
	1/2	3/4	2 : 3
	1/2	5/6	3 : 5
	1/2	7/8	4 : 7
	2/3	1/2	4 : 3
64-QAM	1/2	1/2	1 : 2
	1/2	2/3	3 : 8
	1/2	**3/4**	**1 : 3**
	1/2	5/6	3 : 10
	1/2	7/8	2 : 7
	2/3	1/2	2 : 3

Table 2.5: The partitioning ratios for the high- and low-priority partition's output bitrate based on the modulation mode and code rates selected for the DVB-T hierarchical mode. The line in bold corresponds to our worked example.

in the third line of the 64-QAM section of Table 2.5. We do not have to take the Reed-Solomon code rate into account, since both partitions invoke the same Reed-Solomon codec. Based on these facts and upon referring to Figure 2.13, the input bitrates B_3 and B_4 of the modulator must be in the ratio 1:2, since the two MSBs of the 64-QAM constellation are assigned to the high-priority video partition and the remaining four bits to the low-priority video partition.

At the same time, the ratio of B_3 to B_4 is related to the ratio of B_1 to B_2 with the FEC redundancy taken into account, requiring:

$$\begin{aligned} \frac{B_3}{B_4} &= \frac{2\times B_1}{\frac{4}{3}\times B_2} \overset{64-QAM}{=} \frac{1}{2} \\ &= \frac{3}{2}\cdot\frac{B_1}{B_2} \overset{64-QAM}{=} \frac{1}{2} \\ \\ \frac{B_1}{B_2} &= \frac{1}{2}\times\frac{2}{3} \\ &= \frac{1}{3}. \end{aligned} \tag{2.2}$$

If, for example, the input video bitrate to the data partitioner module is 1 Mbit/s, the output bitrate of the high- and low-priority partition would be $B_1 = 250$ kbit/s and $B_2 = 750$ kbit/s, respectively, according to the ratio indicated by Equation 2.2.

In this section, we have outlined the operation of the data partitioning scheme, which we used in the DVB-T hierarchical transmission scheme. Its performance in the context of the overall system will be characterised in Section 2.8. Let us, however, first evaluate the BER-sensitivity of the partitioned MPEG-2 bitstream to randomly distributed bit errors using various partitioning ratios.

2.6 Performance of the Data Partitioning Scheme

Let us consider here the 16-QAM modem and refer to the equally split rate 1/2 convolutional coded high- and low-priority scenario as Scheme 1. Furthermore, the 16-QAM rate 1/3 convolutional coded high-priority data and rate 2/3 convolutional coded low-priority data based scenario is referred to here as Scheme 2. Lastly, the 16-QAM rate 2/3 convolutional coded high-priority data and rate 1/3 coded low-priority data based partitioning scheme is termed as Scheme 3. We then programmed the partitioning scheme of Figure 2.13 for maintaining the required splitting ratio B_1/B_2, as seen in Table 2.6. This was achieved by continuously adjusting the PBP using Algorithm 1 and Algorithm 2. The 704 × 576-pixel "Football" High Definition Television (HDTV) video sequence was used in these investigations.

	Modulation	Conv. Code Rate (High Priority) (B1)	Conv. Code Rate (Low Priority) (B2)	Bit Rate Ratio (High Priority : Low Priority) (B1 : B2)
Scheme 1	16-QAM	1/2	1/2	1 : 1
Scheme 2	16-QAM	1/3	2/3	1 : 2
Scheme 3	16-QAM	2/3	1/3	2 : 1

Table 2.6: Summary of the three schemes employed in our investigations into the performance of the data partitioning scheme. The FEC-coded high-priority video bitstream B3, as shown in Figure 2.13, was mapped to the high-priority 16-QAM subchannel, while the low-priority B4-stream to the low-priority 16-QAM subchannel.

Figures 2.14 to 2.16 show the relative frequency at which a particular PBP value occurs for each image of the "Football" video sequence for the three different schemes of Table 2.6 mentioned earlier. The reader may recall from Table 2.4 that the PBP values indicate the proportion of encoded video parameters, which are to be directed into the high-priority partition. As the PBP value increases, the proportion of video data mapped to the high-priority partition increases and vice versa. Comparing Figures 2.14 to 2.16, we observe that Scheme 3 has the most data in the high-priority partition associated with the high PBPs of Table 2.4, followed by Scheme 1 and Scheme 2. This observation can be explained as follows. We shall consider Scheme 3 first. In this scheme, the high-priority video bits are protected by a rate 2/3 convolutional code and mapped to the higher-integrity 16-QAM subchannel. By contrast, the low-priority video bits are encoded by a rate 1/3 convolutional code and mapped to the lower-integrity 16-QAM subchannel. Again, assuming that 16-QAM is used in our experiment according to line 3 of Table 2.6, 2/3 of the video bits will be placed in the high-priority 16-QAM partition and the remaining video bits in the low-priority 16-QAM partition, following the approach of Equation 2.2. The BER difference of the 16-QAM subchannels depend on the channel error statistics, but the associated BERs are about a factor of 2 – 3 different [100]. In contrast to Scheme 3, Scheme 2 will have 1/3 of the video bits placed in the high-priority 16-QAM partition and the remaining 2/3 of the video bits mapped to the low-priority 16-QAM partition, according to line 2 of Table 2.6. Lastly, Scheme 1 will have half

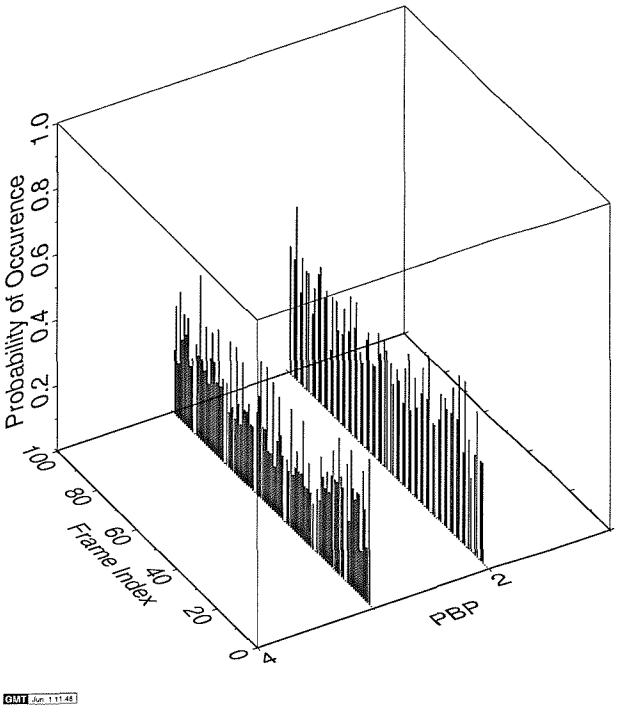
of the video bits in the high- and low-priority 16-QAM partitions, according to line 1 of Table 2.6. This explains our observation in the context of Scheme 3 in Figure 2.16, where a PBP value as high as 80 is achieved in some image frames. However, each PBP value encountered has a lower probability of being selected, since the total number of 3600 occurrences associated with investigated 3600 MPEG-2 video slices per 100 image frames is spread over a higher variety of PBPs. Hence, Scheme 3 directs about 2/3 of the original video bits after 2/3-rate coding to the high-priority 16-QAM subchannel. This observation is in contrast to Scheme 2 of Figure 2.15, where the majority of the PBPs selected are only up to the value of 65. This indicates that about 2/3 of the video bits are concentrated in the lower-priority partition, as indicated in line 2 of Table 2.6.

Figures 2.17(a) to 2.19(a) show the average probability at which a particular PBP value is selected by the rate control scheme, as discussed in Section 2.5, during the encoding of the video sequence. Again, we observe that Scheme 3 encounters the widest range of PBP values, followed by Scheme 1 and Scheme 2, respectively, since according to Table 2.6 these schemes map a decreasing number of bits to the high-priority partition in this order.

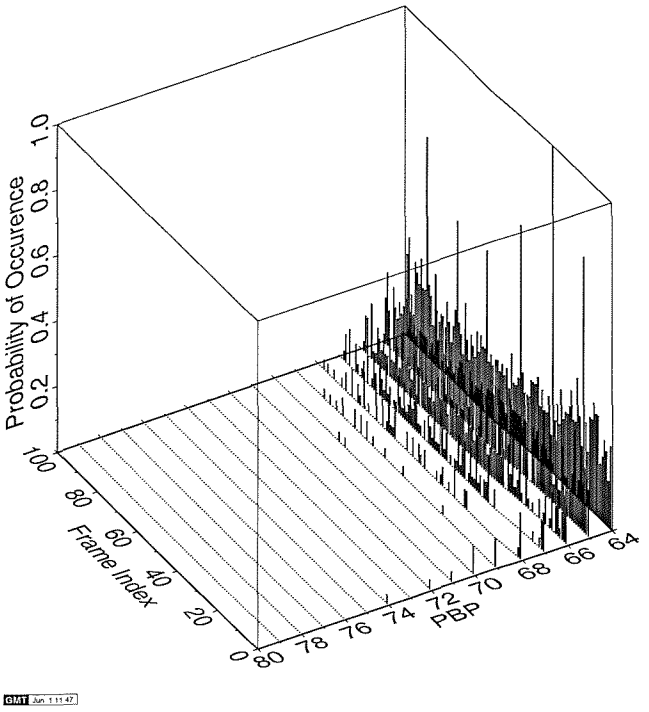
We then embarked on quantifying the error sensitivity of the partitioning Schemes 1 to 3 characterised in Table 2.6, when each partition was subjected to randomly distributed bit errors, although in practice the error distribution will depend on the fading channel's characteristics. Specifically, the previously defined average PSNR degradation was evaluated for given error probabilities inflicting random errors imposed on one of the partitions, while keeping the other partition error-free. These results are portrayed in Figures 2.17(b), 2.18(b) and 2.19(b), for Schemes 1 to 3 respectively.

Comparing Figures 2.17(b) to 2.19(b), we observe that the average PSNR degradation exhibited by the three schemes of Table 2.6, when only their high-priority partitions are corrupted, is similar. The variations in the average PSNR degradation in these cases are caused by the different quantity of sensitive video bits, which resides in the high-priority partition. If we compare the performance of the schemes summarised in Table 2.6 at a BER of 2×10^{-3}, Scheme 3 experienced approximately 8.8 dB average video PSNR degradation, while Schemes 1 and 2 exhibited approximately 5 dB degradation. This trend was expected, since Scheme 3 had the highest portion of the video bits – namely 2/3 – residing in the high-priority partition, followed by Scheme 1 hosting 1/2 and Scheme 2 having 1/3 of the bits in this partition.

On the other hand, we can observe a significant difference in the average PSNR degradation measured for Schemes 1 to 3 of Table 2.6, when only the low-priority partitions are corrupted by comparing the curves shown as broken lines in Figures 2.17(b) to 2.19(b). Under this condition, Scheme 2 experienced approximately 16 dB average video PSNR degradation at a BER of 2×10^{-3}. By contrast, Scheme 1 exhibited an approximately 4 dB average video PSNR degradation, while Scheme 3 experienced about 7.5 dB degradation at this BER. The scheme with the highest portion of video bits in the lower-priority partition, i.e. Scheme 2, experienced the highest average video PSNR degradation. This observation correlates well with our earlier findings in the context of the high-priority

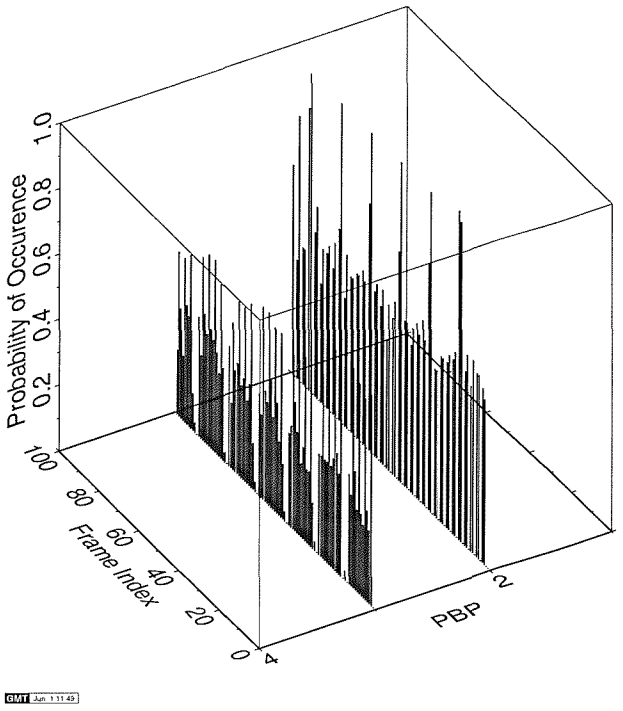


(a)

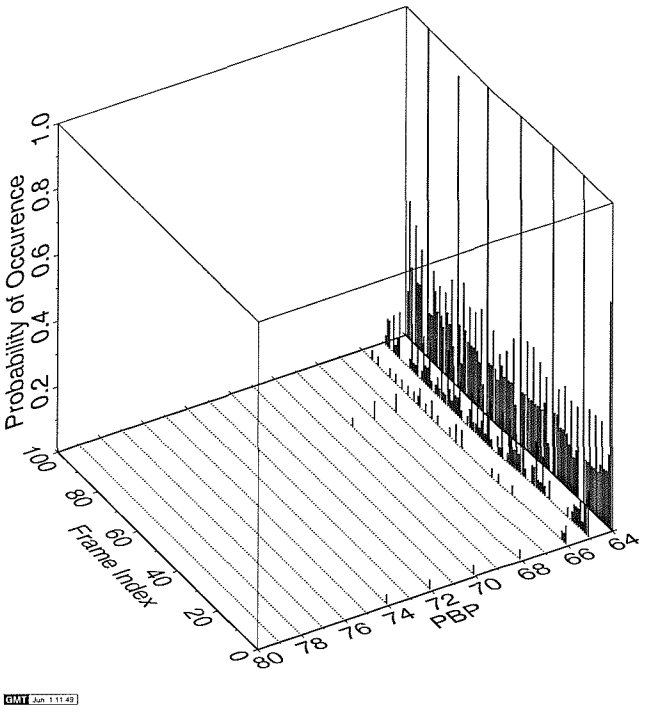


(b)

Figure 2.14: Evolution of the probability of occurrence of PBP values from one picture to another of the 704 × 576-pixel "Football" video sequence for Scheme 1 of Table 2.6.

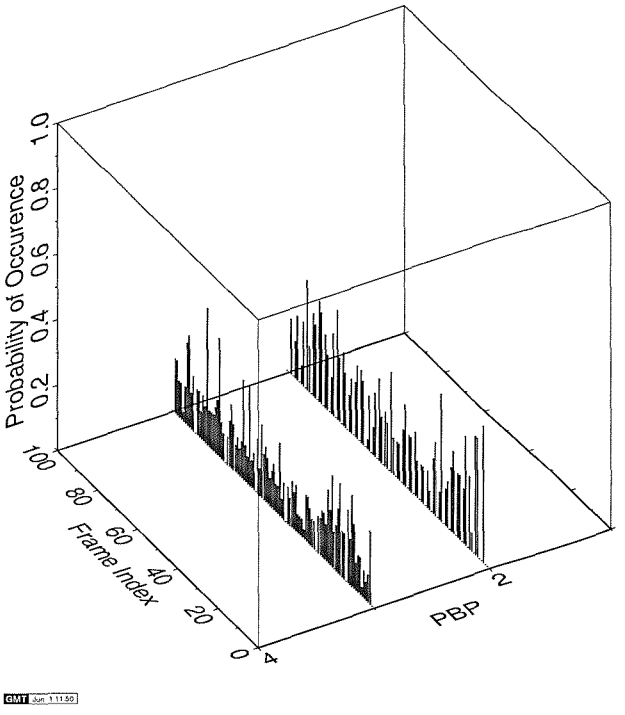


(a)

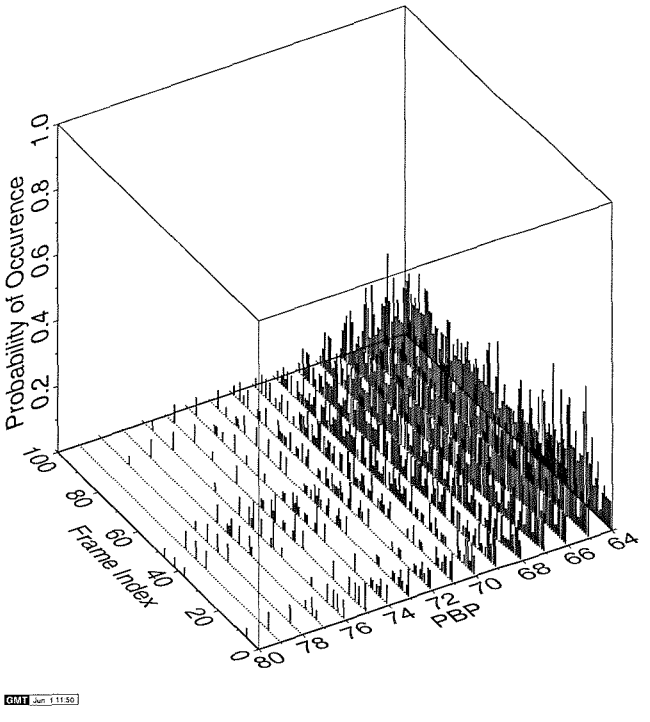


(b)

Figure 2.15: Evolution of the probability of occurrence of PBP values from one picture to another of the 704 × 576-pixel "Football" video sequence for Scheme 2 of Table 2.6.

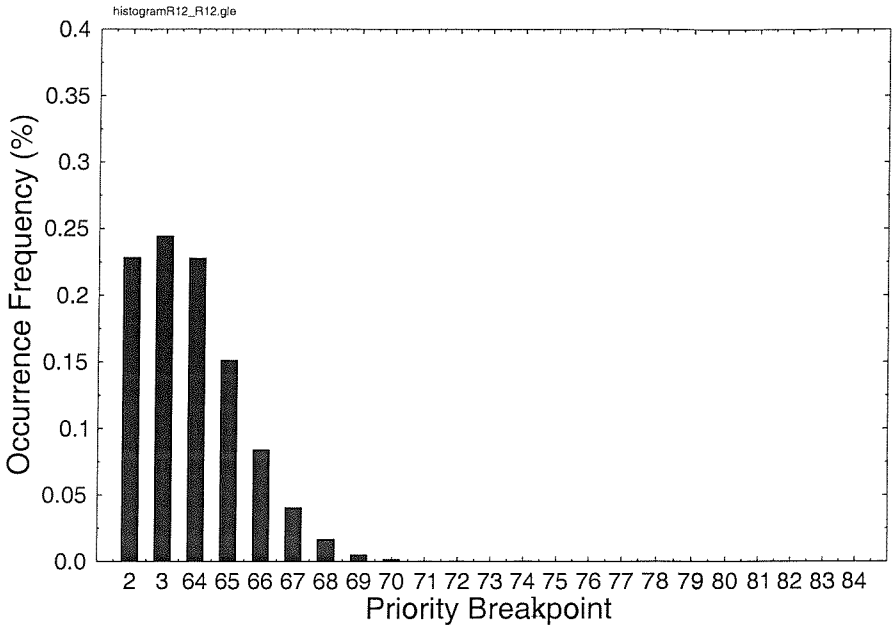


(a)

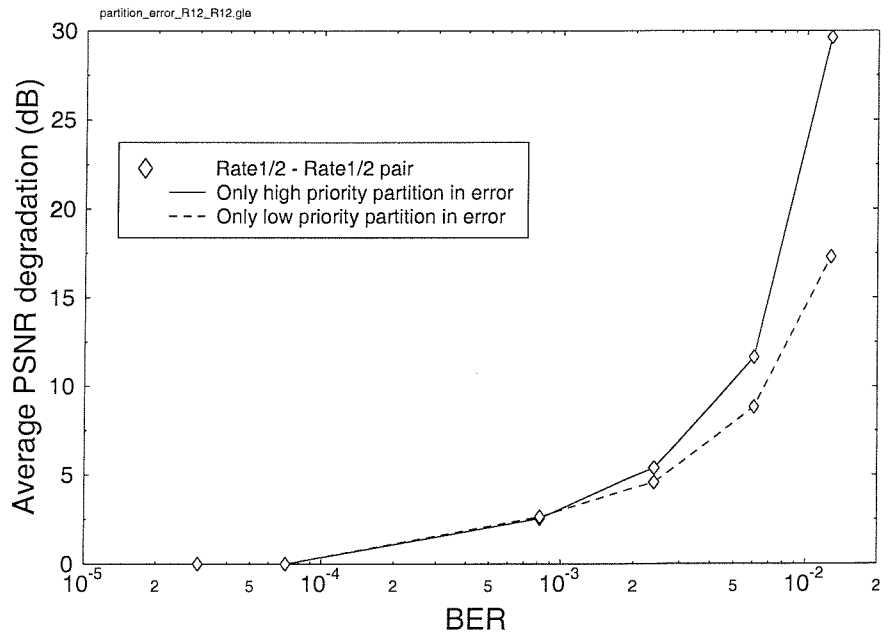


(b)

Figure 2.16: Evolution of the probability of occurrence of PBP values from one picture to another of the 704 × 576-pixel "Football" video sequence for Scheme 3 of Table 2.6.

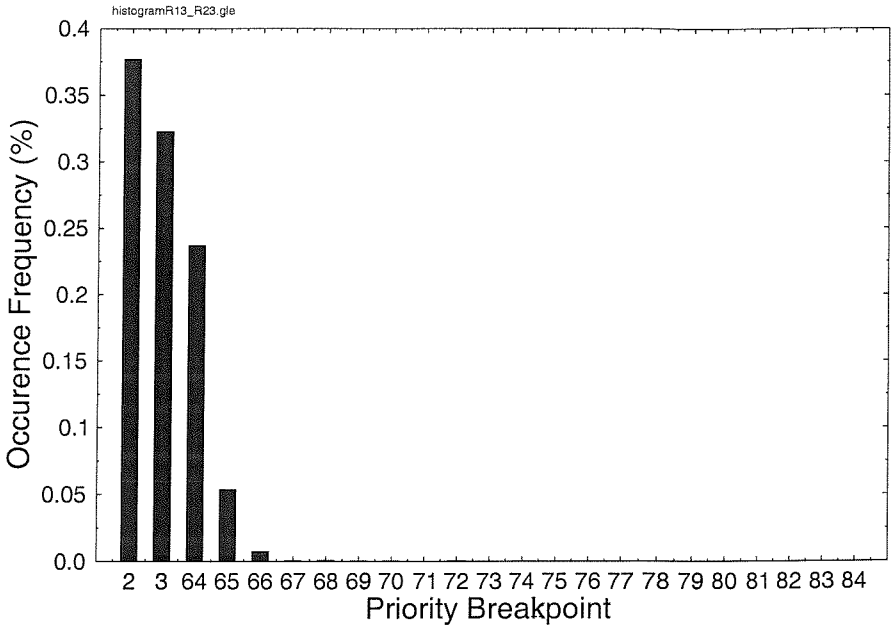


(a)

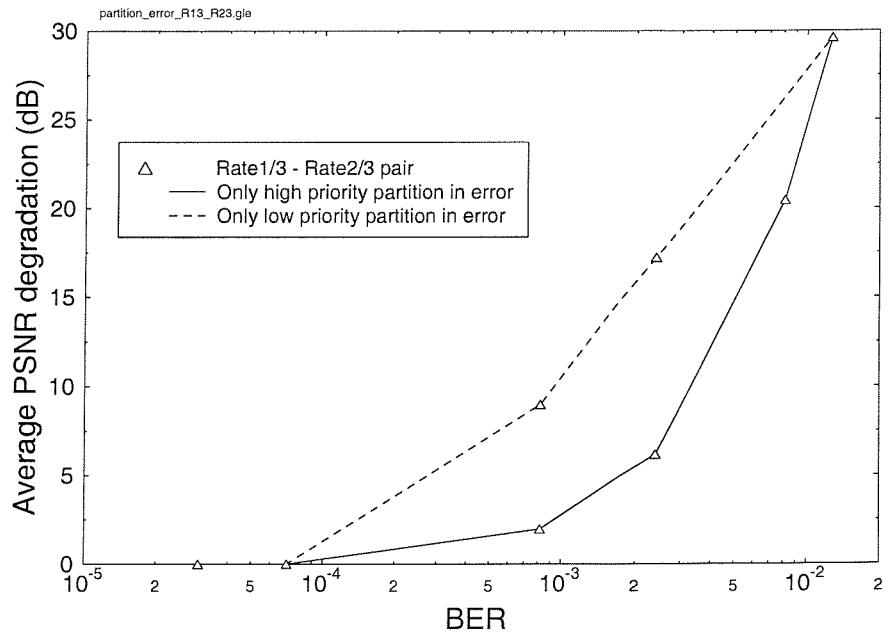


(b)

Figure 2.17: (a) Histogram of the probability of occurrence for various priority breakpoints and (b) average PSNR degradation versus BER for rate 1/2 convolutional coded high- and low-priority data in Scheme 1 of Table 2.6.

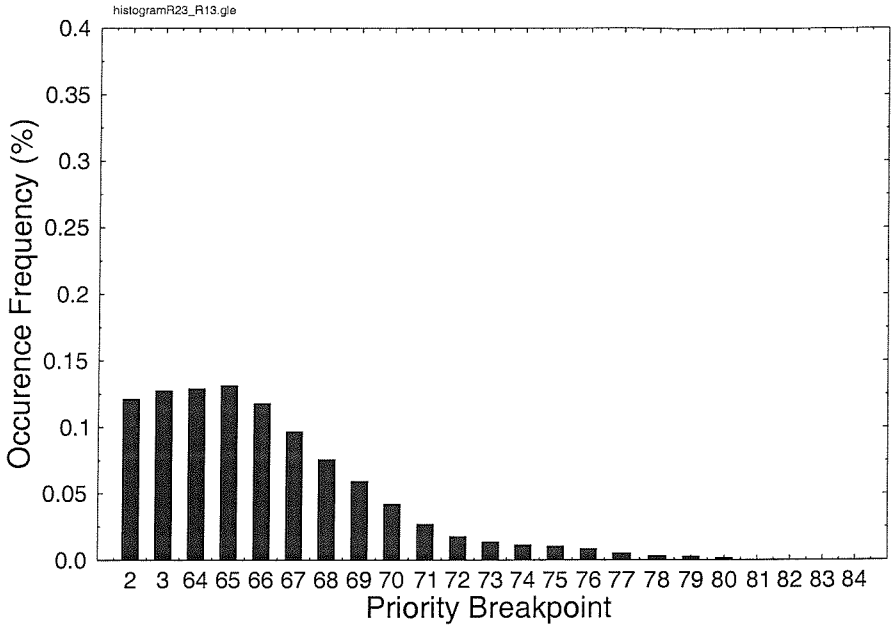


(a)

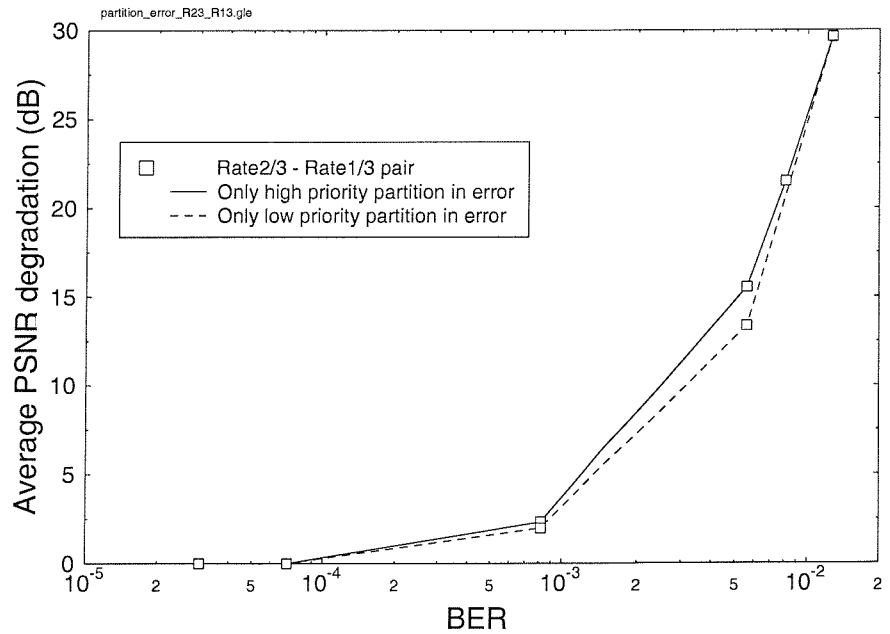


(b)

Figure 2.18: (a) Histogram of the probability of occurrence for various priority breakpoints and (b) average PSNR degradation versus BER for the rate 1/3 convolutional coded high-priority data and rate 2/3 convolutional coded low-priority data in Scheme 2 of Table 2.6.



(a)



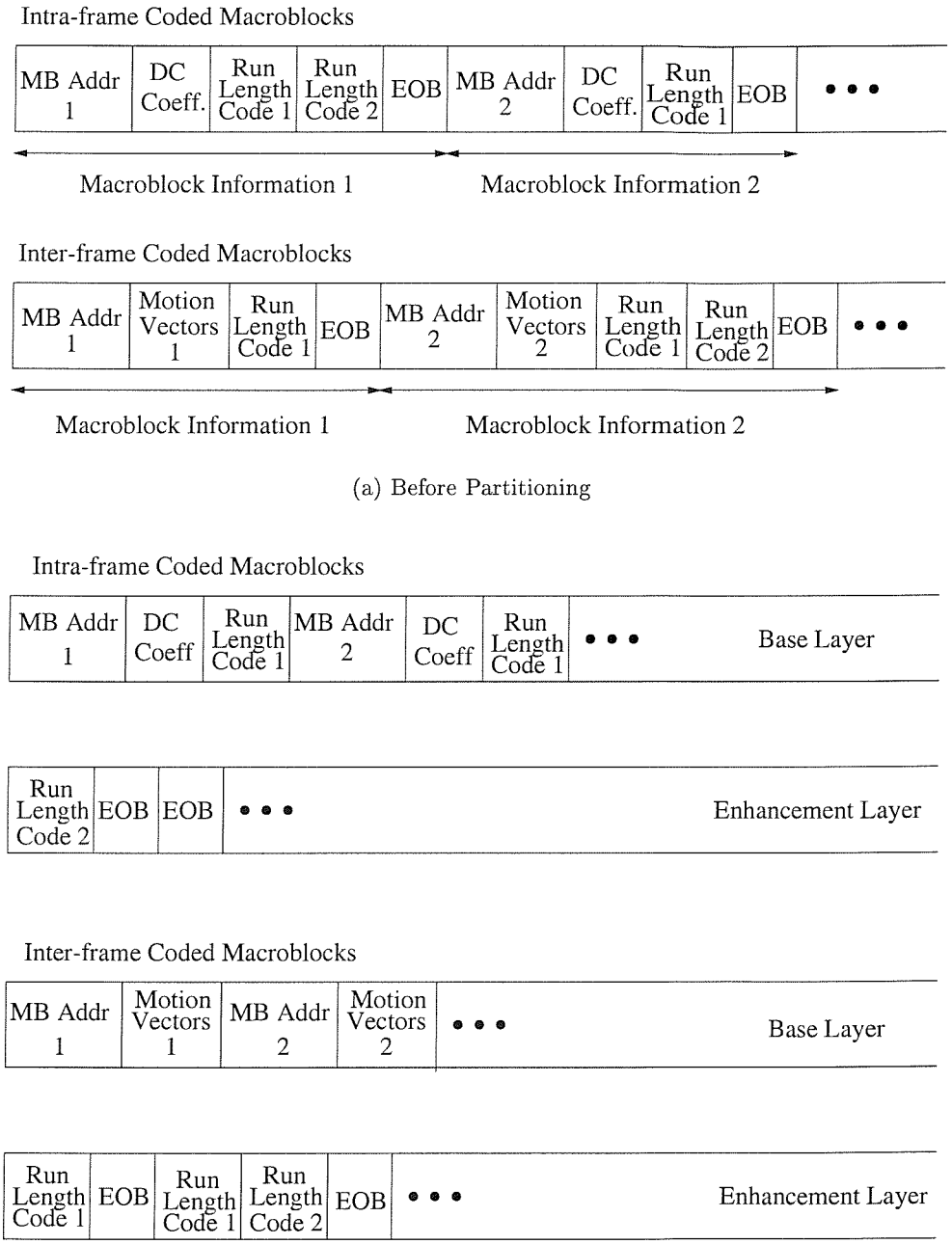
(b)

Figure 2.19: (a) Histogram of the probability of occurrence for various priority breakpoints and (b) average PSNR degradation versus BER for the rate 2/3 convolutional coded high-priority data and rate 1/3 convolutional coded low-priority data in Scheme 3 of Table 2.6.

partition scenario, where the partition holding the highest portion of the video bits in the error-impaired partition, exhibited the highest average PSNR degradation.

Having discussed our observations for the three schemes of Table 2.6 from the perspective of the relative amount of video bits in one partition compared to the other, we shall now examine the details of the data partitioning process further, in order to relate them to our observations. Figure 2.20 shows a typical example of an MPEG-2 video bitstream both prior to data partitioning and after data partitioning. There are two scenarios to be considered here, namely intra-frame coded macroblock partitioning and inter-frame coded macroblock partitioning. We have selected the PBP value of 64 from Table 2.4 for the intra-frame coded macroblock scenario and the PBP value of 3 for the inter-frame coded macroblock scenario, since these values have been selected frequently by the rate control arrangement for Schemes 1 and 2. This is evident from Figures 2.14 and 2.15 as well as from Figures 2.17(a) and 2.18(a). This implies, with the aid of Table 2.4 and Figure 2.20, that only the macroblock (MB) header information and a few low-frequency DCT coefficients will reside in the high-priority partition, while the rest of the DCT coefficients will be stored in the low-priority partition. These can be termed as base layer and enhancement layer, as seen in Figure 2.20. In the worst-case scenario, where the entire enhancement layer or low-priority partition data is lost due to a transmission error near the beginning of the associated low-priority bitstream, the MPEG-2 video decoder will only have the bits of the high-priority partition in order to reconstruct the encoded video sequence. Hence, the MPEG-2 decoder certainly cannot reconstruct good quality images. Although the results reported by Ghanbari and Seferidis [184] suggested that adequate video reconstruction is possible, provided that the motion vectors are correctly decoded, this observation is only true if the previous intra-coded frame is correctly reconstructed. If the previous intra-coded frame contains artifacts, these artifacts will be further propagated to forthcoming video frames by the motion vectors. By attempting to provide higher protection for the high-priority partition or base layer, we have indirectly forced the rate control scheme, of Section 2.5, to reduce the proportion of video bits directed into the high-priority partition under the constraint of a given fixed bitrate, which is imposed by the 16-QAM subchannels.

In order to elaborate a little further, at a BER of 2×10^{-3} Scheme 1 in Figure 2.17(a) exhibited a near-identical PSNR degradation for the high- and low-priority video bits. Upon assigning more bits to the low-priority partition, in order to be able to accommodate a stronger FEC code in the high-priority partition results in an increased proportion of error-impaired bits and the associated higher error sensitivity seen in Figure 2.18(b). As such, there is a trade-off between the amount of video data protected and the code rate of the channel codec. As a comparison to the above scenarios in the context of Schemes 1 and 2, we shall now examine Scheme 3. In this scheme, more video data – namely half the bits – can be directed into the high-priority partition, as demonstrated by Figure 2.16 due to encountering higher PBPs. This can also be confirmed with reference to Figures 2.18(b) and 2.19(b) by observing the PSNR degradations associated with the curves plotted in broken lines. If the low-priority partition is lost in Scheme 3, its effect on the quality of the



(a) Before Partitioning

(b) After Partitioning

Figure 2.20: Example of video bitstream (a) before data partitioning and (b) after data partitiong for intra-frame coded macroblocks (MB) assuming a PBP of 64 and for inter-frame coded macroblocks assuming a PBP of 3.

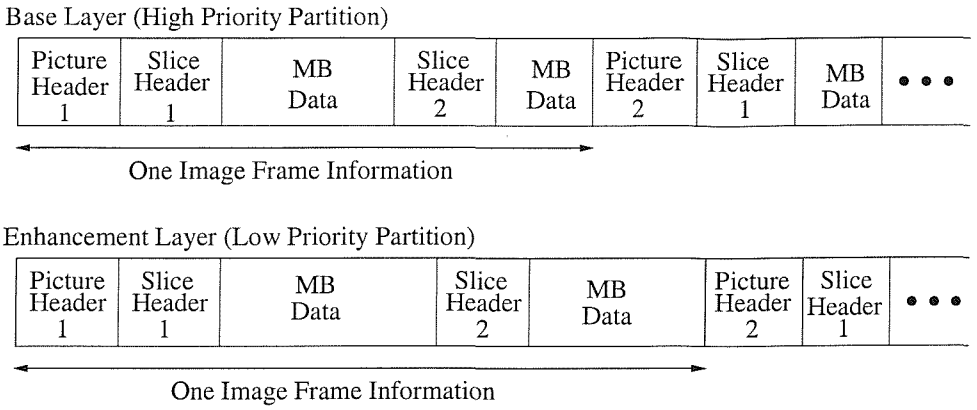


Figure 2.21: Example of high level bitstream syntax structure of a data partitioned MPEG-2 video bitstream. The "MB data" shown in the diagram refers to the macroblock header information and to the variable-length coded DCT coefficients, which have been partitioned as shown in Figure 2.20.

reconstructed images is less detrimental than that of Scheme 2, since Scheme 3 loses only half the bits, rather than 2/3. Hence, it is interesting to note that Scheme 3 experiences slightly higher average PSNR degradation than Scheme 1 at a BER of 2×10^{-3}, when only the low-priority partition is lost in both cases, despite directing only 1/3 rather than 1/2 of the bits to the low-priority partition. This observation can be explained as follows.

Apart from partitioning the macroblock header information and the variable-length coded DCT coefficients into the high- and low-priority partitions, synchronization information such as the Picture Header Information [4] is replicated in the enhancement layer, as suggested by Gharavi *et al.* [183] as well as the MPEG-2 standard [4]. The purpose is to enable the MPEG-2 decoder to keep the base and enhancement layers synchronised during decoding. An example of this arrangement is shown in Figure 2.21. This resynchronisation measure is only effective, when the picture start code of both the high- and low-priority partitions are received correctly. If the picture start code in the low-priority partition is corrupted, for example, the MPEG-2 decoder may not detect this PSC and all the data corresponding to the current image frame in the low-priority partition will be lost. The MPEG-2 decoder will then interpret the bits received for the low-priority partition of the next frame as the low-priority data expected for the current frame. As expected, due to this synchronisation problem the decoded video would have a higher average PSNR degradation, than for the case where picture start codes are unimpaired. This explains our observation of a higher average PSNR degradation for Scheme 3, when only its lower-priority partition was corrupted by the transmission channel. On the other hand, in this specific experiment, Scheme 1 did not experience the loss of synchronisation due to corruption of its picture start code. Viewing events from another perspective, by opting for allocating less useful video bits to the low-priority partition, the probability of transmission errors affecting the fixed-length PSC within the reduced-sized low-priority partition becomes higher.

These findings will assist us in explaining our observations in the context of the hierarchical transmission scheme of Section 2.8, suggesting that the data partitioning scheme

did not provide overall gain in terms of error resilience over the non-partitioned case. Let us, however, consider first the performance of the non-hierarchical DVB-T scheme in the next section.

2.7 Performance of the DVB Terrestrial Scheme Employing Non-hierarchical Transmission

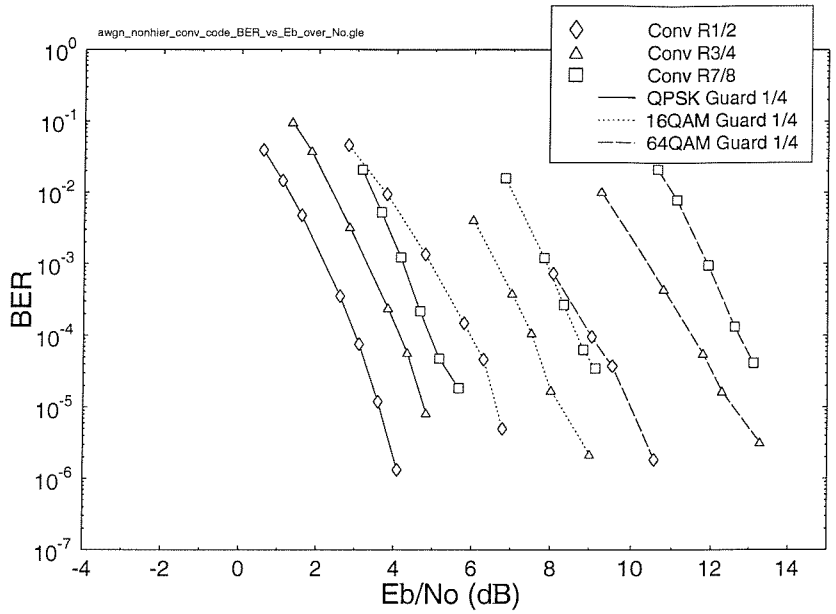
In this section we shall elaborate on our findings when the convolutional code used in the standard non-hierarchical DVB scheme [13] is replaced by a turbo code. We will invoke a range of standard-compliant schemes as benchmarks. The 704 × 576-pixel HDTV-resolution "Football" video sequence was used in our experiments. The MPEG-2 decoder employs a simple error concealment algorithm to fill in missing portions of the reconstructed image in the event of decoding errors. The concealment algorithm will select the specific portion of the previous reconstructed image, which corresponds to the missing portion of the current image, in order to conceal the errors.

In Figures 2.22(a) and 2.22(b) the bit error rate (BER) performance of the various modem modes in conjunction with our diverse channel coding schemes are portrayed over stationary, narrowband Additive White Gaussian Noise channels (AWGN), where the turbo codec exhibits a significantly steeper BER reduction in comparison to the convolutionally coded arrangements.

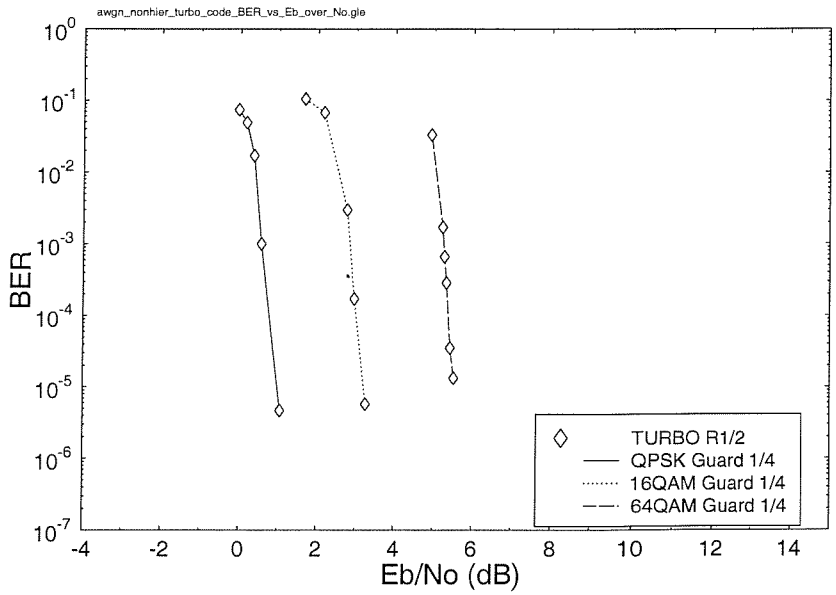
Specifically, comparing the performance of the various turbo and convolutional codes for QPSK and 64-QAM at a BER of 10^{-4}, the turbo code exhibited an additional coding gain of about 2.24 dB and 3.7 dB respectively, when using half-rate codes in Figures 2.22(a) and 2.22(b). Hence, the Peak Signal to Noise Ratio (PSNR) versus channel Signal-to-Noise Ratio (SNR) graphs in Figure 2.25 demonstrate that approximately 2 dB and 3.5 dB lower channel SNRs are required in conjunction with the rate 1/2 turbo codec for QPSK and 64-QAM, respectively, than for convolutional coding, in order to maintain high reconstructed video quality. The term unimpaired as used in Figure 2.25 and Figure 2.26 refers to the condition, where the PSNR of the MPEG-2 decoder's reconstructed image at the receiver is the same as the PSNR of the same image generated by the local decoder of the MPEG-2 video encoder, corresponding to the absence of channel – but not MPEG-2 coding – impairments.

Comparing the BER performance of the 1/2-rate convolutional decoder in Figure 2.23(a) and the so-called Log-Map [181] turbo decoder using eight iterations in Figure 2.23(b) for QPSK modulation over the worst-case fading mobile channel of Figure 2.11, we observe that at a BER of about 10^{-4} the turbo code provided an additional coding gain of 6 dB in comparison to the convolutional code. By contrast, for 64-QAM using similar codes, a 5 dB coding gain was observed at this BER.

Similar observations were also made with respect to the average Peak Signal to Noise Ratio (PSNR) versus channel Signal to Noise Ratio (SNR) plots of Figure 2.26. For example, for the QPSK modulation mode and a 1/2 coding rate, the turbo code required an approximately 5.5 dB lower channel SNR for maintaining near-unimpaired video quality than the convolutional code.

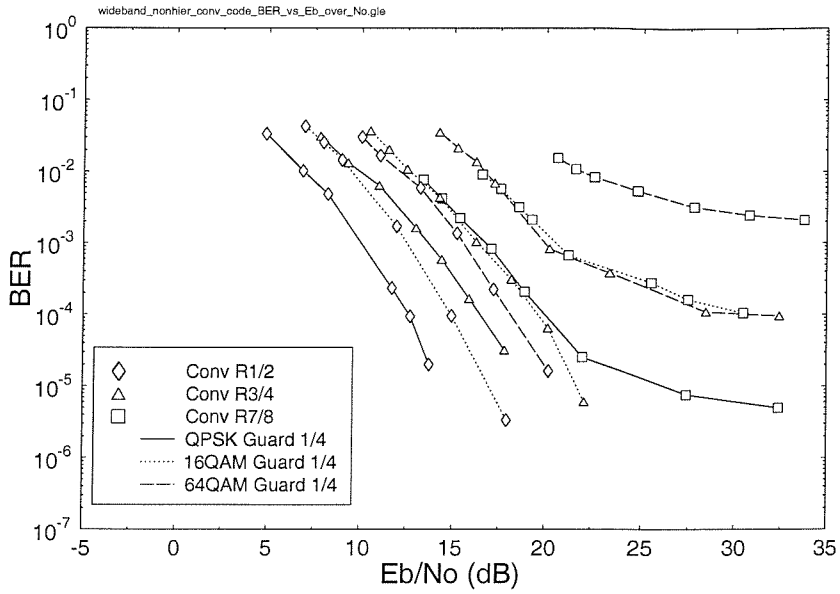


(a) Convolutional Code

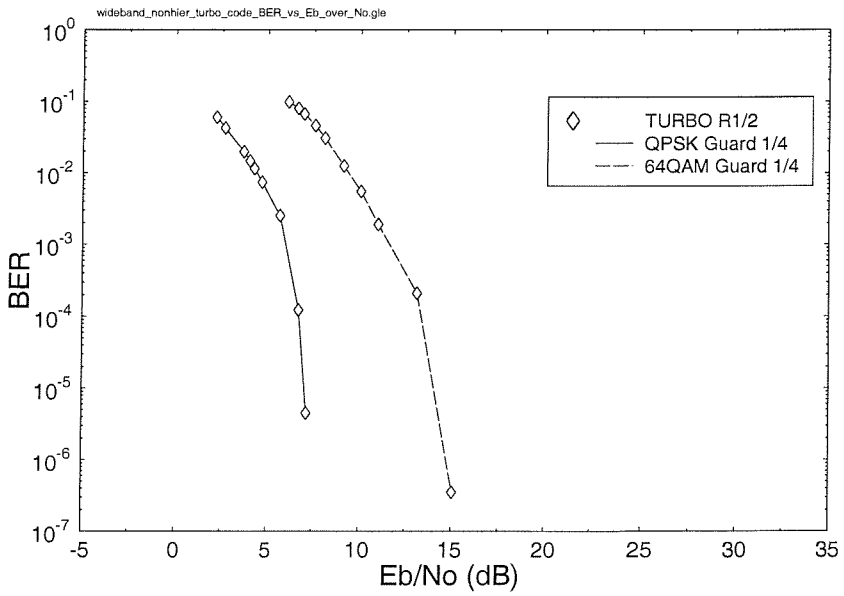


(b) Turbo Code

Figure 2.22: BER after (a) convolutional decoding and (b) turbo decoding for the DVB-T scheme over stationary, non-dispersive **AWGN** channels for **non-hierarchical transmission**.

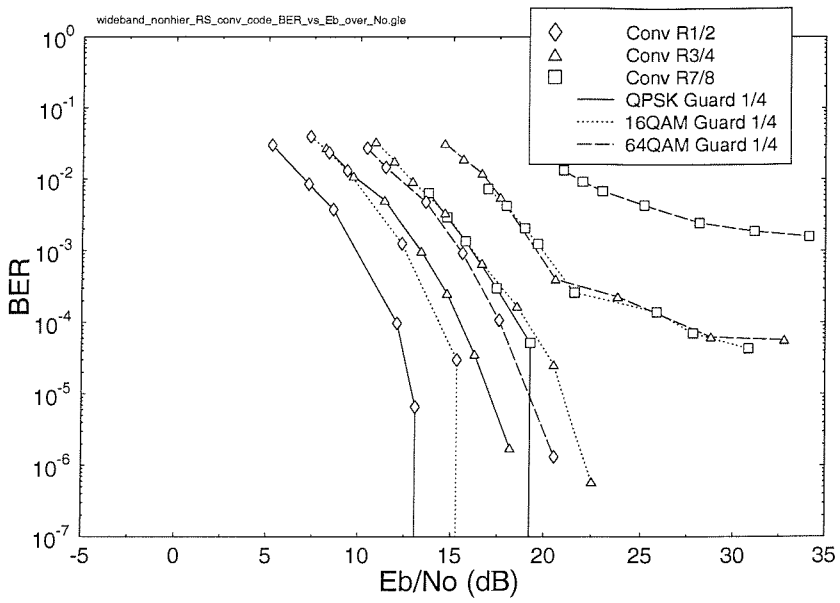


(a) Convolutional Code

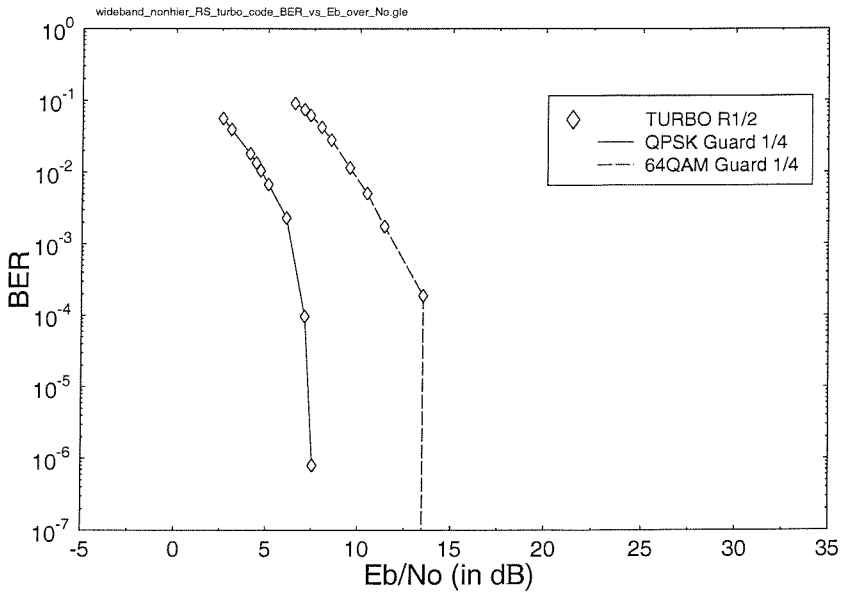


(b) Turbo Code

Figure 2.23: BER after (a) convolutional decoding and (b) turbo decoding for the DVB-T scheme over the **wideband fading channel** of Figure 2.11 for **non-hierarchical transmission.**



(a) RS and Convolutional Code



(b) RS and Turbo Code

Figure 2.24: BER after (a) RS and convolutional decoding and (b) RS and turbo decoding for the DVB-T scheme over the **wideband fading channel** of Figure 2.11 for **non-hierarchical transmission.**

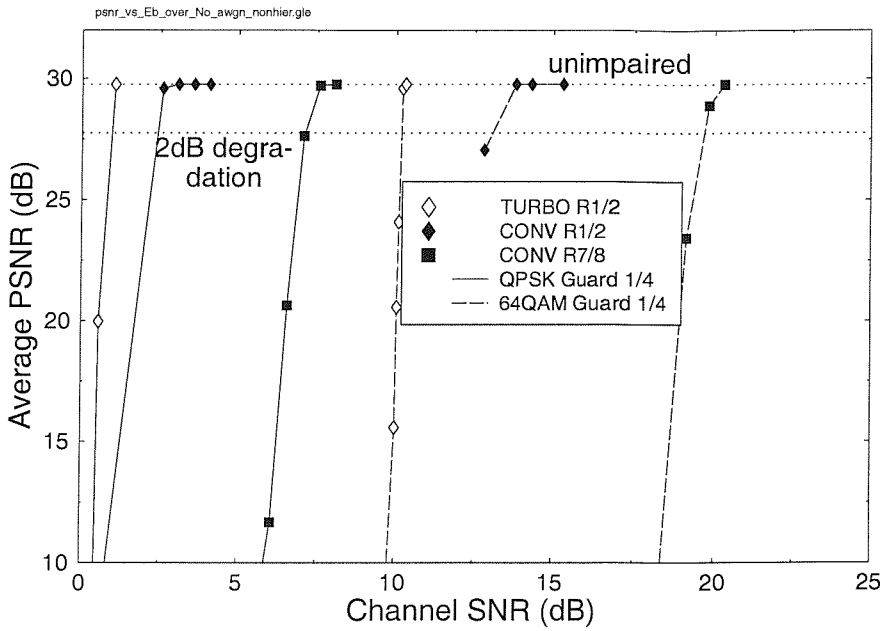


Figure 2.25: Average PSNR versus channel SNR of the DVB scheme [13] over non-dispersive **AWGN** channels for **non-hierarchical transmission**.

Comparing Figure 2.23(a) and Figure 2.24(a), we note that the Reed-Solomon decoder becomes effective in lowering the bit error probability of the transmitted data further below the BER threshold of 10^{-4}. From these figures we also observe that the rate 3/4 convolutional code is unsuitable for transmission over the highly dispersive hilly terrain channel used in this experiment, when 64-QAM is employed. When the rate 7/8 convolutional code is used, both the 16-QAM and 64-QAM schemes perform poorly. As for the QPSK modulation scheme, a convolutional code rate as high as 7/8 can still provide a satisfactory performance after Reed-Solomon decoding.

In conclusion, Tables 2.7 and 2.8 summarize the system performance in terms of the Channel SNR (CSNR) required for maintaining less than 2 dB PSNR video degradation. It was observed that at this PSNR degradation decoding errors were still perceptually unnoticable to the viewer due to the 30 frames/s refresh-rate, although the typical still-frame shown in Figure 2.27 in this scenario exhibits some degradation. It is important to underline once again that the $K = 3$ turbo code and the $K = 7$ convolutional code exhibited comparable complexities. The higher performance of the turbo codec facilitates for example the employment of turbo-coded 16-QAM at a similar channel SNR, where convolutional-coded QPSK can be invoked. This in turn allows us to double the bitrate within the same bandwidth and hence improve the video quality. In the next section, we shall present the results of our investigations employing the DVB-T system [13] in a hierarchical transmission scenario.

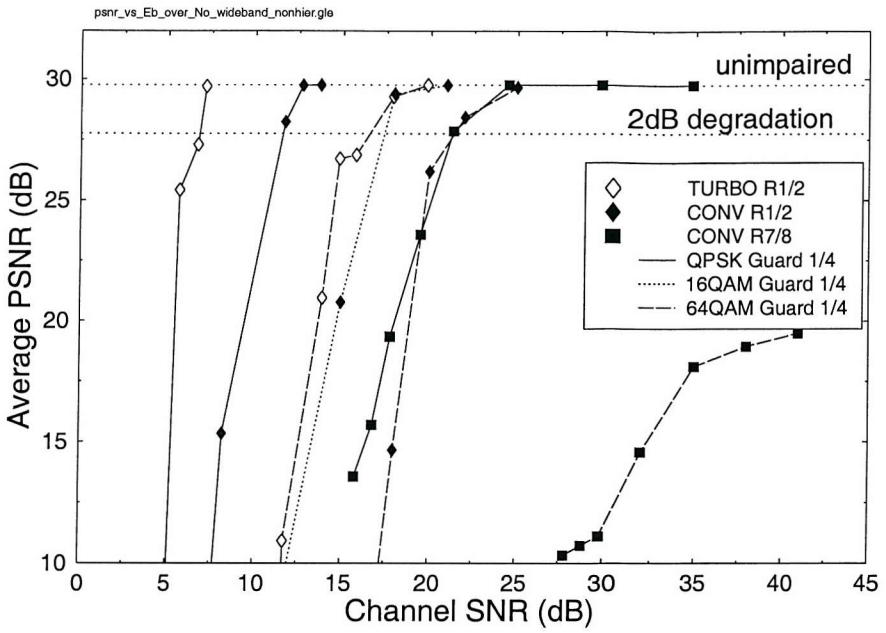


Figure 2.26: Average PSNR versus channel SNR of the DVB scheme [13] over the **wideband fading channel** of Figure 2.11 for **non-hierarchical transmission.**

Figure 2.27: Frame 79 of "Football" sequence, which illustrates the visual effects of minor decoding errors at a BER of 2.10^{-4} after convolutional decoding. The PSNR degradation observed is approximately 2 dB. The sequence was coded using a rate 7/8 convolutional code and transmitted employing QPSK modulation.

Mod.	Code	CSNR (dB)	E_b/N_0	BER
QPSK	Turbo (1/2)	1.02	1.02	$6{\cdot}10^{-6}$
64-QAM	Turbo (1/2)	9.94	5.17	$2{\cdot}10^{-3}$
QPSK	Conv (1/2)	2.16	2.16	$1.1{\cdot}10^{-3}$
64-QAM	Conv (1/2)	12.84	8.07	$6{\cdot}10^{-4}$
QPSK	Conv (7/8)	6.99	4.56	$2{\cdot}10^{-4}$
64-QAM	Conv (7/8)	19.43	12.23	$3{\cdot}10^{-4}$

Table 2.7: Summary of the **non-hierarchical** performance results over non-dispersive **AWGN** channels tolerating a PSNR degradation of 2dB. The BER measure refers to BER after Viterbi or turbo decoding.

Mod.	Code	CSNR (dB)	E_b/N_0	BER
QPSK	Turbo (1/2)	6.63	6.63	$2.5{\cdot}10^{-4}$
64-QAM	Turbo (1/2)	15.82	11.05	$2{\cdot}10^{-3}$
QPSK	Conv (1/2)	10.82	10.82	$6{\cdot}10^{-4}$
64-QAM	Conv (1/2)	20.92	16.15	$7{\cdot}10^{-4}$
QPSK	Conv (7/8)	20.92	18.49	$3{\cdot}10^{-4}$

Table 2.8: Summary of the **non-hierchical** performance results over **wideband fading channels** tolerating a PSNR degradation of 2dB. The BER measure refers to BER after Viterbi or turbo decoding.

2.8 Performance of the DVB Terrestrial Scheme Employing Hierarchical Transmission

The philosophy of the hierarchical transmission mode is that the natural BER difference of a factor 2 to 3 of the 16-QAM modem is exploited for providing unequal error protection for the FEC-coded video streams B3 and B4 of Figure 2.13 [100]. If the sensitivity of the video bits requires a different BER ratio between the B3 and B4 streams, the choice of the FEC codes protecting the video streams B1 and B2 of Figure 2.13 can be appropriately adjusted to equal out or to augment these differences.

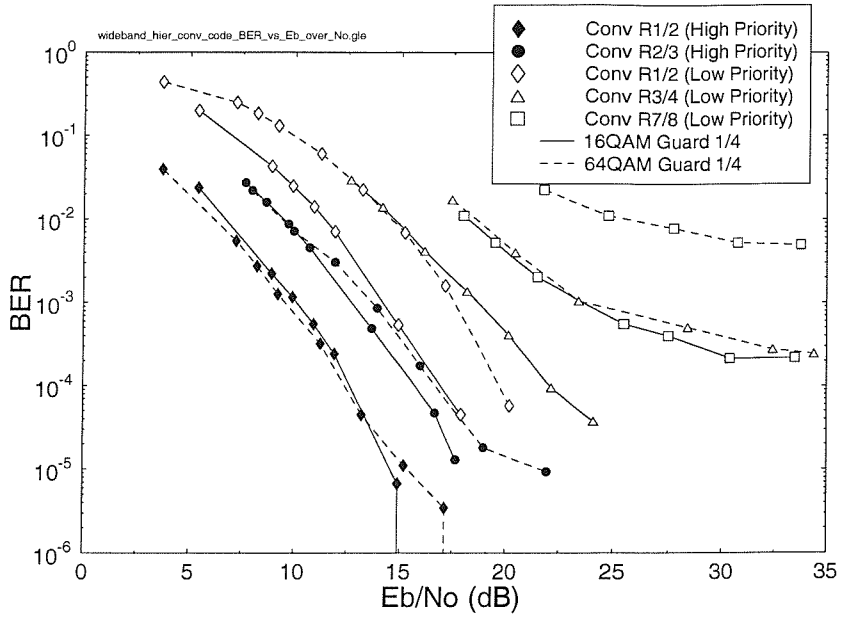
Below we will invoke the DVB-T hierarchical scheme in a mobile broadcasting scenario. We shall also demonstrate the improvements which turbo codes offer, when replacing the convolutional code in the standard scheme. Hence, the convolutional codec in both the high- and low-priority partitions was replaced by the turbo codec. We have also investigated replacing only the high-priority convolutional codec with the turbo codec, pairing the 1/2-rate turbo codec in the high-priority partition with the convolutional codec in the low-priority partition. Again, the "Football" sequence was used in these experiments. Partitioning was carried out using the schematic of Figure 2.13 as well as Algorithms 1 and 2. The FEC-coded high-priority video partition B3 of Figure 2.13 was mapped to the higher-integrity 16-QAM or 64-QAM subchannel. By contrast, the low-priority partition B4 of Figure 2.13 was directed to the lower-integrity 16-QAM or 64-QAM subchannel. Lastly, no specific mapping was required for QPSK, since it exhibits no subchannels. We note, however, that further design trade-offs become feasible, when

reversing the above mapping rules. Indeed, this is necessary, for example, in conjunction with Scheme 2 of Table 2.6, since the high number of bits in the low-priority portion render it more sensitive than the high-priority partition. Again, the 16-QAM subchannels exhibit a factor of 2 to 3 BER difference under various channel conditions, which improves the robustness of the reverse-mapped Scheme 2 of Table 2.6.

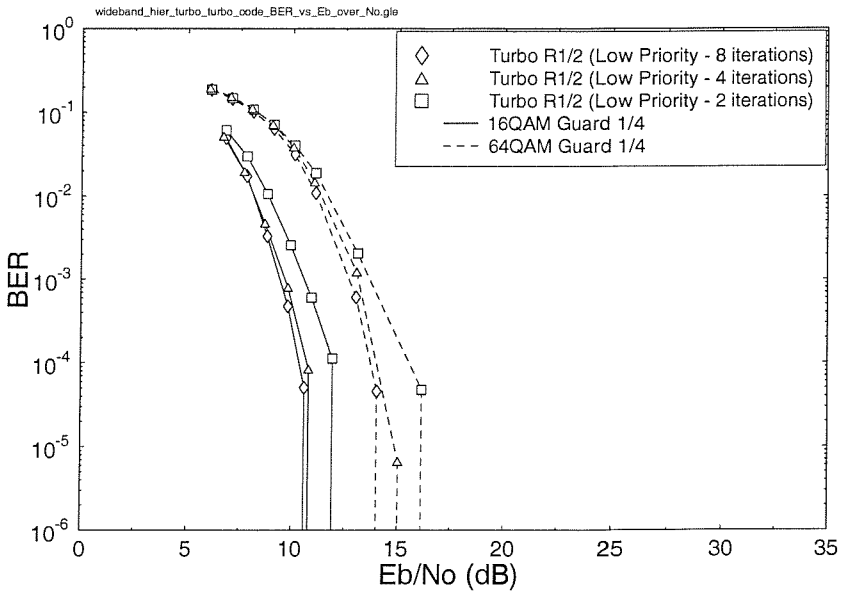
Referring to Figure 2.28 and comparing the performance of the 1/2-rate convolutional code and turbo code at a BER of 10^{-4} for the low-priority partition, the turbo code, employing 8 iterations, exhibited a coding gain of about 6.6 dB and 5.97 dB for 16-QAM and 64-QAM, respectively. When the number of turbo decoding iterations was reduced to 4, the coding gains offered by the turbo code over that of the convolutional code were 6.23 dB and 5.7 dB for 16-QAM and 64-QAM respectively. We observed that by reducing the number of iterations to 4 halved the associated complexity, but the turbo code exhibited a coding loss of only about 0.37 dB and 0.27 dB in comparison to the 8-iteration scenario for 16-QAM and 64-QAM, respectively. Hence, the computational complexity of the turbo codec can be halved by sacrificing only a small amount of coding gain. The substantial coding gain provided by turbo coding is also reflected in the PSNR versus channel SNR graphs of Figure 2.30. In order to achieve transmission with very low probability of error, Figure 2.30 demonstrated that approximately 5.72 dB and 4.56 dB higher channel SNRs are required by the standard scheme compared to the scheme employing turbo coding, when using 4 iterations in both partitions. We have only shown the performance of turbo coding for the low-priority partition in Figures 2.28(b) and 2.29(b), since the turbo or convolutional-coded high-priority partition was received with very low probability of error after Reed-Solomon decoding for the range of SNRs used.

We also observed that the rate 3/4 and rate 7/8 convolutional codes in the low-priority partition were unable to provide sufficient protection to the transmitted video bits, as it becomes evident from Figures 2.28(a) and 2.29(a). In these high coding rate scenarios, due to the presence of residual errors even after the Reed-Solomon decoder, the decoded video exhibited some decoding errors, which is evidenced by the flattening of the PSNR versus channel SNR curves in Figure 2.30(a), before reaching the error free PSNR.

A specific problem faced when using the data partitioning scheme in conjunction with the high-priority partition being protected by the rate 1/2 code and the low-priority partition protected by the rate 3/4 and 7/8 codes was that when the low-priority partition data was corrupted, the error-free high-priority data available was insufficient for concealing the errors, as discussed in Section 2.6. We have also experimented with the combination of rate 2/3 convolutional coding and rate 1/2 convolutional coding, in order to protect the high- and low-priority data, respectively. From Figure 2.30(a) we observed that the performance of this 2/3- and 1/2-rate combination approached that of the rate 1/2 convolutional code in both partitions. This was expected, since now more data can be inserted into the high-priority partition. Hence, in the event of decoding errors in the low-priority data we had more error-free high-priority data that can be used to reconstruct the received image.

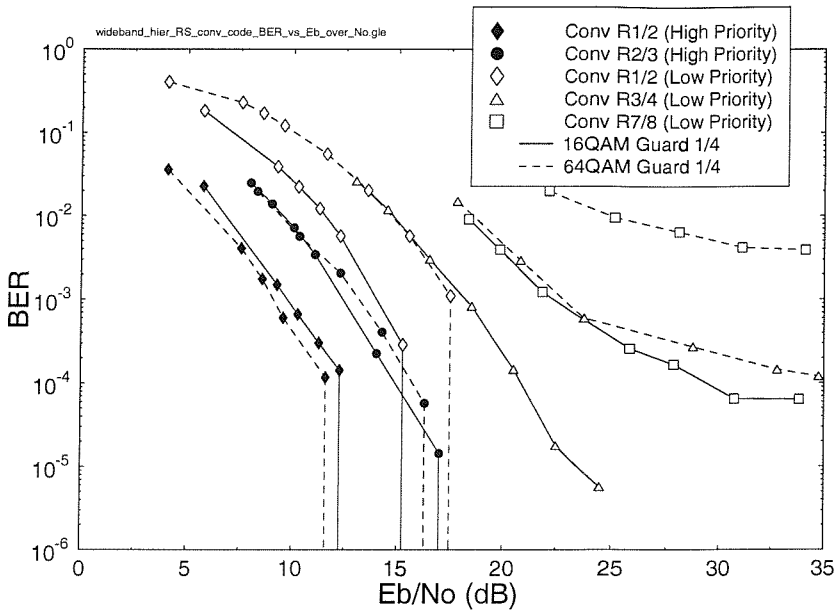


(a) Convolutional Code

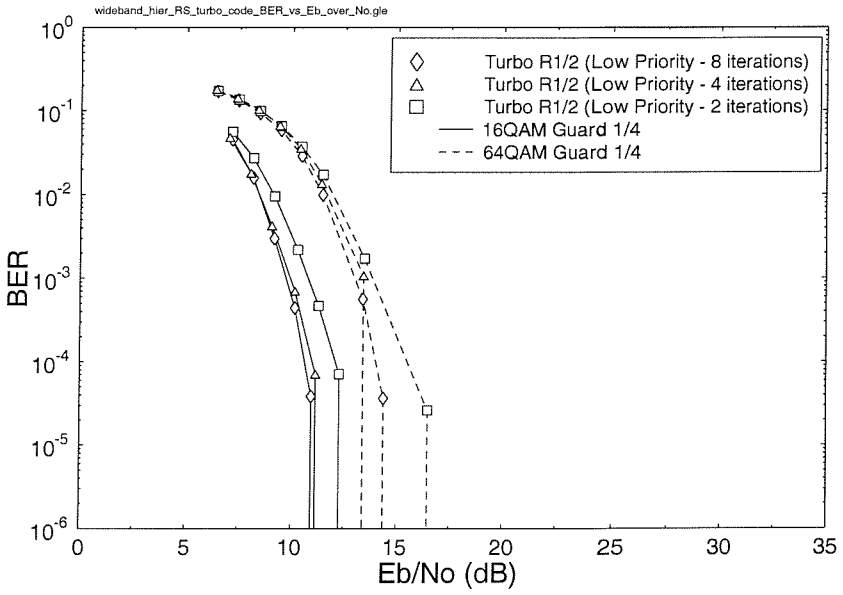


(b) Turbo Code

Figure 2.28: BER after (a) convolutional decoding and (b) turbo decoding for the **DVB-T hierarchical scheme** over the **wideband fading channel** of Figure 2.11 using the schematic of Figure 2.13 as well as Algorithms 1 and 2. In (b), the BER of the turbo or convolutional-coded high-priority partition is not shown.

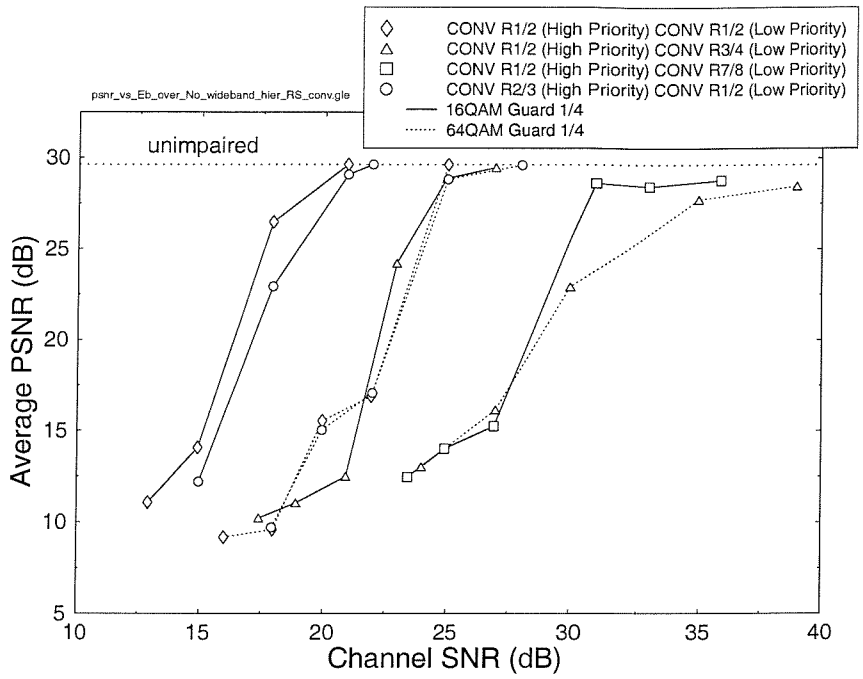


(a) RS and Convolutional Code

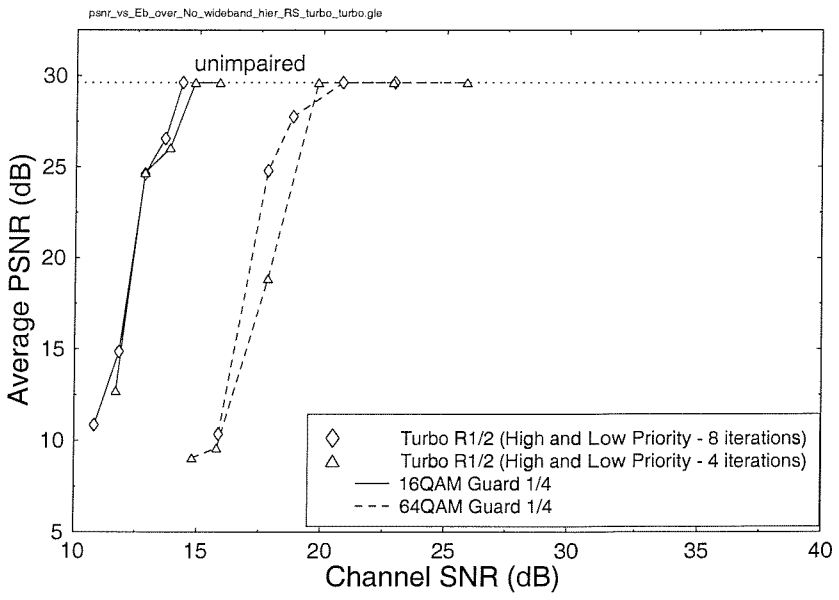


(b) RS and Turbo Code

Figure 2.29: BER after (a) RS and convolutional decoding and (b) RS and turbo decoding for the **DVB-T hierarchical scheme** over the **wideband fading channel** of Figure 2.11 using the schematic of Figure 2.13 as well as Algorithms 1 and 2. In (b), the BER of the turbo or convolutional-coded high-priority partition is not shown.



(a) Standard Scheme



(b) Turbo Code in both partitions

Figure 2.30: Average PSNR versus channel SNR for (a) standard DVB scheme [13] and (b) system with turbo coding employed in both partitions, for transmission over the **wideband fading channel** of Figure 2.11 for **hierarchical transmission** using the schematic of Figure 2.13 as well as Algorithms 1 and 2.

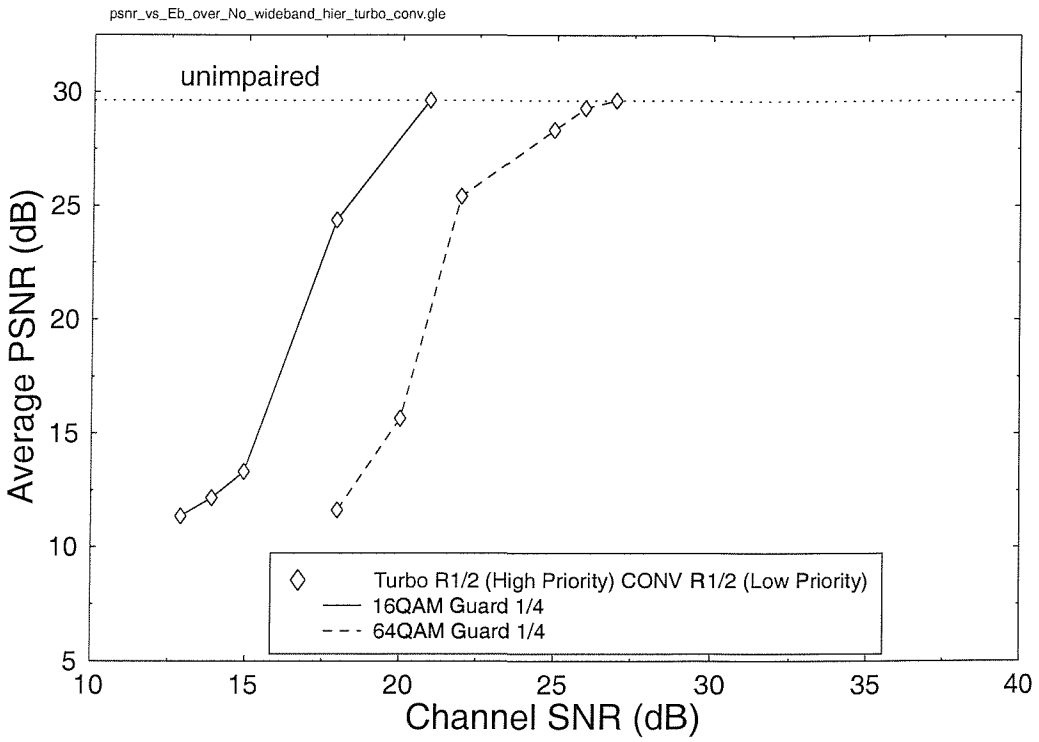


Figure 2.31: Average PSNR versus channel SNR of the DVB scheme, employing turbo coding in the high-priority partition and convolutional coding in the low-priority partition, over the **wideband fading channel** of Figure 2.11 for **hierarchical transmission** using the schematic of Figure 2.13 as well as Algorithms 1 and 2.

Our last combination investigated involved using rate 1/2 turbo coding and convolutional coding for the high- and low-priority partitions, respectively. Comparing Figures 2.31 and 2.30(a), the channel SNR required for achieving unimpaired video transmission in both cases were similar. This was expected, since the turbo-convolutional combination's video performance is dependent on the convolutional code's performance in the low-priority partition.

Lastly, comparing Figures 2.30 and 2.26, we found that the unimpaired PSNR condition was achieved at similar channel SNRs for the hierarchical and non-hierarchical schemes, suggesting that the data partitioning scheme had not provided sufficient performance improvements in the context of the mobile DVB scheme to justify its added complexity. Again, this was a consequence of relegating a high proportion of video bits to the low-integrity partition.

2.9 Conclusions and Future Work

In this contribution, we have investigated the performance of a turbo-coded DVB system in a mobile environment. A range of system performance results was presented based on the standard DVB-T scheme as well as on an improved turbo-coded scheme. The convolutional code specified in the standard system was replaced by turbo coding, which

resulted in a substantial coding gain of around 5 dB. It is important to underline once again that the $K = 3$ turbo code and the $K = 7$ convolutional code exhibited comparable complexities. The higher performance of the turbo codec facilitates, for example, the employment of turbo-coded 16-QAM at a similar SNR, where convolutional-coded QPSK can be invoked. This in turn allows us to double the video bitrate within the same bandwidth and hence to improve the video quality. We have also applied data partitioning to the MPEG-2 video stream to gauge its efficiency in increasing the error resilience of the video codec. However, from these experiments we found that the data partitioning scheme did not provide substantial improvements compared to the non-partitioned video transmitted over the non-hierarchical DVB-T system. We have also experimented with trellis coded modulation (TCM) and turbo trellis coded modulation (TTCM) based OFDM. The performance of the TCM- and TTCM-OFDM systems relative to the DVB-T scheme is presented in Chapter 4.

Our future work will be focused on extending this DVB-T system study to incorporate various types of channel models, as well as on investigating the effects of different Doppler frequencies on the system. The impact of employing various types of turbo interleavers on the system performance is also of interest. A range of further wireless video communications issues are addressed in [42, 185].

Chapter 3

Satellite Based Turbo-coded, Blind-Equalised Single-Carrier 4-QAM and 16-QAM Digital Video Broadcasting

3.1 Background and Motivation

In recent years three harmonised Digital Video Broadcasting (DVB) standards have emerged in Europe for terrestrial [13], cable-based [16] and satellite-oriented [12] delivery of DVB signals. The dispersive wireless propagation environment of the terrestrial system requires concatenated Reed-Solomon [23, 170] (RS) and Rate Compatible Punctured Convolutional Coding [23, 170] (RCPCC) combined with Orthogonal Frequency Division Multiplexing (OFDM) based modulation [100]. The satellite-based system employs the same concatenated channel coding arrangement, as the terrestrial scheme, while the cable-based system refrains from using concatenated channel coding, opting for RS coding only. The performance of both of the satellite and the cable based schemes can be improved upon invoking blind-equalised multi-level modems [100], although the associated mild dispersion or linear distortion does not necessarily require channel equalisation. However, since we propose invoking turbo-coded 4-bit/symbol 16-level Quadrature Amplitude Modulation (16-QAM) in order to improve the system's performance at the cost of increased complexity, in this contribution we additionally invoked blind channel equalisers. This is further justified by the associated high video transmission rates, where the dispersion may become a more dominant performance limitation.

Lastly, the video codec used in all three systems is the Motion Pictures Expert Group's MPEG-2 codec. These standardisation activities were followed by a variety of system performance studies in the open literature [186–189]. Against this background, in this chapter we suggest turbo-coding based improvements to the satellite-based DVB system [12] and present performance studies of the proposed system under dispersive channel conditions in conjunction with a variety of blind channel equalisation algorithms. The transmitted

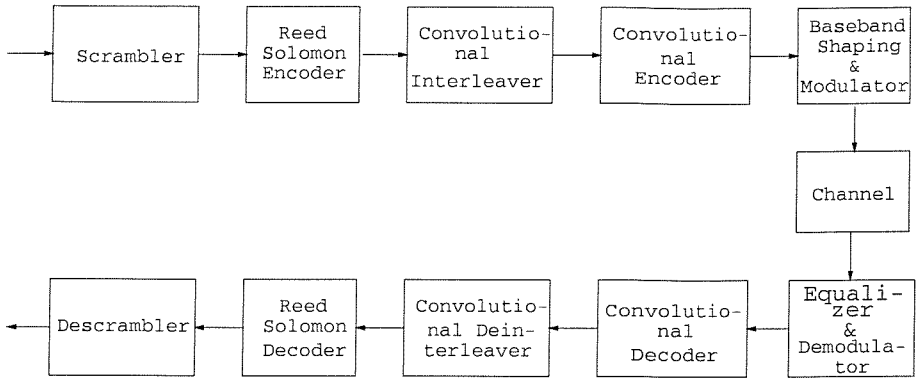


Figure 3.1: Schematic of the DVB satellite system.

power requirements of the standard system employing convolutional codecs can be reduced upon invoking more complex, but more powerful turbo codecs. Alternatively, the standard quaternary or 2-bit/symbol system's Bit Error Rate (BER) versus Signal-to-Noise Ratio (SNR) performance can almost be matched by a turbo-coded 4-bit/symbol 16-QAM based scheme, while doubling the achievable bitrate within the same bandwidth and hence improving the associated video quality. This is achieved at the cost of an increased system complexity.

The remainder of the chapter is organised as follows. A succinct overview of the turbo-coded and standard DVB satellite scheme is presented in Section 3.2, while our channel model is described in Section 3.3. A brief summary of the blind equaliser algorithms employed is presented in Section 3.4. Following this the performance of the improved DVB satellite system is examined for transmission over a dispersive two-path channel in Section 3.5, before our conclusions and future work areas are presented in Section 3.6.

3.2 DVB Satellite Scheme

The block diagram of the DVB satellite (DVB-S) system [12] is shown in Figure 3.1, which is composed of a MPEG-2 video encoder (not shown in the diagram), channel coding modules and a Quadrature Phase Shift Keying (QPSK) modem [100]. The bitstream generated by the MPEG-2 encoder is packetised into frames of 188-byte long. The video data in each packet is then randomized by the scrambler. The details concerning the scrambler have not been included in this paper, since these may be obtained from the DVB-S standard [12].

Due to the poor error resilience of the MPEG-2 video codec, powerful concatenated channel coding is employed. The concatenated channel codec comprises a shortened Reed-Solomon (RS) outer code and an inner convolutional encoder. The 188-byte MPEG-2 video packet is extended by the Reed-Solomon encoder [23, 170] with parity information to facilitate error recovery to form a 204-byte packet. The Reed-Solomon decoder can then correct up to eight erroneous bytes for each 204-byte packet. Following this, the RS-coded packet is interleaved by a convolutional interleaver and further protected by a half-rate inner convolutional encoder with a constraint length of 7 [23, 170].

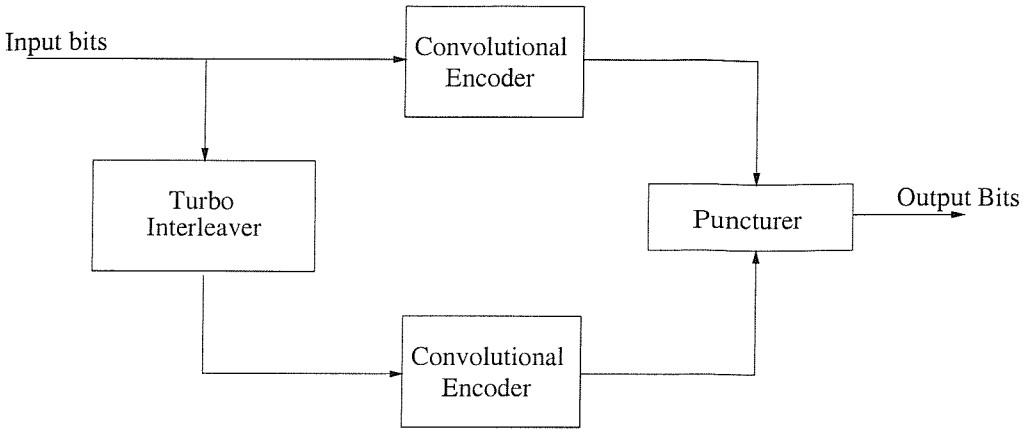


Figure 3.2: Block diagram of turbo encoder.

Furthermore, the overall code rate of the concatenated coding scheme can be adapted by variable puncturing, not shown in the figure, which supports code rates of 1/2 (no puncturing) as well as 2/3, 3/4, 5/6 and 7/8. The parameters of the convolutional encoder are summarised in Table 3.1.

Convolutional Coder Parameters	
Code Rate	1/2
Constraint Length	7
n	2
k	1
Generator Polynomials (octal format)	171, 133

Table 3.1: Parameters of the CC(n,k,K) convolutional inner encoder of the DVB-S modem.

In addition to implementing the standard DVB-S system as a benchmark, we have improved the system's performance with the aid of a turbo codec [156, 178]. The block diagram of the turbo encoder is shown in Figure 3.2. The turbo encoder is constructed of two component encoders. Each component encoder is a half-rate convolutional encoder, whose parameters are listed in Table 3.2. The two component encoders are used to encode the same input bits, although the input bits of the second component encoder are interleaved before encoding. The output bits of the two component codes are punctured and multiplexed, in order to form a single output bitstream. The component encoder used here is known as a half-rate Recursive Systematic Convolutional (RSC) encoder [179]. It generates one parity bit and one systematic output bit for every input bit. In order to provide an overall coding rate of one half, half the output bits from the two encoders must be punctured. The puncturing arrangement used in our work is to transmit all the systematic bits from the first encoder and every other parity bit from both encoders.

Readers interested in further details of the DVB-S system are referred to the DVB-S standard [12]. The performance of the standard DVB-S system and that of the turbo-coded system is characterised in Section 3.5. Let us now briefly consider the multipath channel model used in our investigations.

Turbo Coder Parameters	
Turbo Code Rate	1/2
Input block length	17952 bits
Interleaver Type	Random
Number of turbo decoder iterations	8
Turbo Encoder Component Code Parameters	
Component Code Encoder Type	Recursive Systematic Convolutional (RSC)
Component Code Decoder Type	Log-MAP [181]
Constraint Length	3
n	2
k	1
Generator Polynomials (octal format)	7, 5

Table 3.2: Parameters of the inner turbo encoder used to replace the DVB-S system's convolutional coder (RSC: recursive systematic code).

3.3 Channel Model

The DVB-S system was designed to operate in the 12 GHz frequency band (K-band). Within this frequency band, tropospheric effects such as the transformation of electromagnetic energy into thermal energy due to induction of currents in rain and ice crystals lead to signal attenuations [101, 190]. In the past 20 years, various researchers have concentrated their efforts on attempting to model the satellite channel, typically within a land mobile satellite channel scenario. However, the majority of the work conducted for example by Vogel and his colleagues [191–194] concentrated on modelling the statistical properties of a narrowband satellite channel in lower frequency bands, such as the 870 MHz UHF band and the 1.5 GHz L-band.

However, our high bitrate DVB satellite system requires a high bandwidth, hence the video bitstream is exposed to dispersive wideband propagation conditions. Recently, Saunders *et al.* [195, 196] have proposed the employment of multipath channel models to study the satellite channel, although their study was concentrated on the L-band and S-band only.

Due to the dearth of reported work on wideband satellite channel modelling in the K-band, we have adopted a simpler approach. The channel model employed in this study was the two-path (nT)-symbol spaced impulse response, where T is the symbol-duration. In our studies we used $n = 1$ and $n = 2$. This corresponds to a stationary dispersive transmission channel. Our channel model assumed that the receiver had a direct line-of-sight with the satellite as well as a second path caused by a single reflector probably from a nearby building or due to ground reflection. The ground reflection may be strong, if the satellite receiver dish is only tilted at a low angle.

Based on these channel models, we studied the ability of a range of blind equaliser algorithms to converge under various path delay conditions. In the next section we provide a brief overview of the various blind equalisers employed in our experiments, noting that

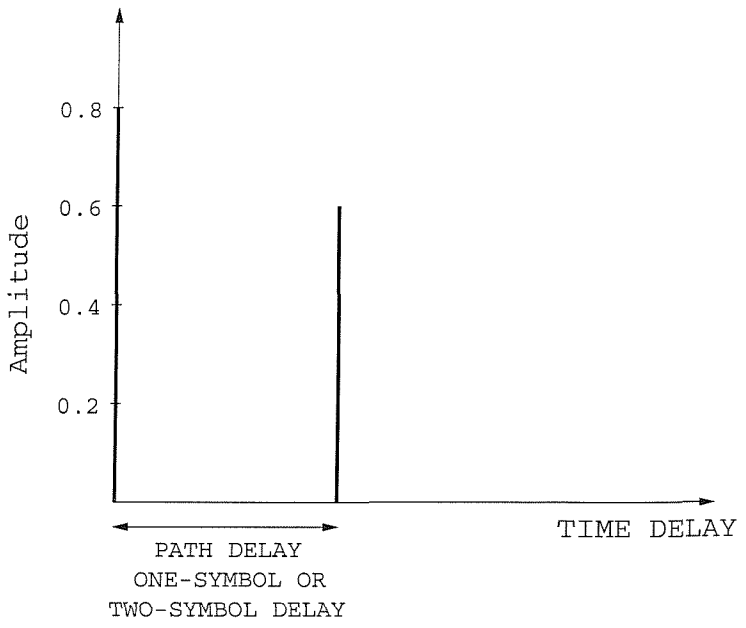


Figure 3.3: Two-path satellite channel model with either a one-symbol or two-symbol delay.

the readers who are mainly interested in the system's performance may proceed directly to our performance analysis section, namely to Section 3.5.

3.4 The Blind Equalisers

In this section the blind equalisers used in the system are presented. The following blind equalisers have been studied:

- The Modified Constant Modulus Algorithm (MCMA) [197]
- The Benveniste-Goursat Algorithm (B-G) [198]
- The Stop-and-Go Algorithm (S-a-G) [199]
- The Per-Survivor Processing (PSP) Algorithm [200].

We will now briefly introduce these algorithms.

First we define the variables that we will use:

$$\mathbf{y}(n) = [y(n+N_1), \ldots, y(0), \ldots, y(n-N_2)]^T \tag{3.1}$$

$$\mathbf{c}^{(n)} = [c_{-N_1}, \ldots, c_o, \ldots, c_{N_2}]^T \tag{3.2}$$

$$z(n) = \left(\mathbf{c}^{(n)}\right)^T \mathbf{y}(n) = \mathbf{y}^T(n) \cdot \mathbf{c}^{(n)} \tag{3.3}$$

where $\mathbf{y}(n)$ is the received symbol vector at time n, containing the $N_1 + N_2 + 1$ most recent received symbols, while N_1, N_2 are the number of equaliser feedback and feedforward taps, respectively. Furthermore, $\mathbf{c}^{(n)}$ is the equaliser tap-vector, consisting of the equaliser tap values and $z(n)$ is the equalised symbol at time n, given by the convolution of the received signal with the equaliser's impulse response, while $()^T$ stands for matrix transpose. Note that the variables of Equations 3.1–3.3 assume complex values, when multi-level modulation is employed.

The **Modified CMA** (MCMA) is an improved version of Godard's well-known *Constant Modulus Algorithm (CMA)* [201]. The philosophy of the CMA is based on forcing the magnitude of the equalised signal to a constant value. In mathematical terms the CMA is based on minimising the cost function:

$$J^{(CMA)} = E\left[\left(|z(n)|^2 - R_2\right)^2\right], \tag{3.4}$$

where R_2 is a suitably chosen constant and $E[]$ stands for the expectation. Similarly to the CMA, the MCMA, which was proposed by Wesolowsky [197], forces the real and imaginary parts of the complex signal to the constant values of $R_{2,R}$ and $R_{2,I}$ respectively, according to the equaliser tap update equation of [197]:

$$\begin{aligned}\mathbf{c}^{(n+1)} = \mathbf{c}^{(n)} - \lambda \cdot \mathbf{y}^*(n)\{&Re[z(n)]\left((Re[z(n)])^2 - R_{2,R}\right) + \\ &+ j \cdot Im[z(n)]\left((Im[z(n)])^2 - R_{2,I}\right)\},\end{aligned} \tag{3.5}$$

where λ is the step–size parameter and the $R_{2,R}$, $R_{2,I}$ constant parameters of the algorithm are defined as:

$$R_{2,R} = \frac{E\left[(Re[a(n)])^4\right]}{E\left[(Re[a(n)])^2\right]} \tag{3.6}$$

$$R_{2,I} = \frac{E\left[(Im[a(n)])^4\right]}{E\left[(Im[a(n)])^2\right]}, \tag{3.7}$$

where $a(n)$ is the transmitted signal at time instant n.

The **Benveniste-Goursat** (B-G) algorithm [198] is an amalgam of the Sato's algorithm [202] and the decision-directed (DD) algorithm [100]. Strictly speaking, the decision-directed algorithm is not a blind equalisation technique, since its convergence is highly dependent on the channel.

This algorithm estimates the error between the equalised signal and the detected signal as:

$$\epsilon^{DD}(n) = z(n) - \hat{z}(n), \tag{3.8}$$

where $\hat{z}(n)$ is the receiver's estimate of the transmitted signal at time instant n. Similarly to the DD algorithm's error term, the Sato-type error [202] is defined as:

$$\epsilon^{Sato}(n) = z(n) - \gamma \cdot csgn(z(n)), \tag{3.9}$$

where γ is a constant parameter of the Sato-algorithm, defined as:

$$\gamma = \frac{E\left[(Re\left[a(n)\right])^2\right]}{E\left[|Re\left[a(n)\right]|\right]} = \frac{E\left[(Im\left[a(n)\right])^2\right]}{E\left[|Im\left[a(n)\right]|\right]} \tag{3.10}$$

and $csgn(x) = sign(Re\{x\}) + j \cdot sign(Im\{x\})$ is the complex sign function. The B-G algorithm combines the above two error terms into one:

$$\epsilon^G(n) = k_1 \cdot \epsilon^{DD}(n) + k_2 \cdot |\epsilon^{DD}(n)| \cdot \epsilon^{Sato}(n), \tag{3.11}$$

where the two error terms are suitably weighted by the constant parameters k_1 and k_2 in Equation (3.11). Using this error term, the B-G equaliser updates the equaliser coefficients according to the the following equaliser tap update equations [198]:

$$\mathbf{c}^{(n+1)} = \mathbf{c}^{(n)} - \lambda \cdot \mathbf{y}^*(n) \cdot \epsilon^G(n). \tag{3.12}$$

In our investigations, the weights were chosen as $k_1 = 1$, $k_2 = 5$, so that the Sato-error was weighted more heavily, than the DD-error.

The **Stop-and-Go** (S-a-G) algorithm [199] is a variant of the decision-directed algorithm [100], where at each equaliser coefficient adjustment iteration the update is enabled or disabled, depending on whether the update is likely to be correct. The update equations of this algorithm are given by [199]:

$$\mathbf{c}^{(n+1)} = \mathbf{c}^{(n)} - \lambda \cdot \mathbf{y}^*(n)[f_{n,R} Re\{\epsilon^{DD}(n)\} + j f_{n,I} Im\{\epsilon^{DD}(n)\}] \tag{3.13}$$

where * stands for the complex conjugate, $\epsilon^{DD}(n)$ is the decision-directed error as in Equation (3.8) and the binary functions $f_{n,R}$, $f_{n,I}$ enable or disable the update of the equaliser according to the following rule. If the sign of the Sato-error (the real or the imaginary part independently) is the same as the sign of the decision-directed error, then the update takes place, otherwise it does not.

In mathematical terms, this is equivalent to [199]:

$$f_{n,R} = \begin{cases} 1 & \text{if } sgn(Re[\epsilon^{DD}(n)]) = sgn(Re[\epsilon^{Sato}(n)]) \\ 0 & \text{if } sgn(Re[\epsilon^{DD}(n)] \neq sgn(Re[\epsilon^{Sato}(n)]) \end{cases} \tag{3.14}$$

$$f_{n,I} = \begin{cases} 1 & \text{if } sgn(Im\{\epsilon^{DD}(n)\}) = sgn(Im\{\epsilon^{Sato}(n)\}) \\ 0 & \text{if } sgn(Im\{\epsilon^{DD}(n)\}) \neq sgn(Im\{\epsilon^{Sato}(n)\}). \end{cases} \tag{3.15}$$

For a blind equaliser, this condition provides us with a measure of the probability of the coefficient update being correct.

The **PSP algorithm** [200] is based on employing convolutional coding and hence it is a trellis-based sequence estimation technique, in which the channel is not known "a priori". Hence, an iterative channel estimation technique is employed, in order to estimate the channel jointly with the modulation symbol. In this sense, an initial channel estimation is used and the estimation is updated at each new symbol's arrival.

In our case the update was based on the *Least Mean Squares (LMS)* estimates, according to the following channel-tap update equations [200]:

$$\hat{\mathbf{h}}^{(n+1)} = \hat{\mathbf{h}}^{(n)} + \lambda \cdot \hat{\mathbf{a}}^*(n) \cdot \left(y(n) - \hat{\mathbf{a}}^T(n) \cdot \hat{\mathbf{h}}^{(n)}\right), \tag{3.16}$$

where $\hat{\mathbf{h}}^{(n)} = (\hat{h}_{-L_1}^{(n)}, \ldots, \hat{h}_o^{(n)}, \ldots, \hat{h}_{L_2}^{(n)})^T$ is the estimated (for one surviving path) channel tap-vector at time instant n, $\hat{\mathbf{a}}(n) = (\hat{a}(n+L_1), \ldots, \hat{a}(0), \ldots, \hat{a}(n-L_2))^T$ is the associated estimated transmitted symbol vector and $y(n)$ is the actually received symbol at time instant n.

Each of the surviving paths in the trellis carries not only its own signal estimation, but also its own channel estimation. Moreover, convolutional decoding can take place jointly with this channel and data estimation procedure, leading to improved Bit Error Rate (BER) performance. A summary of the various equalisers' parameters is given in Table 3.3.

	Step–size λ	No. of Equal. Taps	Initial Tap–Vector
Benveniste-Goursat	$5\text{x}10^{-4}$	10	$(1.2, 0, \cdots, 0)$
Modified-CMA	$5\text{x}10^{-4}$	10	$(1.2, 0, \cdots, 0)$
Stop-and-Go	$5\text{x}10^{-4}$	10	$(1.2, 0, \cdots, 0)$
PSP (1 sym delay)	10^{-2}	2	$(1.2, 0)$
PSP (2 sym delay)	10^{-2}	3	$(1.2, 0, 0)$

Table 3.3: Summary of the equaliser parameters used in the simulations. The tap–vector $(1.2, 0, \cdots, 0)$ indicates that the first equaliser coefficient is initialised to the value 1.2, while the others to 0

Having described the components of our enhanced DVB-S system, let us now consider the overall system's performance.

3.5 Performance of the DVB-S Scheme

In this section, the performance of the DVB-S system was evaluated by means of simulations. Two modulation types were used, namely the standard QPSK and the enhanced 16-QAM schemes [100]. The channel model of Figure 3.3 was employed. The first channel model had a one-symbol second-path delay, while in the second one the path-delay corresponded to the period of two symbols. The average BER versus SNR per bit performance was evaluated after the equalisation and demodulation process, as well as after Viterbi [23] or turbo decoding [156]. The SNR per bit or E_b/N_o is defined as follows:

$$\text{SNR per bit} = 10log_{10}\frac{\bar{S}}{\bar{N}} + \delta \tag{3.17}$$

where $\bar{S}$ is the average received signal power, $\bar{N}$ is the average received noise power and δ, which is dependent on the type of modulation scheme used and channel code rate (R), is defined as follows:

$$\delta = 10log_{10}\frac{1}{\text{R} \times \text{Bits per modulation symbol}}. \tag{3.18}$$

Our results are further divided into two subsections for ease of discussion. First, we will present the system performance over the one-symbol delay two-path channel in Section 3.5.1. Next, the system performance over the two-symbol delay two-path channel is

presented in Section 3.5.2. Lastly, a summary of the system performance is provided in Section 3.5.3.

3.5.1 Transmission Over the Symbol-spaced Two-path Channel

The linear equalisers' performance was quantified and compared using QPSK modulation over the one-symbol delay two-path channel model of Figure 3.4. Since all the equalisers' BER performance was similar, only the Modified CMA results are shown in the figure.

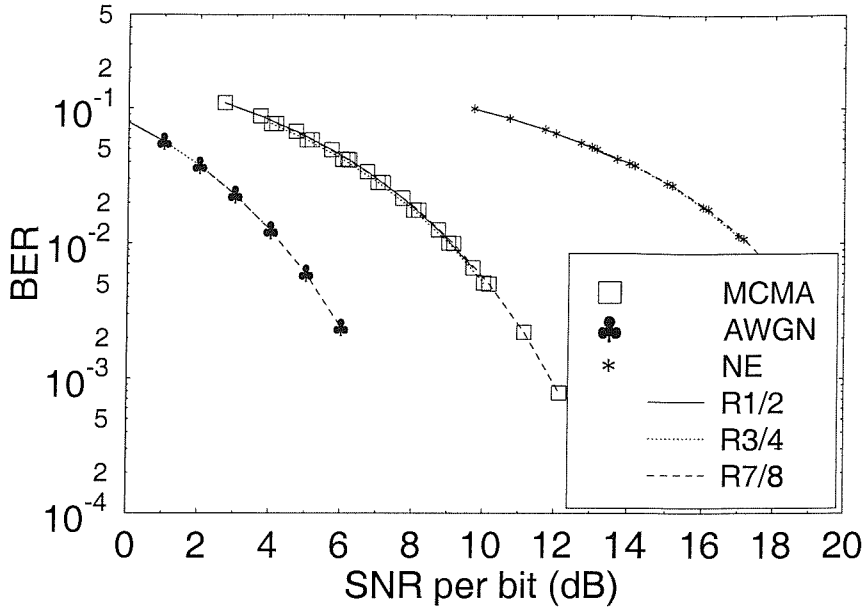
The equalised performance over the one symbol-spaced channel was inferior to that over the non-dispersive AWGN channel. However, as expected, it was better than without any equalisation. Another observation for Figure 3.4 was that the different punctured channel coding rates appeared to give slightly different bit error rates after equalisation. This was because the linear blind equalisers required uncorrelated input bits in order to converge. However, the input bits were not entirely random, when convolutional coding was used. The consequences of violating the zero-correlation constraint are not generally known. Nevertheless, two potential problems were apparent. Firstly, the equaliser may diverge from the desired equaliser equilibrium [203]. Secondly, the performance of the equaliser is expected to degrade, owing to the violation of the randomness requirement, which is imposed on the input bits in order to ensure that the blind equalisers will converge.

Since the channel used in our investigations was static, the first problem was not encountered. Instead, the second problem was what we actually observed. Figure 3.5 quantifies the equalisers' performance degradation due to convolutional coding. We can observe a 0.1 dB SNR degradation, when the convolutional codec creates correlation among the bits for this specific case.

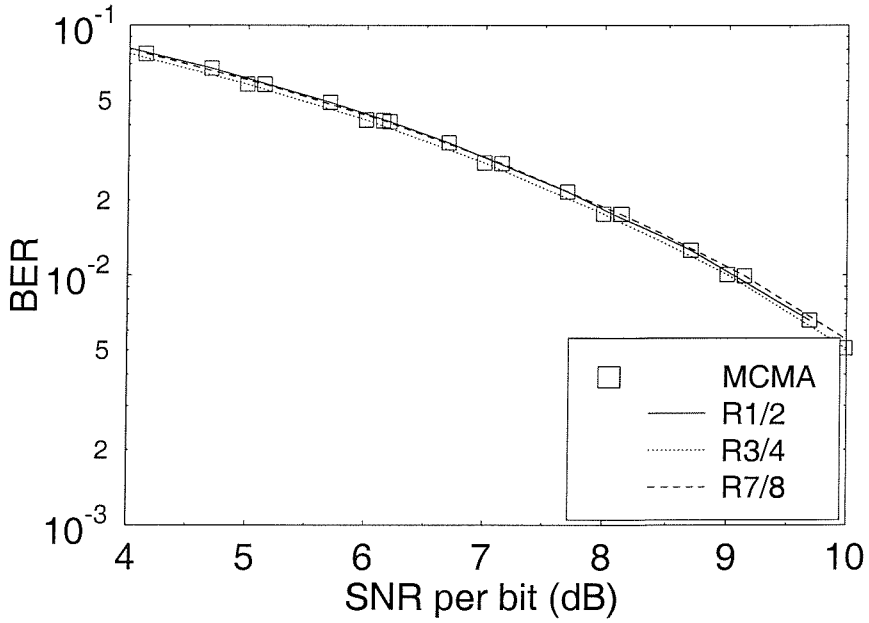
The average BER curves after Viterbi or turbo decoding are shown in Figure 3.6(a). In this figure, the average BER over the non-dispersive AWGN channel after turbo decoding constitutes the best case performance, while the average BER of the one-symbol delay two-path MCMA-equalised rate 7/8 convolutionally coded scenario exhibits the worst case performance. Again, in this figure only the Modified-CMA was featured for simplicity. The performance of the remaining equalisers was characterised in Figure 3.6(b). Clearly, the performance of all the linear equalisers investigated was similar.

It is observed in Figure 3.6(a) that the combination of the Modified CMA blind equaliser with turbo decoding exhibited the best SNR performance over the one-symbol delay two-path channel. The only comparable alternative was the PSP algorithm. Although the performance of the PSP algorithm was better at low SNRs, the associated curves cross over and the PSP algorithm's performance became inferior below the average BER of 10^{-3}. Although not shown in Figure 3.6, the Reed-Solomon decoder, which was concatenated to either the convolutional or the turbo decoder, became effective, when the average BER of its input was below approximately 10^{-4}. In this case, the PSP algorithm performed by at least 1 dB worse in the area of interest, which is at an average BER of 10^{-4}.

A final observation in the context of Figure 3.6(a) is that when convolutional decoding was used, the associated E_b/N_o performance of the rate 1/2 convolutional-coded scheme



(a) After equalisation and demodulation



(b) Same as (a) but enlarged in order to show performance difference of the blind equaliser, when different convolutional code rates are used.

Figure 3.4: Average BER versus SNR per bit performance after equalisation and demodulation employing **QPSK** modulation and **one-symbol delay channel** (**NE**: Non-Equalised; **MCMA**: Modified Constant Modulus Algorithm).

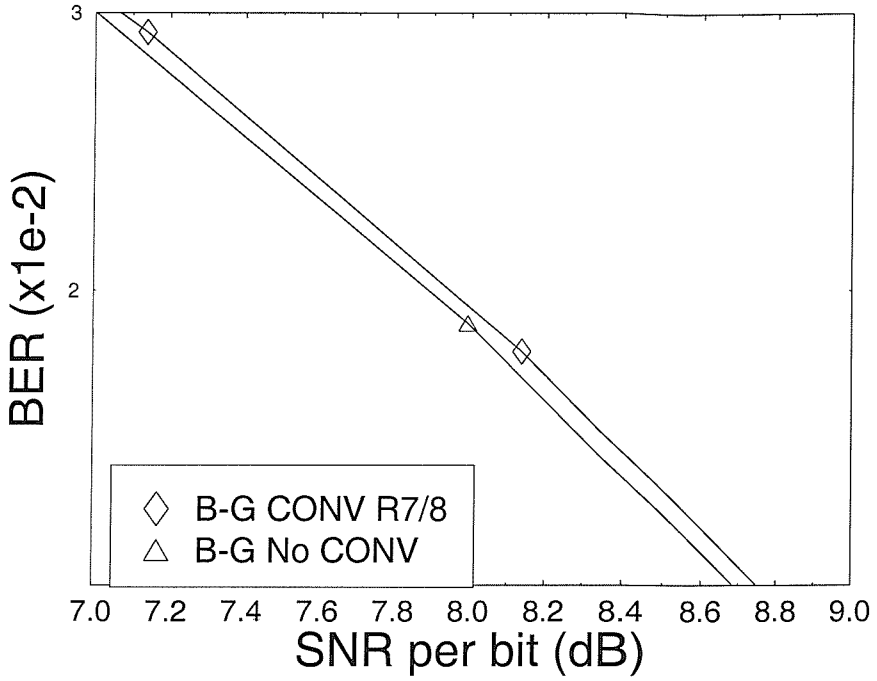
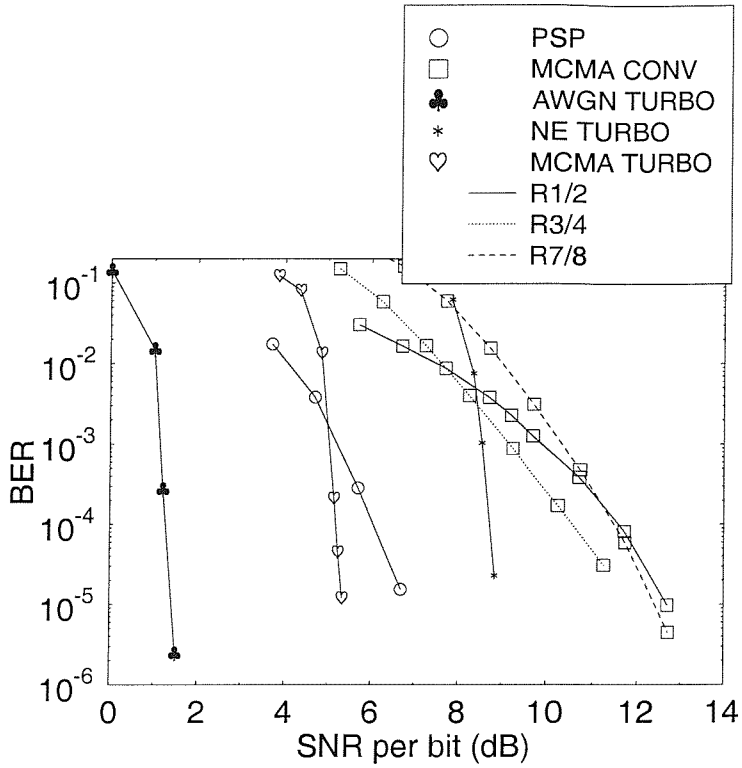


Figure 3.5: Average BER versus SNR per bit performance after equalisation and demodulation employing **QPSK** modulation and the **one-symbol delay two-path channel** of Figure 3.3, for the Benveniste-Goursat algorithm, where the input bits are random (No CONV) or correlated (CONV 7/8) as a result of convolutional coding having a coding rate of 7/8.

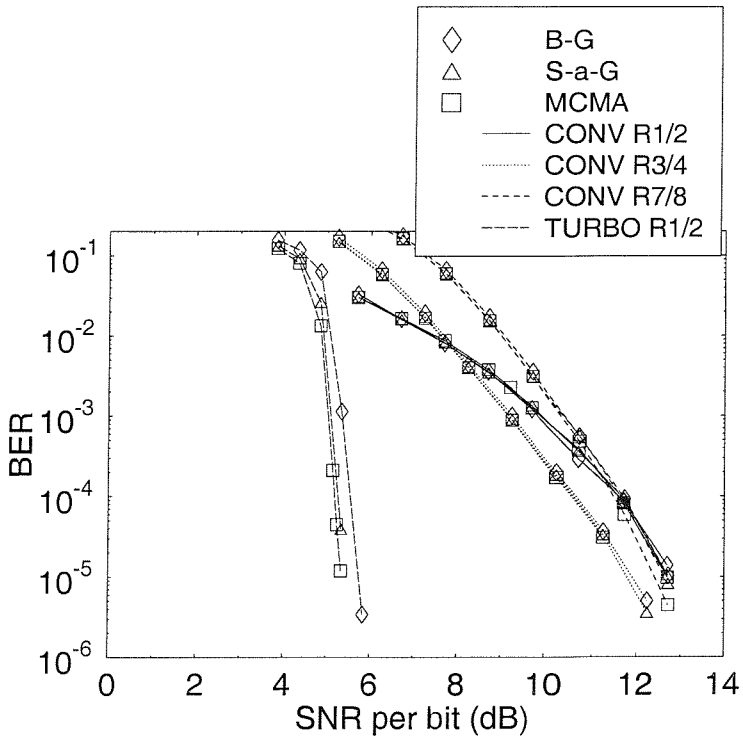
appeared slightly inferior to that of the rate 3/4 and the rate 7/8 scenarios beyond certain E_b/N_o values. This was deemed to be a consequence of the fact that the 1/2-rate encoder introduced more correlation into the bitstream than its higher rate counterparts and this degraded the performance of the blind channel equalisers, which performed best, when fed with random bits.

Having considered the QPSK case, we shall now concentrate on the enhanced system, which employed 16-QAM under the same channel and equaliser conditions. In Figure 3.7 and Figure 3.8, the performance of the DVB system employing 16-QAM is presented. Again, for simplicity, only the Modified CMA results are given. In this case the ranking order of the different coding rates followed our expectations more closely in the sense that the lowest coding rate of 1/2 was the best performer, followed by rate 3/4 codec, in turn followed by the least powerful rate 7/8 codec.

The Stop-and-Go algorithm has been excluded from these results, since it does not converge for high SNR values. This happens, because the equalisation procedure is only activated, when there is a high probability of correct decision-directed equaliser update. In our case, the equaliser is initialised far from its convergence point and hence the decision–directed updates are unlikely to be correct. In the absence of noise this leads to the update algorithm being permanently de-activated. If noise is present though, then some random perturbations from the point of the equaliser's initialization can activate

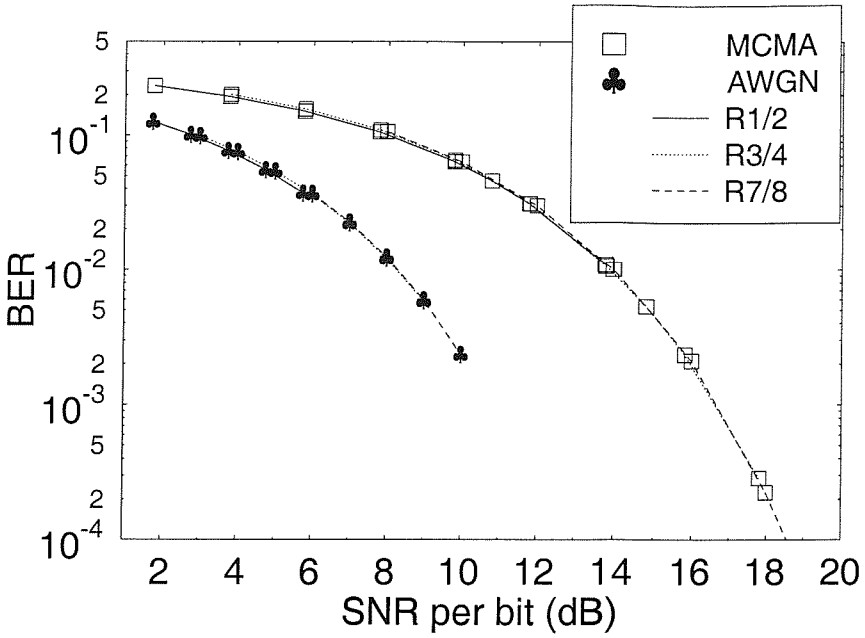


(a) PSP and linear equalisers

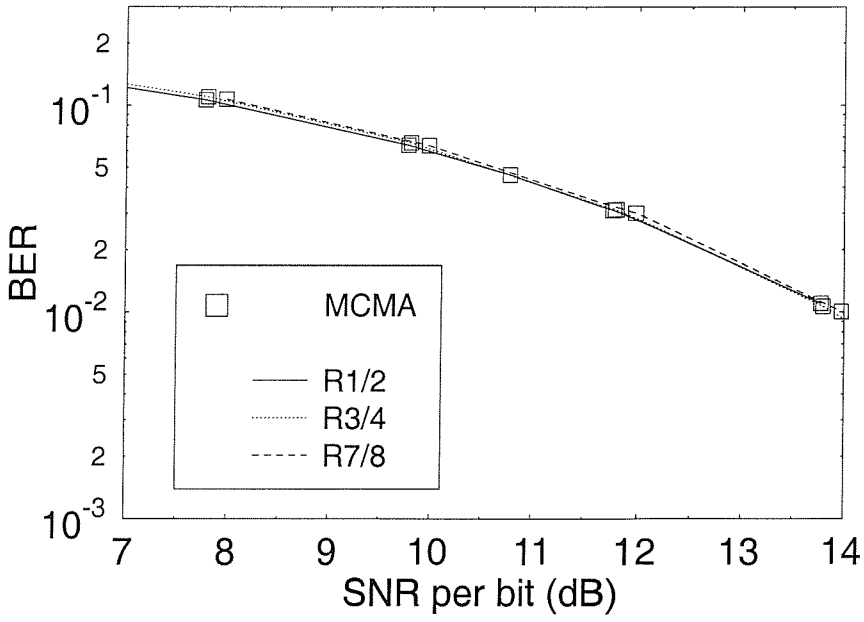


(b) Linear equalisers only

Figure 3.6: Average BER versus SNR per bit performance after convolutional or turbo decoding for **QPSK** modulation and **one-symbol delay channel** (**NE**: Non-Equalised; **B-G**: Benveniste-Goursat; **S-a-G**: Stop-and-Go; **MCMA**: Modified Constant Modulus Algorithm; **PSP**: Per-Survivor-Processing).



(a) After equalisation and demodulation



(b) Same as (a) but enlarged in order to show performance difference of the blind equaliser, when different convolutional code rates are used.

Figure 3.7: Average BER versus SNR per bit after equalisation and demodulation for **16-QAM** over the **one-symbol delay two-path channel** of Figure 3.3 (**MCMA**: Modified Constant Modulus Algorithm).

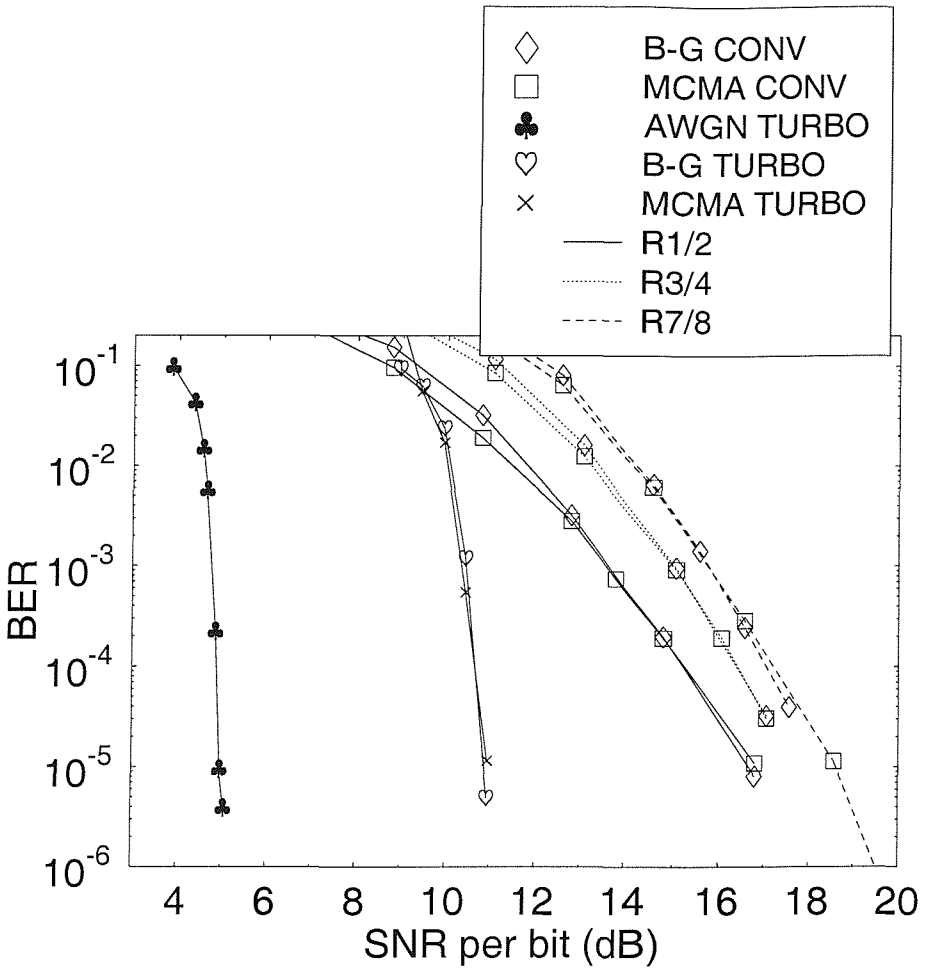


Figure 3.8: Average BER versus SNR per bit after Viterbi or turbo decoding for **16-QAM** over the **one-symbol delay two-path channel** of Figure 3.3 (**B-G**: Benveniste-Goursat; **S-a-G**: Stop-and-Go; **MCMA**: Modified Constant Modulus Algorithm; **PSP**: Per-Survivor-Processing).

the Stop-and-Go algorithm and can lead to convergence. We made this observation at medium SNR values in our simulation study. For high SNR values though, the algorithm did not converge.

It is also interesting to compare the performance of the system for the QPSK and 16-QAM schemes. When the one-symbol delay two-path channel model of Figure 3.3 was considered, the system was capable of supporting the use of 16-QAM with the provision of an additional SNR per bit of approximately 4–5 dB. This observation was made by comparing the performance of the DVB system when employing the Modified CMA and the half-rate convolutional or turbo code in Figure 3.6 and Figure 3.8 at a BER of 10^{-4}. Although the original DVB-Satellite system only employs QPSK modulation, our simulations had shown that 16-QAM can be employed equally well for the range of blind equalisers that we have used in our work. This allowed us to double the video bitrate and hence to substantially improve the video quality. **The comparison of Figures 3.6 and 3.8 also reveals that the extra SNR requirement of approximately 4–5 dB of 16-QAM over QPSK can be eliminated by employing turbo coding at the cost**

of a higher implementational complexity. This allowed us to accommodate a doubled bitrate within a given bandwidth, which improved the video quality.

3.5.2 Transmission Over the Two-symbol Delay Two-path Channel

In Figures 3.9 (only for the Benveniste-Goursat algorithm for simplicity) and 3.10 the corresponding BER results for the two-symbol delay two-path channel of Figure 3.3 are given for QPSK. The associated trends are similar to those in Figures 3.4 and 3.6, although some differences can be observed, as listed below:

- The "cross-over point", beyond which the performance of the PSP algorithm was inferior to that of the Modified CMA in conjunction with turbo decoding is now at 10^{-4}, which is in the range, where the RS decoder guarantees an extremely low probability of error.
- The rate 1/2 convolutional decoding was now the best performer, when convolutional decoding is concerned, while the rate 3/4 scheme exhibited the worst performance.

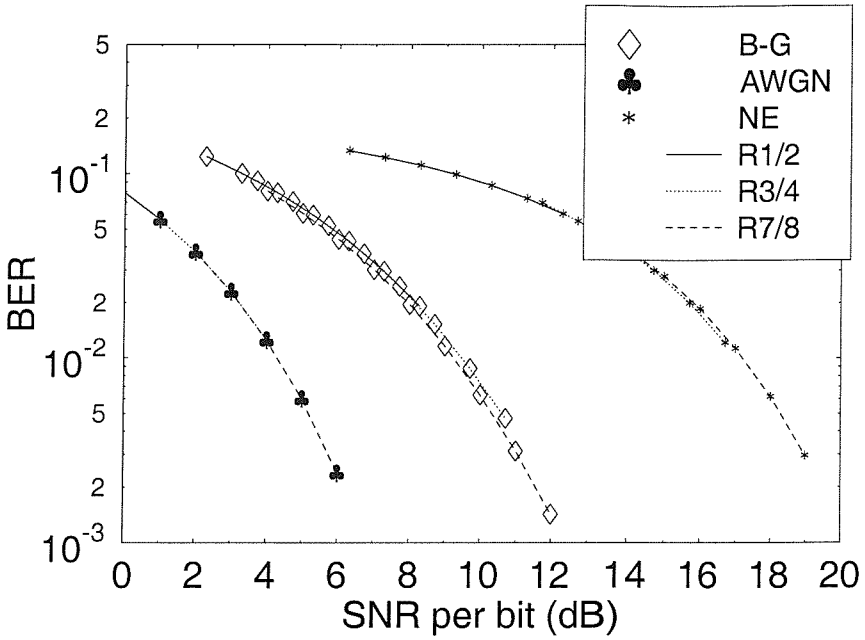
Finally, in Figure 3.11, the associated 16-QAM results are presented. Notice that the Stop-and-Go algorithm was again excluded from the results. Furthermore, we observe a high performance difference between the Benveniste-Goursat algorithm and the Modified CMA. In the previous cases we did not observe such a significant difference. The difference in this case is that the channel exhibits an increased delay spread. This illustrated the capability of the equalisers to cope with more widespread multipaths, while keeping the equalizer order constant at 10. The Benveniste-Goursat equaliser was more efficient, than the Modified CMA in this case.

It is interesting to note that in this case, the performance of the different coding rates was again in the expected order, the rate 1/2 being the best, followed by the rate 3/4 and then the rate 7/8 scheme.

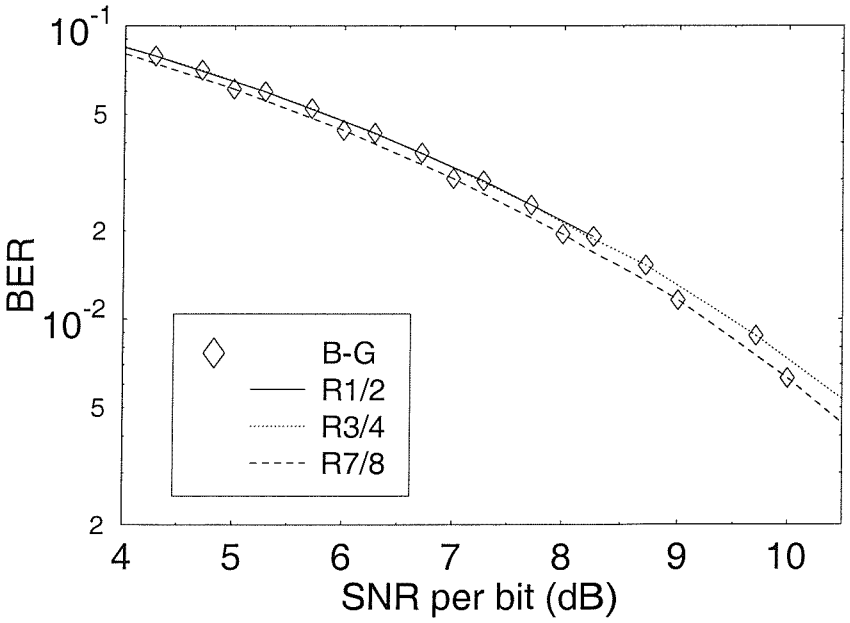
If we compare the performance of the system employing QPSK and 16-QAM over the two-symbol delay two-path channel of Figure 3.3, we again observe that **16-QAM can be incorporated into the DVB system, if an extra 5 dB of SNR per bit is affordable in power budget terms. However, only the B-G algorithm is worthwhile considering here out of the three linear equalisers of Table 3.3.** This observation was made by comparing the performance of the DVB system when employing the Benveniste-Goursat equalizer and the half-rate convolutional coder in Figure 3.10 and Figure 3.11.

3.5.3 Performance Summary of the DVB-S System

Table 3.4 provides an approximation of the convergence speed of each blind equalisation algorithm of Table 3.3. It is clear that PSP exhibited the fastest convergence, followed by the Benveniste-Goursat algorithm. In our simulations the convergence was quantified by observing the slope of the BER curve, and finding when this curve was reaching the associated residual BER, implying that the BER has reached its steady–state value. Figure 3.12 gives an illustrative example of the equaliser's convergence for 16-QAM. The Stop-and-Go algorithm converges significantly slower than the other algorithms, which

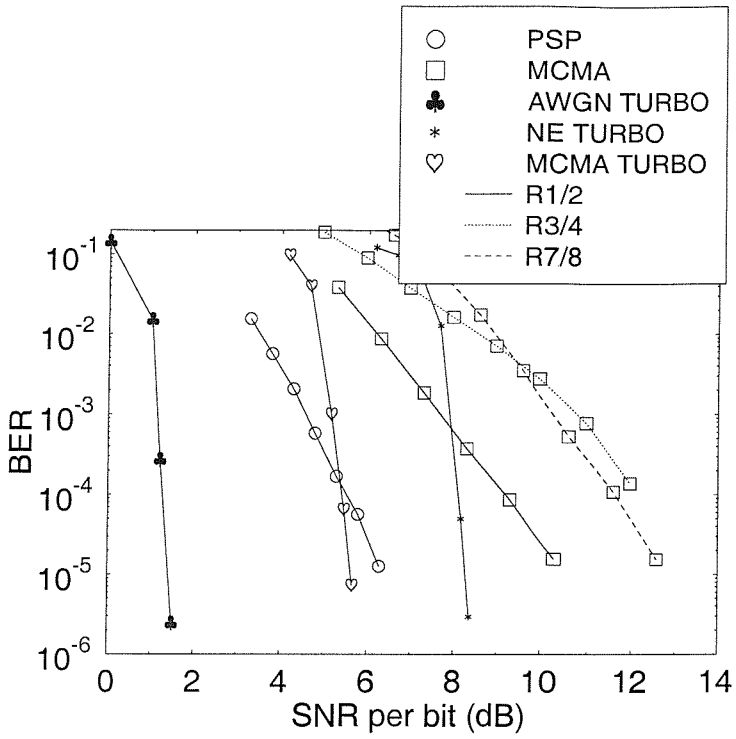


(a) After equalisation and demodulation

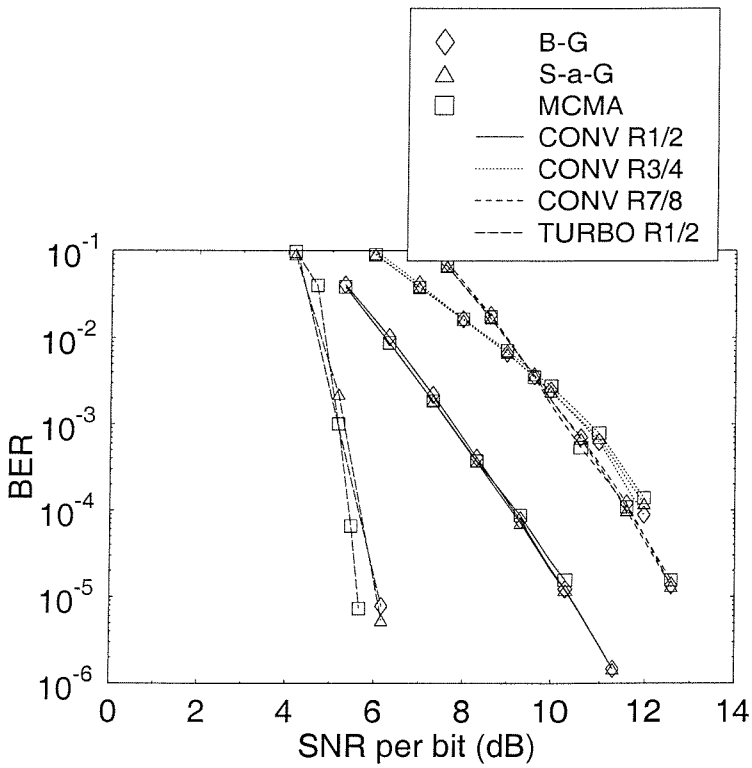


(b) Same as (a) but enlarged in order to show performance difference of the blind equaliser, when different convolutional code rates are used.

Figure 3.9: Average BER versus SNR per bit performance after equalisation and demodulation for **QPSK** modulation over the **two-symbol delay two-path channel** of Figure 3.3 (**B-G**: Benveniste-Goursat).

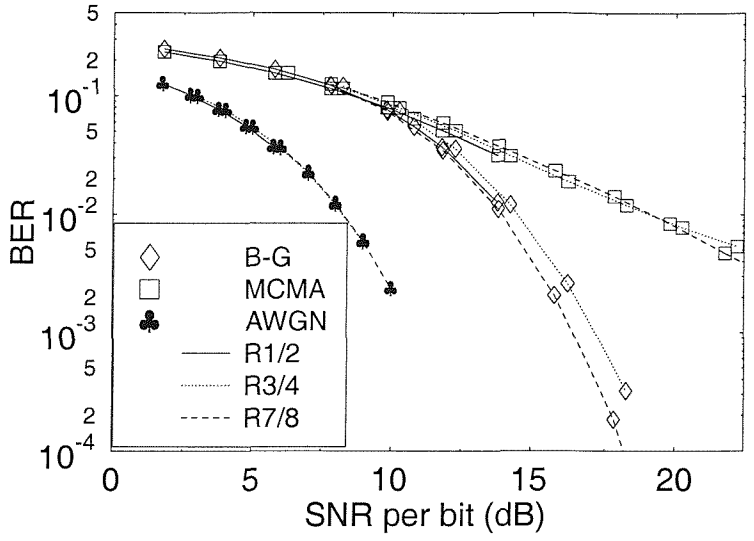


(a) PSP and linear equalisers

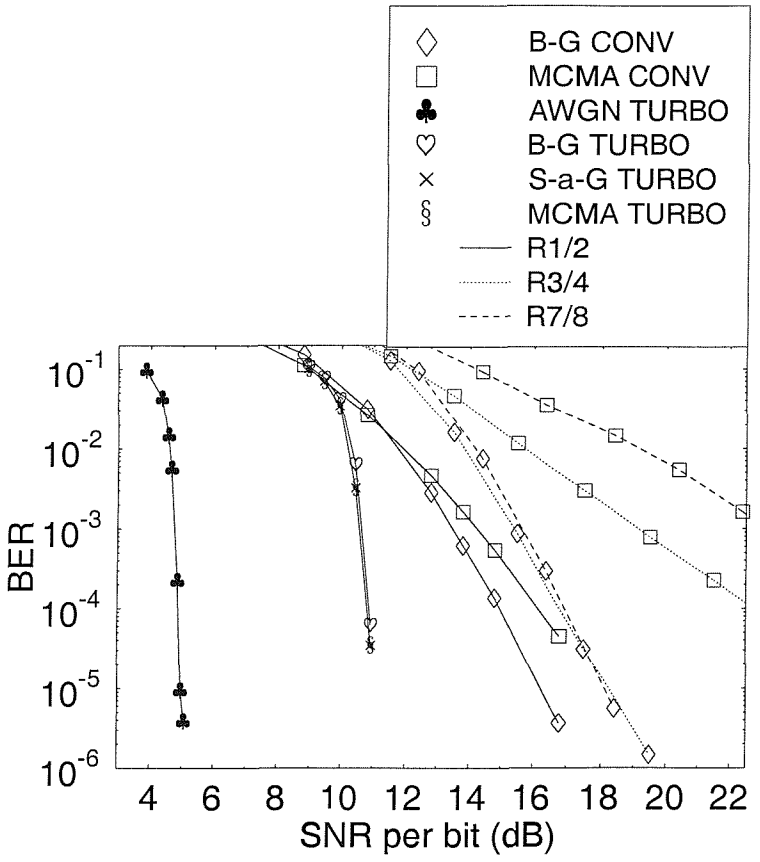


(b) Linear equalisers only

Figure 3.10: Average BER versus SNR per bit performance after convolutional or turbo decoding for **QPSK** modulation over the **two-symbol delay two-path channel** of Figure 3.3 (**B-G**: Benveniste-Goursat; **S-a-G**: Stop-and-Go; **MCMA**: Modified Constant Modulus Algorithm; **PSP**: Per-Survivor-Processing).



(a) After equalisation and demodulation



(b) After viterbi or turbo decoding

Figure 3.11: Average BER versus SNR per bit performance (a) after equalisation and demodulation and (b) after Viterbi or turbo decoding for **16-QAM** over the **two-symbol delay two-path channel** of Figure 3.3 (**B-G**: Benveniste-Goursat; **S-a-G**: Stop-and-Go; **MCMA**: Modified Constant Modulus Algorithm; **PSP**: Per-Survivor-Processing).

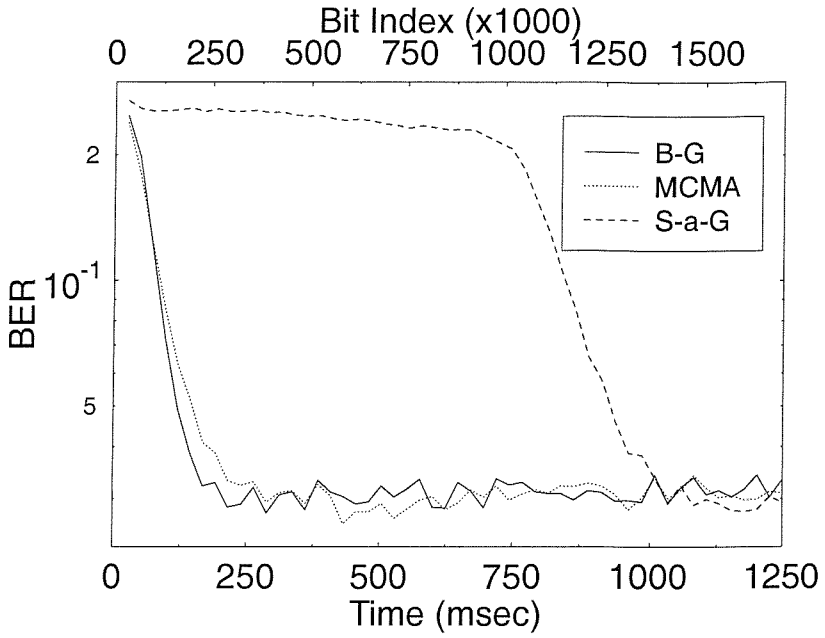


Figure 3.12: Learning curves for 16-QAM, one-symbol delay two-path channel at SNR=18dB.

can also be seen from Table 3.4. This happens because, during the startup, the algorithm is de-activated most of the time, an effect which becomes more severe with an increasing QAM order.

	B-G	MCMA	S-a-G	PSP
QPSK 1 sym	73	161	143	0.139
QPSK 2 sym	73	143	77	0.139
16-QAM 1 sym	411	645	1393	
16-QAM 2 sym	359	411	1320	

Table 3.4: Equaliser convergence speed (in miliseconds) measured in the simulations, given as an estimate of time required for convergence when 1/2 rate puncturing is used (x sym: x-symbol delay two-path channel and x can take either the value 1 or 2).

Figure 3.13 portrays the corresponding reconstructed video quality in terms of the average Peak Signal-to-Noise Ratio (PSNR) versus Channel SNR (CSNR) for the one-symbol delay and two-symbol delay two-path channel model of Figure 3.3. The definition of PSNR which is defined by Equation 2.1 and repeated here for convenience, is defined as follows:

$$\text{PSNR} = 10log_{10}\frac{\sum_{n=0}^{N}\sum_{m=0}^{M}255^2}{\sum_{n=0}^{N}\sum_{m=0}^{M}\Delta^2}, \tag{3.19}$$

where Δ is the difference between the uncoded pixel value and the reconstructed pixel value. The variables M and N refer to the dimension of the image. The maximum possible 8-bit represented pixel luminance value of 255 was used in Equation 3.19 in order to mitigate the PSNR's dependence on the video material used. The average PSNR is

then the mean of the PSNR values computed for all the images constituting the video sequence.

Tables 3.5 and 3.6 provide a summary of the DVB-Satellite system's performance tolerating a PSNR degradation of 2 dB, which was deemed to be nearly imperceptible in terms of subjective video degradations. The average BER values quoted in the tables refer to the average BER achieved after Viterbi or turbo decoding. The channel SNR is quoted in association with the 2 dB average video PSNR degradation, since the viewer will begin to perceive video degradations due to erroneous decoding of the received video around this threshold.

Mod.	Equaliser	Code	CSNR (dB)	E_b/N_0
QPSK	PSP	R=1/2	5.3	5.3
QPSK	MCMA	Turbo (1/2)	5.2	5.2
16-QAM	MCMA	Turbo (1/2)	13.6	10.6
QPSK	MCMA	Conv (1/2)	9.1	9.1
16-QAM	MCMA	Conv (1/2)	17.2	14.2
QPSK	MCMA	Conv (3/4)	11.5	9.7
16-QAM	MCMA	Conv (3/4)	20.2	15.4
QPSK	B-G	Conv (7/8)	13.2	10.8
16-QAM	B-G	Conv (7/8)	21.6	16.2

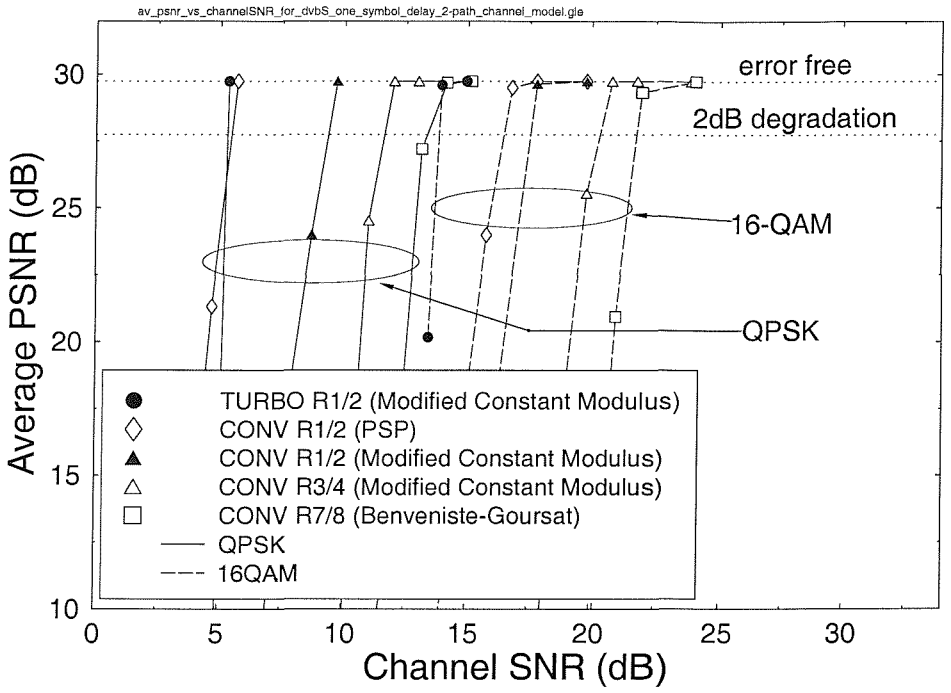
Table 3.5: Summary of performance results over the dispersive one-symbol delay two-path AWGN channel of Figure 3.3 tolerating a PSNR degradation of 2 dB.

Mod.	Equaliser	Code	CSNR (dB)	E_b/N_0
QPSK	PSP	R=1/2	4.7	4.7
QPSK	B-G	Turbo (1/2)	5.9	5.9
16-QAM	B-G	Turbo (1/2)	13.7	10.7
QPSK	B-G	Conv (1/2)	8.0	8.0
16-QAM	B-G	Conv (1/2)	17.0	14.0
QPSK	B-G	Conv (3/4)	12.1	10.3
16-QAM	B-G	Conv (3/4)	21.1	16.3
QPSK	B-G	Conv (7/8)	13.4	11.0
16-QAM	MCMA	Conv (7/8)	29.2	23.8

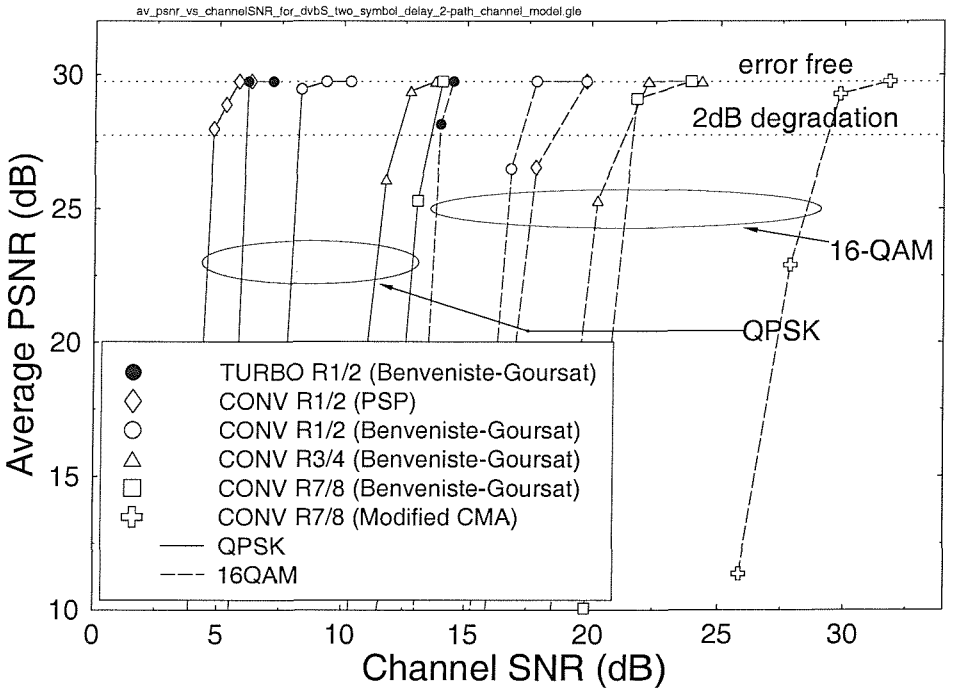
Table 3.6: Summary of performance results over the dispersive two-symbol delay two-path AWGN channel of Figure 3.3 tolerating a PSNR degradation of 2 dB.

Tables 3.7 and 3.8 provide a summary of the SNR per bit required for the various system configurations. The BER threshold of 10^{-4} was selected here, since at this average BER after Viterbi or turbo decoding, the RS decoder becomes effective, guaranteeing near error-free performance. This also translates into near unimpaired reconstructed video quality.

Finally, in Table 3.9 the QAM symbol rate or Baud rate is given for different puncturing rates and for different modulation schemes, based on the requirement of supporting a



(a) One-symbol delay two-path channel model



(b) Two-symbol delay two-path channel model

Figure 3.13: Average PSNR versus channel SNR for (a) the one-symbol delay two-path channel model and (b) the two-symbol delay two-path channel model of Figure 3.3 at a video bitrate of 2.5 Mbit/s using the "Football" sequence.

Mod.	Equaliser	Code	E_b/N_0
QPSK	PSP	R=1/2	6.1
QPSK	MCMA	Turbo (1/2)	5.2
16-QAM	MCMA	Turbo (1/2)	10.7
QPSK	MCMA	Conv (1/2)	11.6
16-QAM	MCMA	Conv (1/2)	15.3
QPSK	MCMA	Conv (3/4)	10.5
16-QAM	MCMA	Conv (3/4)	16.4
QPSK	B-G	Conv (7/8)	11.8
16-QAM	B-G	Conv (7/8)	17.2

Table 3.7: Summary of system performance results over the dispersive one-symbol delay two-path AWGN channel of Figure 3.3 tolerating an average BER of 10^{-4}, which was evaluated after Viterbi or turbo decoding but before RS decoding.

Mod.	Equaliser	Code	E_b/N_0
QPSK	PSP	R=1/2	5.6
QPSK	B-G	Turbo (1/2)	5.7
16-QAM	B-G	Turbo (1/2)	10.7
QPSK	B-G	Conv (1/2)	9.2
16-QAM	B-G	Conv (1/2)	15.0
QPSK	B-G	Conv (3/4)	12.0
16-QAM	B-G	Conv (3/4)	16.8
QPSK	B-G	Conv (7/8)	11.7
16-QAM	MCMA	Conv (7/8)	26.0

Table 3.8: Summary of system performance results over the dispersive two-symbol delay two-path AWGN channel of Figure 3.3 tolerating an average BER of 10^{-4}, which was evaluated after Viterbi or turbo decoding but before RS decoding.

Punctured Rate	4-QAM Baud Rate (MBd)	16-QAM Baud Rate (MBd)
1/2	2.73	1.37
3/4	1.82	0.909
7/8	1.56	0.779

Table 3.9: The channel bitrate for the three different punctured coding rates and for the two modulations.

video bitrate of 2.5 Mbit/sec. We observe that the Baud rate is between 0.779 and 2.73 MBd, depending on the coding rate and the number of bits per modulation symbol.

3.6 Conclusions and Future Work

In this contribution, we have investigated the performance of a turbo-coded DVB system in a satellite broadcast environment. A range of system performance results was presented based on the standard DVB-S scheme, as well as on a turbo-coded scheme in conjunction with blind-equalised QPSK/16-QAM. **The convolutional code specified in the standard system was substituted with turbo coding, which resulted in a substantial coding gain of approximately 4–5 dB. We have also shown that 16-QAM can be utilised instead of QPSK, if an extra 5 dB SNR per bit gain is added to the link budget. This extra transmitted power requirement can be eliminated upon invoking the more complex turbo codec, which requires lower transmitted power for attaining the same performance as the standard convolutional codecs.**

Our future work will be focused on extending the DVB-Satellite system, in order to support mobile users for the reception of satellite broadcast signals. The use of blind turbo equalisers will also be investigated in comparison to conventional blind equalisers. Further work will also be dedicated to trellis coded modulation (TCM) and turbo trellis coded modulation (TTCM) based single-carrier equalised modems. The impact of employing various types of turbo interleavers and turbo codecs on the system's performance is also of significant interest. Naturally, the above performance trends have to be verified also in the context of other channel types as well as transceivers, which was set aside for future research. Finally, a range of further wireless video communications issues are addressed in mode depth in [42].

Chapter 4

Coded Modulation Assisted OFDM DVB Systems for Mobile Receivers

4.1 Introduction

Since the invention of Trellis Coded Modulation (TCM) in the 1970s by Ungerboeck and Csajka [204,205], coded modulation schemes have been popular, since they achieve coding gain without bandwidth expansion. The first application of TCM can be found in the area of digital transmission over telephone lines, notably in the International Telecommunication Union's (ITU) 9.6/14.4 kbps trellis coded modem [100]. TCM has also been found to be suitable for mobile communications scenarios [206,207], where bandwidth efficiency is at a premium. Turbo Trellis Coded Modulation (TTCM) [208] has a structure similar to that of the family of power efficient turbo codes [178], but utilises TCM codes as its constituent codes. Both schemes employ set partitioning based signal labelling, in order to increase the minimum Euclidean distance between the encoded information bits.

A more recently discovered member in the coded modulation family is Bit-Interleaved Coded Modulation (BICM), which uses Gray-coded signal labelling [209]. It combines conventional convolutional codes with several independent bit interleavers. The number of bit interleavers equals the number of bits represented by a modulation symbol. Each bit interleaver is independent of the other bit interleavers.

Recently, Li and Ritcey proposed a BICM enhancement employing iterative decoding [210–212], where the BICM decoder feeds its soft output metrics back to the input of the demodulator a number of times, before hard decision is invoked. The signal mapping is based on set partitioning. We shall refer to this scheme as Iterative Decoded Bit-Interleaved Coded Modulation (BICM-ID).

The aim of our work is to study the decoding complexity and bandwidth efficiency trade-offs struck by these coded modulation schemes in comparison to the pan-European terrestrial DVB system [13, 100, 157]. We are particularly interested in comparing the performance of these coded modulation schemes and the convolutional-coded terrestrial

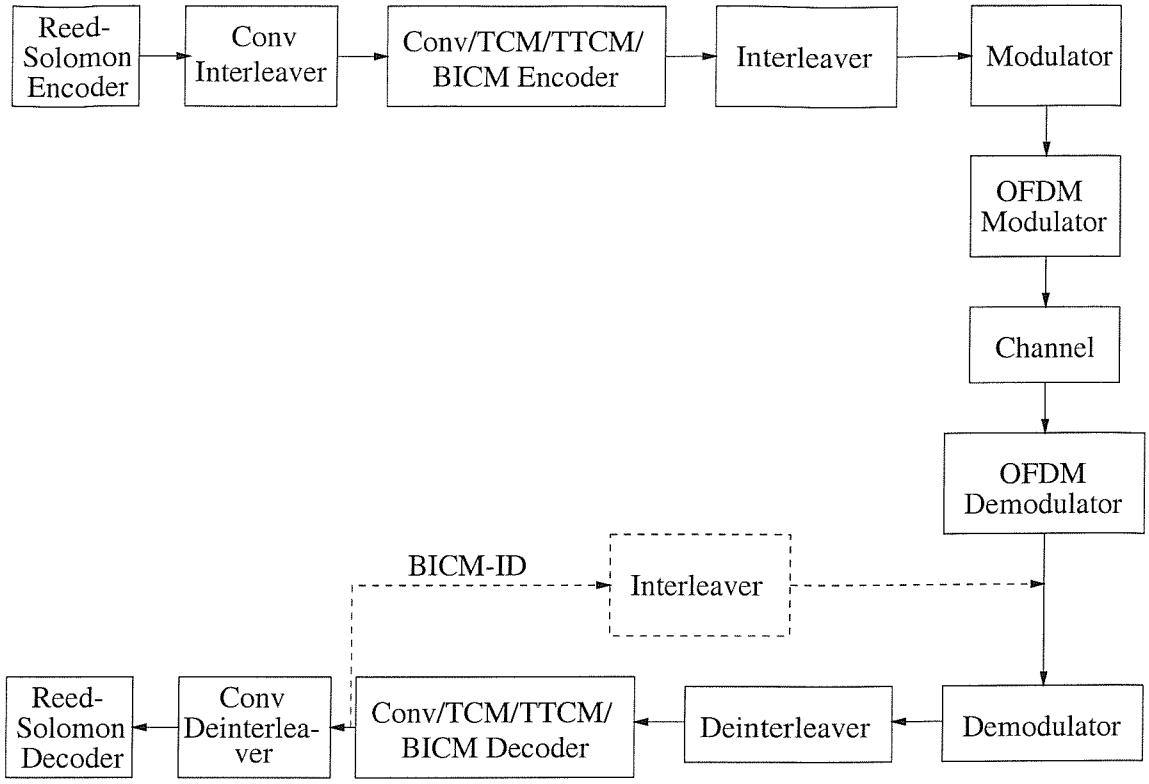


Figure 4.1: Block diagram of the different coded modulation systems.

DVB scheme [13]. In the DVB standard [13], the convolutional code acts as the inner code, which is concatenated to an outer Reed-Solomon code [23].

4.2 System Overview

The block diagram of the coded modulation schemes used in this study is shown in Figure 4.1. Pseudo-random source bits are generated and fed into the Reed-Solomon (RS) encoder. The RS encoder forms a concatenated coding scheme with one of the inner channel encoders, namely with the convolutional encoder, the TCM, BICM or TTCM encoder. The RS encoder adds a 16-byte parity part to the 188-byte information part of the codeword, in order to correct eight erroneous bytes for each 204-byte RS-coded packet. The convolutional interleaver adopted in this study is defined in the terrestrial DVB standard [13]. The coded bitstream is interleaved prior to being mapped to the modulation symbols based on their respective mapping strategies. For convolutional and BICM encoding, the mapping is based on Gray labelling [100]. On the other hand, TCM, TTCM and BICM-ID use set partitioning [205] based mapping of the bits to modulated symbols. In order to counter the effects of dispersive multipath fading channels, we have adopted the DVB standard's Orthogonal Frequency Division Multiplexing (OFDM) based transmission scheme [100]. The parameters of the OFDM DVB system are presented in Table 4.1, while those of the convolutional encoder are summarised in Table 4.2.

Table 4.3 shows the generator polynomials [205, 208] of the TCM and TTCM codes, which are presented in octal format. These codes are systematic codes and the encoder

OFDM Parameters	
Total number of subcarriers	2048 (2K mode)
Number of effective subcarriers	1705
OFDM symbol duration T_s	224 μs
Guard Interval	$T_s/4 = 56\mu$s
Total symbol duration (inc. Guard Interval)	280 μs
Consecutive subcarrier spacing $1/T_s$	4464 Hz
DVB channel spacing	7.61 MHz
QPSK and QAM symbol period	7/64 μs

Table 4.1: Parameters of the OFDM module [13] ©ETSI.

Convolutional Coder Parameters	
Code Rate	1/2
Constraint Length	7
n	2
k	1
Generator Polynomials (octal format)	171, 133

Table 4.2: Parameters of the CC(n, k, K) convolutional inner encoder of the DVB terrestrial modem [13] ©ETSI.

Code Rate	State	m	H^0	H^1	H^2	H^3
1/2 (QPSK)	8	1	11	02	–	–
2/3 (8PSK)	8	2	11	02	04	–
	16	2	23	02	10	–
	128 *	2	277	54	122	–
3/4 (16-QAM)	8	3	11	02	04	10
	16	3	21	02	04	10

Table 4.3: Summary of the TCM and TTCM constituent codes proposed by Ungerboeck [205] ©IEEE, as well as Robertson and Wörz [208] ©IEEE, where m refers to the number of coded information bits. The code generator polynomial, H^i, is presented in octal format. The '*' symbol refers to Ungerboeck's code.

attaches only one parity bit to the information bits. Hence, the code rate of the 2^{m+1}-ary signal is $R = \frac{m}{m+1}$.

Table 4.4 shows the BICM and BICM-ID codes' generator polynomials [213] in octal format. These are nonsystematic codes, which also produce one parity bit only. Hence, the code rates of these codes are similar to those of the TCM and TTCM codes, seen in Table 4.3.

The DVB terrestrial convolutional code uses soft decision Viterbi decoding, while the TCM, TTCM and BICM codes invoke the Maximum A-Posteriori (MAP) [214] decoding algorithm.

Code Rate	State	g^0	g^1	g^2	g^3
1/2 (QPSK)	8	15	17	–	–
2/3 (8PSK)	8	4	2	6	–
		1	4	7	–
	16	7	1	4	–
		2	5	7	–
3/4 (16-QAM)	8	4	4	4	4
		0	6	2	4
		0	2	5	5

Table 4.4: Summary of the convolutional codes employed in the BICM encoder. These codes were obtained from page 331 of [213] ©Prentice Hall. The code generator polynomial, g^i, is presented in octal format.

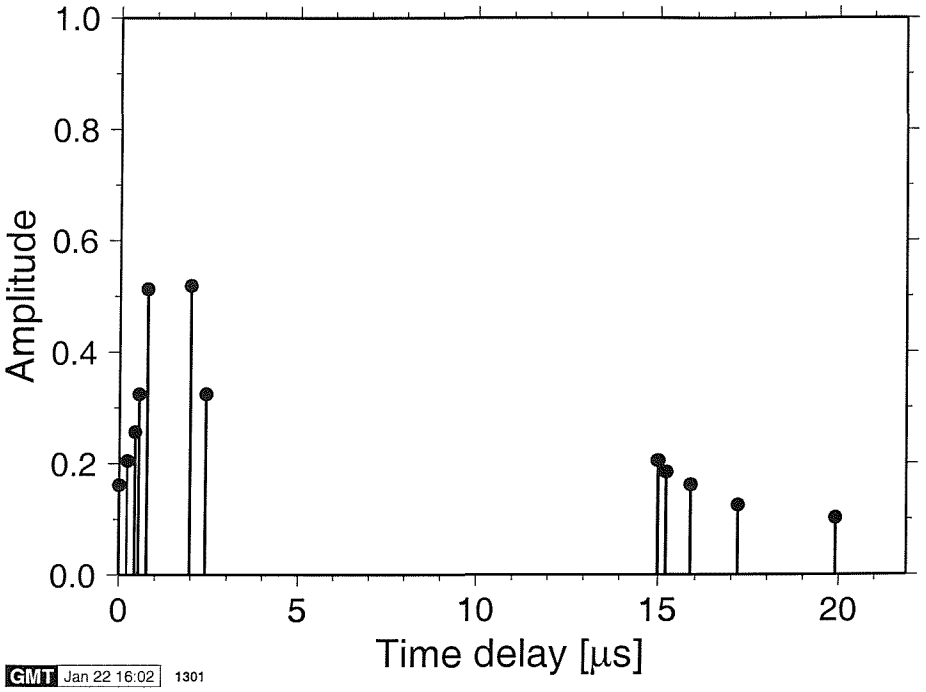


Figure 4.2: COST 207 Hilly Terrain (HT) type impulse response [182].

4.3 Channel Model

The channel model employed in this study was the twelve-path COST 207 [182] Hilly Terrain (HT) type impulse response, with a maximal relative path delay of 19.9 μs. This channel was selected, in order to provide a worst-case propagation scenario for the DVB-T system employed in our study, upon using a mobile receiver.

In the system characterised here, we have used a carrier frequency of 500MHz and a sampling rate of 7/64 μs. Each of the channel paths was faded independently, obeying a Rayleigh fading distribution, according to a normalised Doppler frequency of 10^{-5} [23]. This corresponds to a worst-case vehicular velocity of about 200 km/h. The unfaded impulse response is depicted in Figure 4.2. For the sake of completeness we note that the standard COST 207 channel model was originally defined in order to facilitate the

comparison of different GSM implementations [23] under standard conditions. The associated GSM bitrate was 271 kbit/s. By contrast, in our investigations the bitrate of DVB-quality transmissions can be as high as 20 Mbit/s, where there is a higher number of resolvable multipath components within the dispersion-range considered than at the GSM rate of 271 kbit/s. However, the performance of various wireless tranceivers is well understood by the research community over this standard COST 207 channel and hence its employment is beneficial in benchmarking terms. Furthermore, since the OFDM modem has 2048 subcarriers, the subcarrier signalling rate is effectively 2000-times lower than our maximum DVB-rate of 20 Mbit/s, corresponding to 10 kbit/s. At this subchannel rate, the individual subchannels can be considered nearly frequency-flat. In summary, in conjunction with the 200 km/h vehicular speed used the investigated channel conditions constitute a pessimistic scenario.

4.4 Simulation Results and Discussions

In this section, first we compare the performance of the coded modulation schemes to that of the DVB terrestrial system using the rate 1/2 inner code. Specifically, Figure 4.3 shows the performance of the rate 1/2 TCM, BICM, BICM-ID and DVB-T convolutional coding schemes over the wideband fading channel of Figure 4.2. We note here that these schemes have different decoding complexities. The rate 1/2 TCM code is shown to perform similarly to the DVB-T system's rate 1/2 convolutional code. However, the TCM scheme is less complex, since its code memory is only half that of the convolutional code. The convolutional code begins to show coding gain over the rate 1/2 TCM scheme, when the bit energy exceeds 12.7 dB in this case. On the other hand, the BICM and BICM-ID schemes not only exhibit a lower decoding complexity, than the convolutional code, but also achieve coding gains of 3.5 and 4.9 dB, respectively, at a BER of 10^{-4}.

Figure 4.4 shows the performance of the TCM schemes over the dispersive fading channel of Figure 4.2, when employing rate 2/3 codes using 8-state, 16-state and 64-state trellises. Again, we use the rate 1/2 and 3/4 convolutional codes employed in the DVB terrestrial system as benchmarkers [13, 157]. The performance of the 8-state and 16-state rate 2/3 TCM codes lies in between the rate 1/2 and rate 3/4 convolutional codes' performance. This implies that these rate 2/3 TCM codes perform similarly to the rate 2/3 convolutional code adopted by the DVB standard (not shown in the graph) over the dispersive fading channel of Figure 4.2. Hence, these TCM codes can achieve the same performance as the convolutional codes, but at a lower computational complexity. This observation was also valid for the rate 1/2 TCM scheme. When the code memory of the rate 2/3 TCM code was increased to 7, the associated performance was similar to that of the 64-state rate 1/2 convolutional code. Furthermore, we note that this TCM scheme transmitted using 8PSK modulation, which carries two information bits and one parity bit. On the other hand, the rate 1/2 convolutional codec of the DVB system transmitted using QPSK modulation, which carries one information bit and one parity bit within the same bandwidth. Therefore, the rate 2/3 TCM code is capable of transmitting at twice the useful bitrate compared to the DVB system within the same bandwidth and at a similar codec complexity.

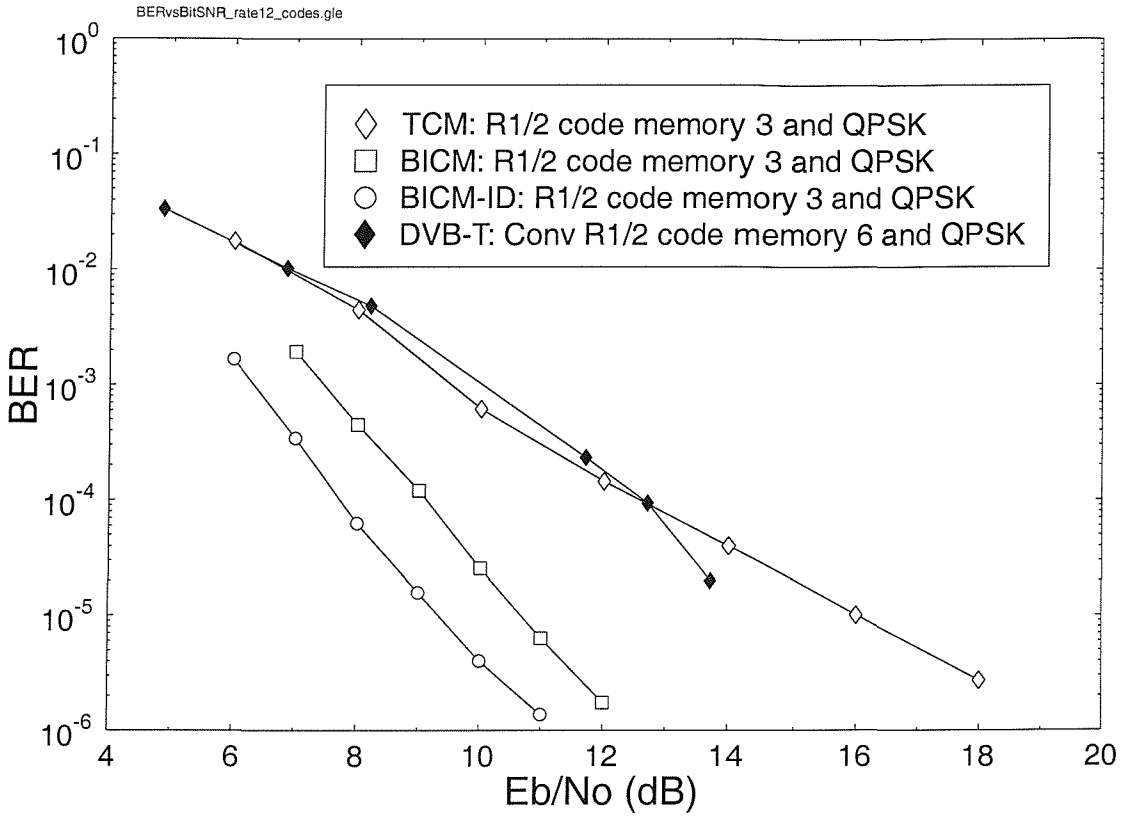


Figure 4.3: Performance of the rate 1/2 TCM, BICM, BICM-ID and DVB-T convolutional coding schemes having a different decoding complexity over the wideband fading channel of Figure 4.2. All schemes have a useful throughput of 1 bit/s/Hz.

Let us now compare the performance of the rate 2/3 BICM and BICM-ID codes to that of the DVB convolutional codes, which is shown in Figure 4.5. The 8-state and 16-state rate 2/3 BICM codes perform similarly to the 64-state rate 1/2 convolutional code. When the complexity of the rate 2/3 BICM code is increased to 64 states, it achieves a spectral efficiency of 2 bits/s/Hz, while also providing a coding gain of 1.3 dB over the 64-state rate 1/2 convolutional code, which only has a bandwidth efficiency of 1 bit/s/Hz. The 8-state and 16-state rate 2/3 BICM-ID codes exhibit an even better performance, as shown in Figure 4.5. Apart from possessing better spectral efficiency than the rate 1/2 convolutional code, the 8-state rate 2/3 BICM-ID code requires only half the decoding complexity, while exhibiting a coding gain of 2.4 dB at a BER of 10^{-4}.

We continued our evaluation of the coded modulation schemes' performance with rate 3/4 codes. The 8-state and 16-state rate 3/4 TCM schemes' performance is similar to that of the 64-state rate 3/4 convolutional code, but at a lower decoding complexity. However, when the rate 3/4 BICM code having a memory of 3 is employed, it exhibits a coding gain of 5.4 dB at a BER of 10^{-4} over the rate 3/4 convolutional code. We can attain further coding gain, if we employ the 8-state rate 3/4 BICM-ID code, which increases the coding gain by an additional 2.1 dB.

Figure 4.7 shows the performance of the corresponding coded modulation systems,

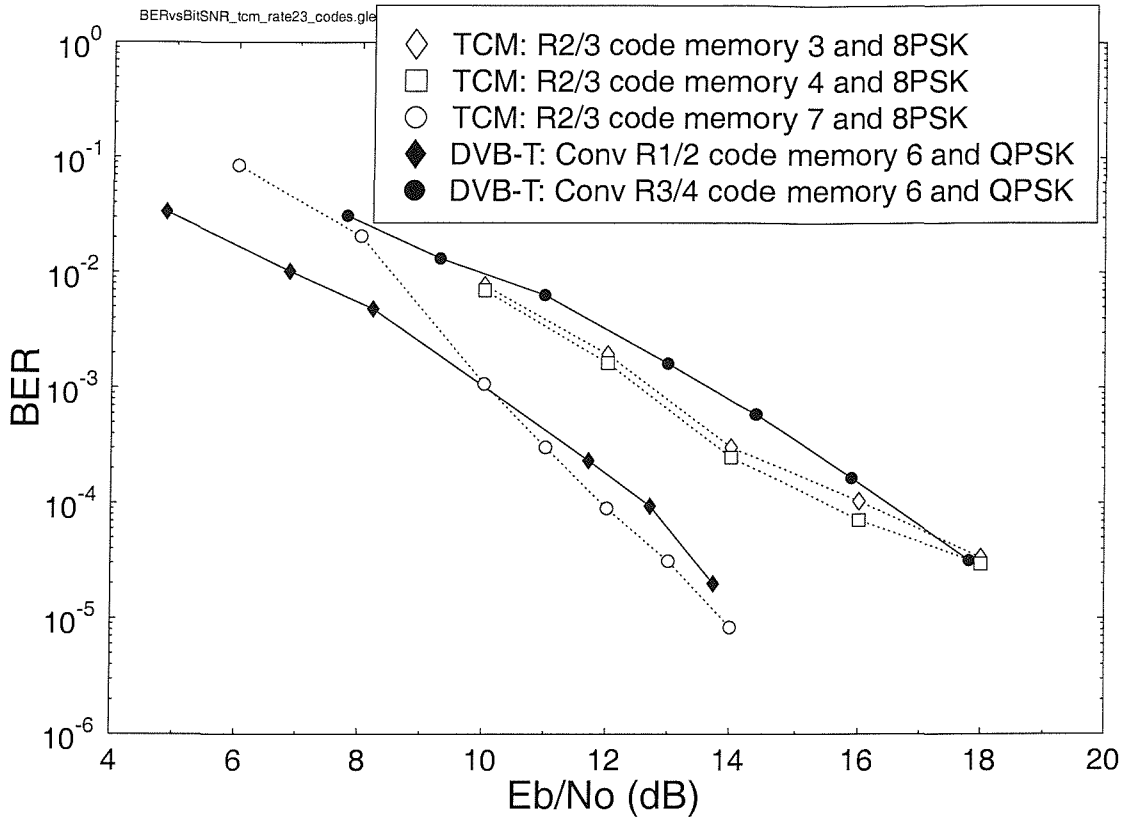


Figure 4.4: Performance of rate 2/3 TCM in comparison to the rate 1/2 and 3/4 DVB-T convolutional codes exhibiting a different decoding complexity for transmissions over the wideband fading channel of Figure 4.2. The coded modulation scheme has a throughput of 2 bits/s/Hz. The DVB scheme employing rate 1/2 convolutional code and QPSK modulation has a throughput of 1 bit/s/Hz, while the rate 3/4 inner code case has a throughput of 1.5 bit/s/Hz.

when TTCM is employed. The rate 1/2 TTCM having a code memory 3 constituent TCM scheme, is seen to perform on par with the rate 1/2 turbo code, which we proposed in [157] as a replacement for the convolutional code. However, the TTCM code employs a smaller interleaver size of 8000 bits, instead of the 17952-bit memory of the turbo code's interleaver. This is equivalent to a potential saving of 60 % of storage space and latency. The rate 2/3 TTCM scheme's performance is observed to be between that of the DVB systems employing QPSK and 64-QAM modulation modes using rate 1/2 turbo codes. This indirectly implies that the rate 2/3 TTCM scheme exhibits similar performance characteristics to the 16-QAM DVB system employing the same rate 1/2 turbo code. From Figure 4.7, we also observe that the rate 3/4 TTCM code has a bandwidth efficiency gain of 50 % in comparison to that of the rate 1/2 turbo-coded DVB-like system utilising 64-QAM. Furthermore, it also exhibits a coding gain of 2.1 dB at a BER of 10^{-4}.

Figures 4.8 to 4.11 show an alternative view of the benefits of employing coded modulation schemes for broadcast applications. The "Salesman" sequence was used in these experiments. The resolution of the sequence was 352-by-288 pixels. The sequence was encoded by the MPEG-4 video encoder. Summary of the target and coded bitrate for

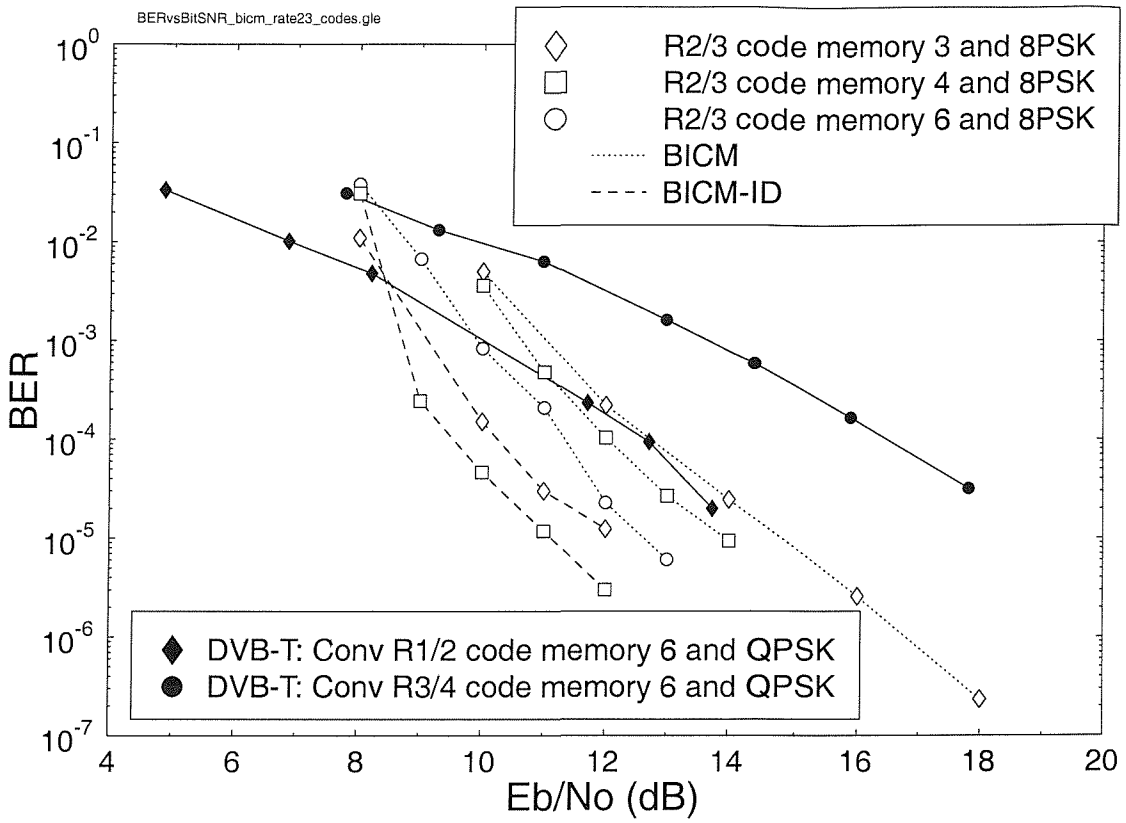


Figure 4.5: Performance of rate 2/3 BICM and BICM-ID in comparison to the rates 1/2 and 3/4 DVB-T convolutional codes exhibiting different decoding complexity for transmissions over the wideband fading channel of Figure 4.2. The coded modulation scheme has throughput of 2 bits/s/Hz. The DVB scheme employing rate 1/2 convolutional code and QPSK modulation has throughput of 1 bit/s/Hz, while the rate 3/4 inner code case has throughput of 1.5 bit/s/Hz.

the encoded "Salesman" sequence is shown in Table 4.5 along with the associated average luminance PSNR and compression ratio. The rate 1/2 coded modulation schemes were used to transmit the "Salesman" sequence coded at 561.89 kbps. When the rate 2/3 coded modulation schemes were used, the 1 Mbps coded "Salesman" sequence was transmitted, while the rate 3/4 coded modulation schemes transmitted video encoded at 1.5 Mbit/s. The increase in bitrate is proportional to the increase in spectral efficiency of the coded modulation schemes, when the code rate is increased from rate 1/2 to rate 3/4. In the event of decoding errors, the video decoder performed error concealment on the affected image region by copying the corresponding image region from the previous reconstructed image. By changing from rate 1/2 to rate 2/3 schemes, the average video quality increased by approximately 1.9 dB in this case. If the code rate is increased further to 3/4, an additional 1.5 dB PSNR average image quality improvement is recorded.

4.5 Summary

In this chapter, we have studied the performance of coded modulation schemes for transmissions over the COST 207 dispersive wideband channel of Figure 4.2. The TCM schemes

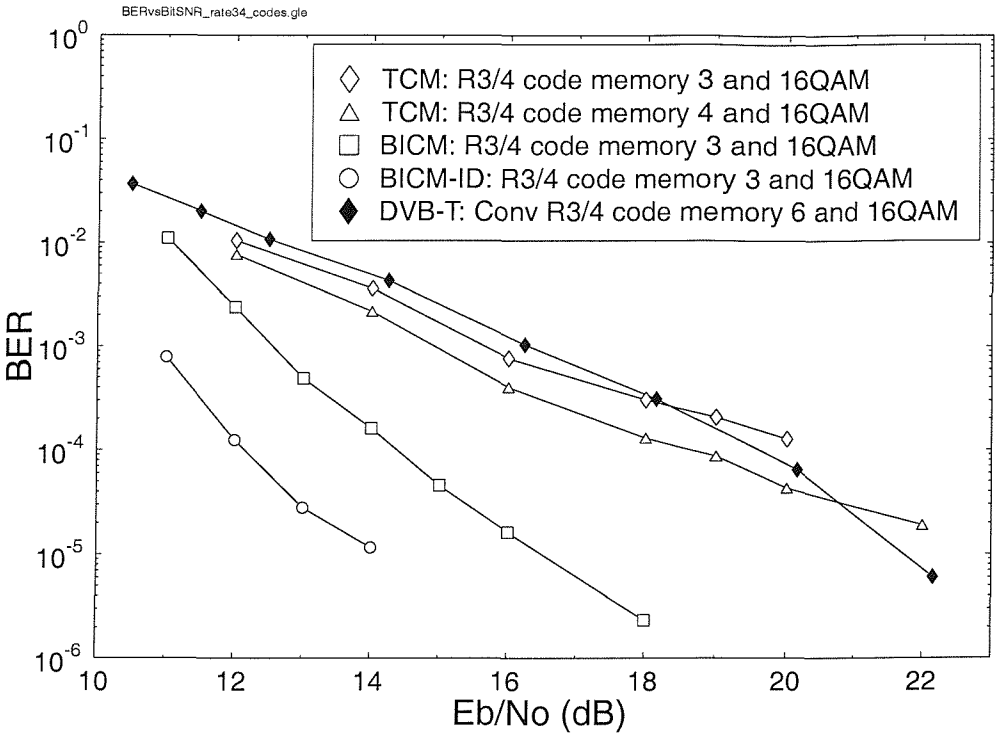


Figure 4.6: Performance of rate 3/4 TCM, BICM, BICM-ID and the DVB-T convolutional codes at different decoding complexity over the wideband fading channel of Figure 4.2. All schemes have a useful throughput of 3 bit/s/Hz.

Target Bitrate (kbps)	Coded Bitrate (kbps)	Average Luminance PSNR (dB)	Compression Ratio
500	561.89	34.32	64.95
1000	1000.35	36.41	36.48
1500	1515.35	37.74	24.08

Table 4.5: Summary of target and coded bitrate of the "Salesman" sequence at resolution 352-by-288 pixels. The associated average luminance PSNR and compression ratio are also tabulated.

attained a performance similar to that of the convolutional codes adopted by the DVB standard, albeit at a lower decoding complexity. The BICM codes consistently outperformed the corresponding convolutional code having the same code rate. The coding gain of BICM was further increased when we employed iterative decoding. Comparing the four coded modulation schemes, the TTCM systems provided the best performance in terms of power efficiency. In higher-order modulation modes, the TTCM scheme was seen to have a better spectral efficiency, than the turbo-coded DVB-like system. However, in low-order constellation modes their spectral efficiency was similar. Furthermore, we also showed that the average video quality improves when more spectrally efficient systems are invoked.

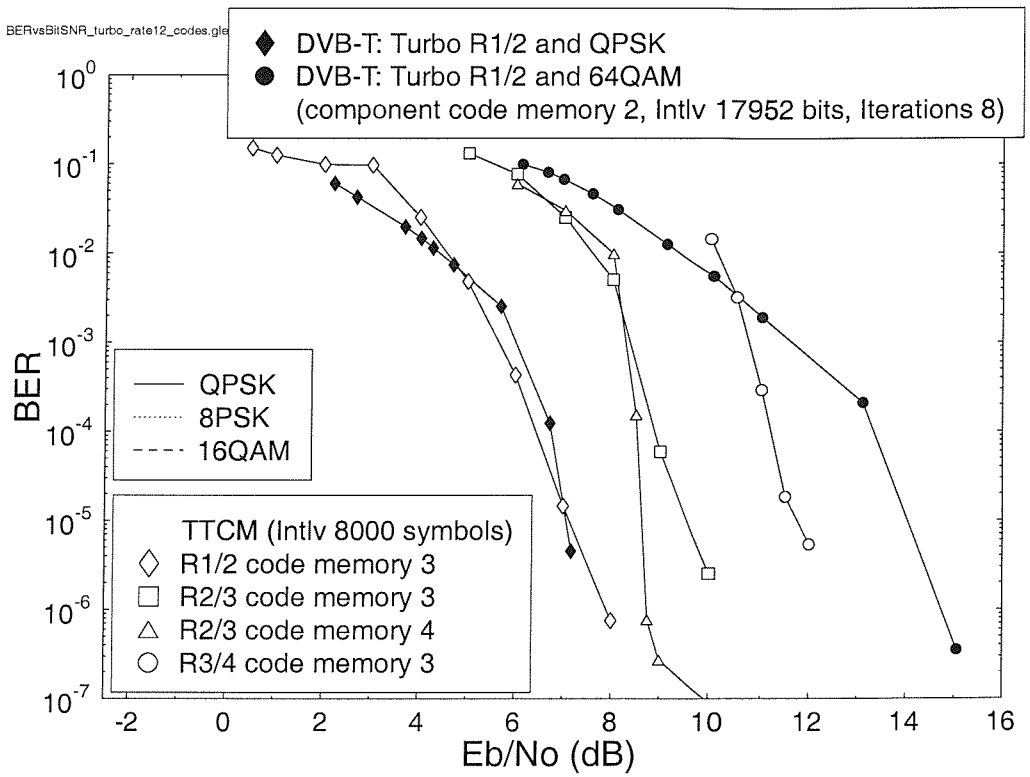


Figure 4.7: Performance of the TTCM scheme and the turbo-coded DVB terrestrial system over the wideband fading channel of Figure 4.2. The turbo-coded DVB schemes have useful throughputs of 1 bit/s/Hz and 3 bit/s/Hz for the QPSK and 64-QAM modes, respectively. The TTCM schemes have useful throughputs of 1, 2 and 3 bit/s/Hz for the rate 1/2, 2/3 and 3/4 cases, respectively.

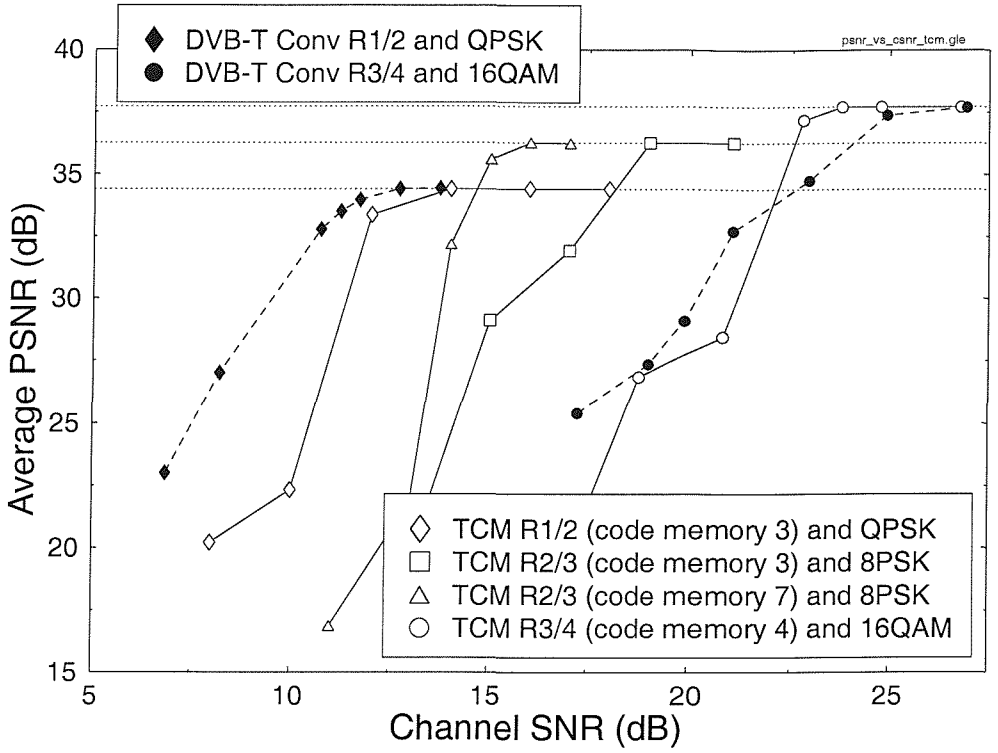


Figure 4.8: Average PSNR versus channel SNR for the MPEG-4 encoded "Salesman" sequence transmitted using the TCM scheme.

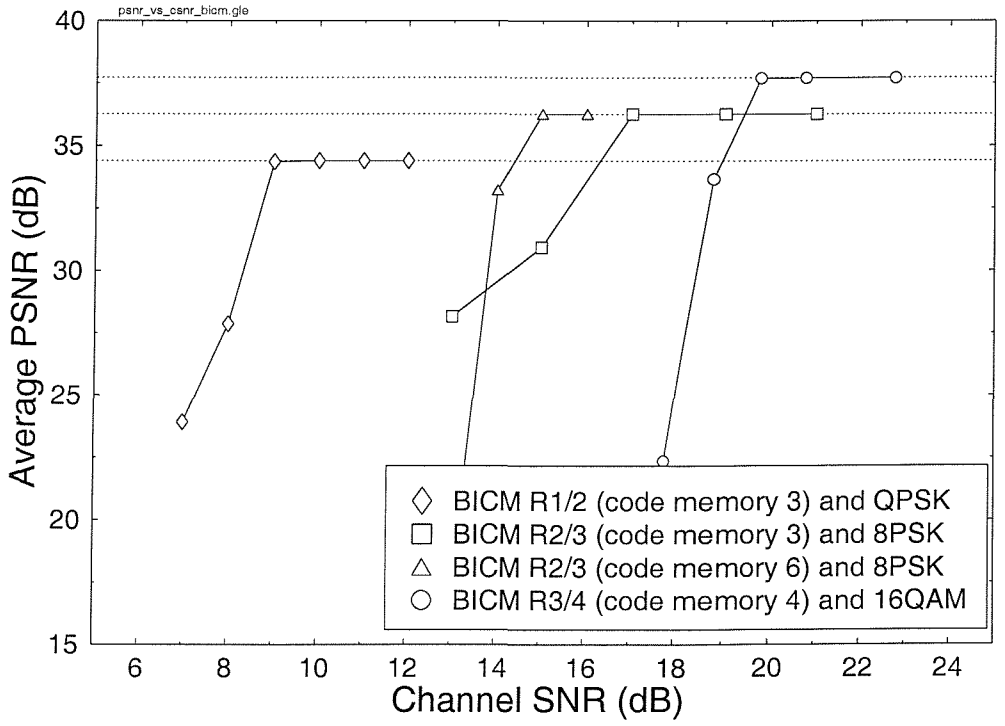


Figure 4.9: Average PSNR versus channel SNR for the MPEG-4 encoded "Salesman" sequence transmitted using the BICM scheme.

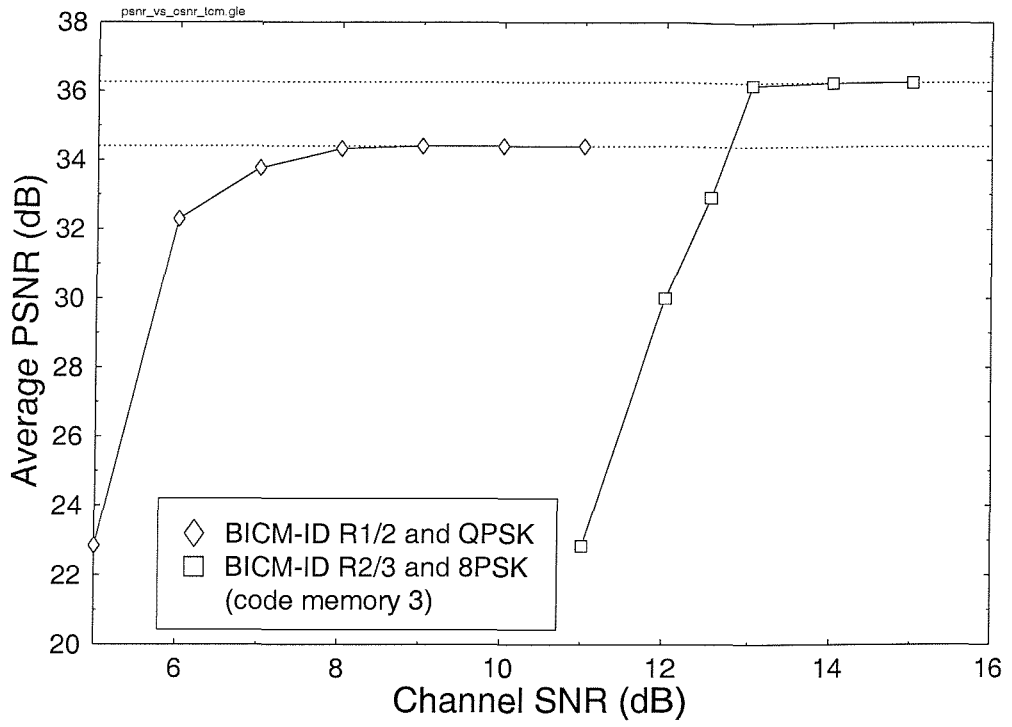


Figure 4.10: Average PSNR versus channel SNR for the MPEG-4 encoded "Salesman" sequence transmitted using the iterative decoded BICM scheme.

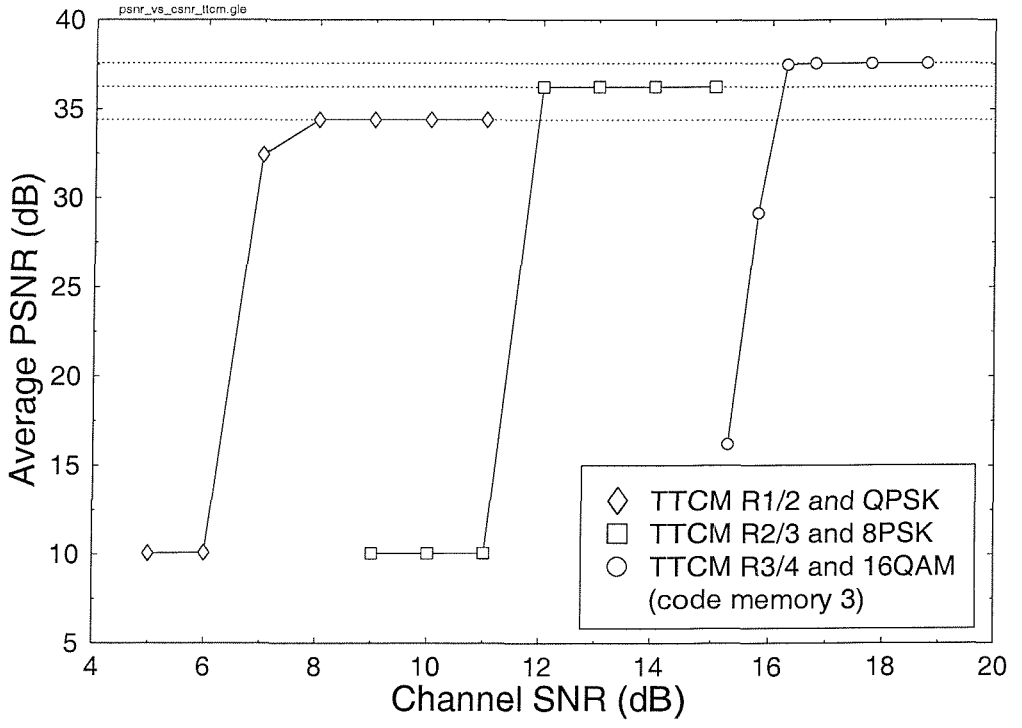


Figure 4.11: Average PSNR versus channel SNR for the MPEG-4 encoded "Salesman" sequence transmitted using the TTCM scheme.

Chapter 5

MPEG-4 Based Digital Video Broadcasting

5.1 Overview

At the time of writing the MPEG-4 video compression standard is not widely known. Hence in this chapter we provide a novel synthesis of the information available in the literature. One of the reasons for the popularity of the World Wide Web (WWW) is its ability to allow users to interact with the information presented to them in the form of links to other documents, which may or may not be at the same server site that provided the master document. These documents can be in the form of textual information, graphic files, audio files or even video files. Hence, the users are not confined to view the presentation of the documents in the order dictated by the content developers. They are free to choose their own viewing preference. This shows a paradigm shift from passive viewing to interactivity with the content.

The term multimedia conveys different meaning to different people. This is partly due to the fact that multimedia technology covers three major disciplines and their associated industries, namely entertainment, telecommunications and computers. In [29], multimedia is defined as the creation and packaging of information and its dissemination as well as distribution through end-user cooperative and interactive applications. Hence, it is not surprising that the disciplines of telecommunications, entertainment and computer businesses are expected to converge. However, as Chiariglione [215] points out, the convergence as stated is ill-posed. Instead, in his view *the barriers* inherited by the three sectors in the form of content production and packaging, information transport and processing as well as user equipment domains *that will be removed.* For example, at the time of writing a broadcast company generally enables its viewers to follow the programmes in sequence only, without any form of interactivity, apart from changing the television channels. On the other hand, a viewer browsing a web page has a choice as to which 'programmes' he or she wants to follow next. At the time of their introduction, telecommunication facilities were set up to convey mainly audio or speech information. However, with the increasing availability of multimedia source information, communication network are expected to carry video as well as other multimedia information [27]. At the time

of writing broadcast companies are poised to commence interactive services, while for the multimedia traffic on the WWW duplex communications is already the norm. Both wireline based and wireless telecommunications systems are now expected to cope with more bandwidth hungry multimedia services. On the whole, in order for integrated duplex multimedia communication to become pervasive, the entertainment industry has to launch interactive programmes, such as for example video-on-demand [15]. More efficient coding of the info is needed in order to better utilise the communication network and the interoperability of different networks has to be ensured.

The production mechanisms of audio-visual material are also evolving. Apart from using a camera and microphone for recording, as in conventional movies, a significant portion of the material can be computer generated. Using synthetic information has been a common feature in computer games, which can be in two-dimensional (2D) or three-dimensional (3D). Therefore, the compression techniques used for 3D and 2D video may differ.

The arguments presented in this section have highlighted some of the problems that have to be solved in order to enable multimedia communication to take off. The MPEG-4 standard is aimed at facilitating multimedia communications. Again in MPEG-4, the multimedia content is viewed as a set of audio-visual objects that are presented, manipulated and transported individually. These capabilities are supported by the introduction of a set of tools, which are defined in several parts of the standard. The visual [5] and audio [216] parts of the standard enable multimedia compression in the form of audio-visual entities, e.g. arbitrary shaped video objects. The conventional rectangular video frame is a special case of the video object. The MPEG-4 standard's system oriented part [217] is responsible for defining the overall architecture. We shall consider the details of these components in our ensuing sections. However, first we elaborate on the functionalities and requirements that the MPEG-4 standard is endeavouring to address in the next section.

5.2 Functionalities and Requirements

The three main requirements that form the central theme of MPEG-4 are

- content-based interactivity
- universal access
- compression.

The last requirement is addressed by the media coding component of the standard. As mentioned above, a significant coding paradigm shift facilitated by MPEG-4 with respect to other audio-visual coding standards, such as H.261, H.263, MPEG-1 and MPEG-2, is the introduction of "audio-visual objects". These objects can be related to each other both spatially and temporally, instead of the conventional approach of coding rectangular video frames along with their associated audio component.

When we refer to a video object, we are considering objects, such as humans, cars, books or even the background of a scene. These objects can be manipulated and/or motion translated relative to one another in time and space for forming a complete scene. In order to communicate the constructed scene to the end-user, the MPEG-4 system [217] has defined a tool known as Binary Format for Scenes (BIFS), which is a scene description

language discussed in Section 5.3.1. We can define audio objects, such as a man's voice, the noise generated by a car engine and music. Hence, the combination of a car and its related engine noise will produce an audio-visual object.

In this section, we have introduced the concept of audio-visual objects. Let us now look at each of the above requirements and their corresponding functionalities in turn.

5.2.1 Content-Based Interactivity

By representing a scene as a collection of objects, we are offered the ability to interact directly with the individual objects. As such, the semantics of the scene representation are important. We can attach interpretations or extra information regarding a particular object, analogously to the hypertext documents known from WWW pages. For instance, if we are viewing an advertisement of a car, we may be interested in obtaining more information concerning the car, such as its price, engine type and the accessories available. By selecting this car object, an additional window will appear on our display panel with the required information, provided that this piece of information is available.

In another scenario, we can reuse our collection of objects to form new scenes, since each object has a certain semantic message attached to it. This gives rise to a need for an object compositor at the decoder end. Its function is to reconstruct the scenes with the aid of the decoded objects. During the encoding process, composition information in the form of spatio-temporal relationships is sent together with the objects in order to facilitate the composition process. This composition information is conveyed to the receiver with the aid of the BIFS.

The scene description process, which is related to the composition process at the receiving MPEG-4 terminal, is independent of the decoding process of the audio-visual objects. This enables the manipulation of objects as well as bitstream editing. Specifically, by manipulating objects, the viewer is empowered with the ability to influence the presentation of the scene. Objects can be readily removed or added. Also, objects from other locations can be added. Furthermore, different objects can have different error protection during transmission, depending on their perceived importance [27].

The MPEG-4 committee also envisaged the composition or amalgamation of natural data with synthetic computer generated data [28]. This provides an innovative set of options for exploitation in applications such as travel advertisement, distance learning, cooperative work and virtual telepresence.

Temporal random access is also facilitated by the MPEG-4 standard, where the user can fast-forward or reverse an arbitrarily shaped image object in a certain video sequence, just like with the aid of a conventional video tape recorder.

In summary, we have three main functionalities under the content-based interactivity requirement, namely:

- content-based manipulation
- synthetic and natural hybrid coding
- temporal random access.

5.2.2 Universal Access

There are two main functionalities to be addressed in this context, such as:

- robustness in error-prone environments
- content-based scalability.

With the aid of the universal access capability, end-users are enabled to receive multimedia information through different types of networks, such as fixed cable networks, optical fibre networks, satellite networks and wireless networks. Wireless networks are particularly hostile. In order to ensure the high-integrity delivery of information, error concealment [88,103,134,135,149,152,153,218,219] and error resilient source coding techniques have to be invoked, which have been reviewed in Chapter 1. The object-based nature of MPEG-4 also facilitates a very low bitrate coding of audio-visual information, which is suitable for supporting mobile multimedia services.

Alternatively, multimedia information may have to be decoded and displayed on different types of hardware. A solution would be to introduce scalability into the content. In [87], content scalability was defined as the ability to identify and selectively decode and reconstruct the video content of interest. The content scalability mentioned entails adjusting the spatial resolution, temporal resolution, quality and complexity [30]. In terms of spatial scalability, the spatial resolution of an object can be increased or decreased. A similar capability is facilitated in the time domain, in terms of increasing or decreasing the frame rate. This enables important objects to have a higher spatial and/or temporal resolution. Therefore, the quality and complexity of different objects can be selectively controlled.

In this section, we discussed two functionalities in the context of the universal access requirement, namely error resilience and content scalability. We will now continue our discussion on the third context of the requirement, namely compression.

5.2.3 Compression

In Section 5.2.1 and Section 5.2.2, it was mentioned that the object-based nature of MPEG-4 facilitates content-based interactivity and content scalability. However, there is another underlying reason for employing coding of arbitrarily shaped objects. Research has shown that if the reconstruction of three-dimensional scene models is possible, then a substantial coding rate reduction can be achieved [29,34–39]. This ultimately corresponds to a higher compression efficiency. Wireless transmission particularly benefits from this improvement in video compression efficiency. Furthermore, more efficient coding requires reduced storage space.

The second compression related functionality is known as multiple concurrent data stream coding. According to [30], MPEG-4 will provide the ability to code multiple views (as well as soundtracks) of a scene. It is also mentioned that the redundancy which exists in multiple-view representation of the same scene can be exploited to provide an efficient representation of three-dimensional natural objects. This capability supports compressing stereoscopic video sequences.

Following the above brief notes on the objectives and motivation behind the design of MPEG-4, let us now consider the MPEG-4 system architecture.

5.3 MPEG-4 System Architecture

The MPEG-4 standard [217] specifies the overall architecture, which supports object-based compression, manipulation and transport. Several tools are defined within the system oriented part of the standard:

- Binary Format for Scenes (BIFS)
- object descriptor framework
- elementary stream management
- system decoder model.

Each of these components will be elaborated on during our discourse related to the MPEG-4 system architecture. Before we proceed further, we introduce a number of terms, which will be repeatedly used in augmenting the above concepts.

As we have argued in Section 5.2, audio or visual entities which can be viewed as individual elements in a particular scene, are known as audio-visual objects. These entities can be natural objects as well as synthetic objects and they may be static or time-variant in their nature. Each object's information is conveyed with the aid of its own streaming channel. These streaming channels are also known as elementary streams. More specifically, the elementary streams are imaginary channels for the delivery of MPEG-4 encoded multimedia and control information, which are mapped to the physical delivery channels with the aid of the tools provided by the MPEG-4 Delivery Multimedia Integration Framework (DMIF) [220]. The MPEG-4 DMIF will be discussed in Section 5.3.5. Since each elementary stream is associated with a particular object, the concept of object descriptors is introduced, in order to uniquely identify the different elementary streams. Since many of these elementary streams have to be synchronised in order to provide a coherent MPEG-4 based multimedia presentation, a time stamping technique is utilised for aligning them along the temporal axis. The system's decoder model, which will be discussed in Section 5.3.4, is introduced in order to provide an abstract notion of time in MPEG-4, as well as to facilitate the modelling and management of decoder buffers. The MPEG-4 system oriented specification, however, refrains from defining the details of the various potential delivery technologies. This isolates the design of MPEG-4 from the specifics of the various transmission systems. As mentioned above, the MPEG-4 elementary streams are mapped to the physical channels with the aid of a separate delivery layer known as the DMIF. The aim of the DMIF is to allow the simultaneous access, presentation and synchronisation of MPEG-4 multimedia contents, which are carried through different delivery technologies. This motivated the definition of a uniform interface between the MPEG-4 system layer and the delivery layer. Figure 5.1 shows the overall architecture of an MPEG-4 multimedia terminal, which will be detailed during our further discourse.

5.3.1 Binary Format for Scenes

The scene description information defines the manner in which the audio-visual objects are identified as well as located, both spatially and temporally. The scene description information is conveyed by the Binary Format for Scenes (BIFS) and it also describes the dynamic behaviour and possible interactive features of the audio-visual objects for the

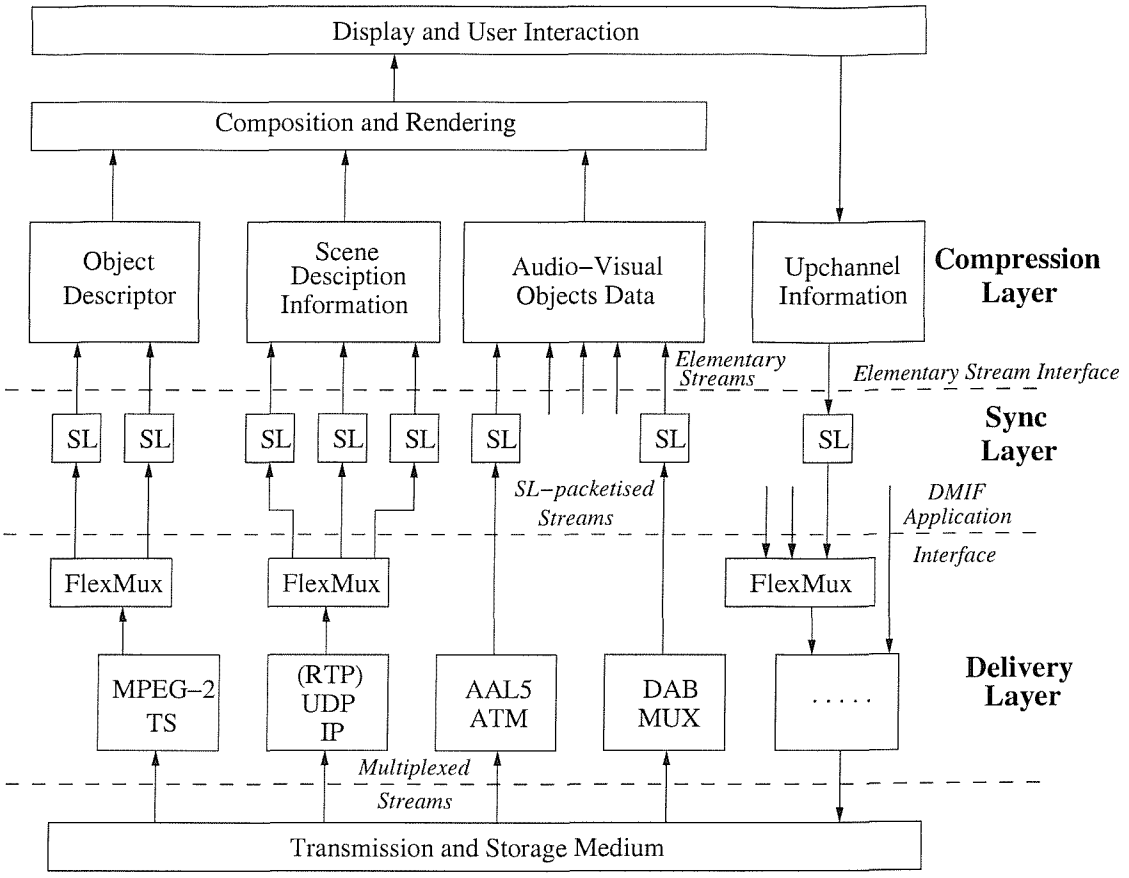


Figure 5.1: MPEG-4 system architecture (**SL**: sync layer; **TS**: transport system; **RTP**: real-time transport protocol; **UDP**: user datagram protocol; **IP**: internet protocol; **ATM**: asynchronous transfer mode protocol; **AAL5**: ATM adaptation layer 5; **DAB**: digital audio broadcast).

user. This scene description information is conveyed to the MPEG-4 codec's compositor in at least one elementary stream, as shown in Figure 5.1. Since the scene description is carried in its own elementary stream, it may include timing information. This feature allows the scene to be altered over time by content creators. Furthermore, user events, such as a mouse click on an audio-visual object in order to disable it, can be captured. These events can lead to the modification of the scene description's behaviour. In order to provide flexibility in this respect, MPEG-4 has adopted ECMAScript [221], formerly known as JavaScript [222], within the scene description process. These scripting tools enable MPEG-4 to incorporate programmability into multimedia terminals. The capturing of mouse events mentioned earlier is an example of programmability. The ECMAScript provides the means to enable the multimedia terminal to react accordingly once the mouse is focused onto the particular audio-visual object and a particular action is selected. In our example above, the corresponding action selected is to disable or remove the audio-visual object.

Since the scene description has to be transported to the user's terminal, a mechanism is required for encoding this information. As mentioned above, the tool offered for this purpose is known as the BIFS . The objects constituting the scene and the relationships

among them are coded on the basis of a tree-structured format, similar to that employed in the Virtual Reality Modelling Language (VRML) [223]. Each node in the tree contains an audio-visual primitive together with its associated attributes. In order to achieve coding efficiency, bitrate compression techniques are applied to the BIFS. Here we refrain from elaborating on the associated compression techniques employed, referring the interested reader to the literature for further details [217, 224, 225].

Apart from providing a means for encoding the scene description, BIFS also contains protocols for animating elements within the scene. A three-dimensional human head and body model has also been defined within the BIFS, such that the content creator can animate the model by controlling the nodes of the VRML tree that form the model [217, 226].

5.3.2 Object Descriptor Framework

As mentioned in our previous discussions, a particular scene description may comprise several audio-visual objects. Each of these objects has a corresponding elementary stream for its transmission. In order to allow the compositor to assign the right elementary stream to the correct object, the object descriptor framework was introduced in the MPEG-4 standard.

During our further discourse, frequent reference is made to Figure 5.2. Each object descriptor has a unique numeric object identifier (OD_ID) and it provides the required association information for a set of elementary streams. In order to differentiate amongst the different elementary streams contained within an object descriptor, a unique elementary stream identifier (ES_ID) is attached to each elementary stream. More specifically, the OD_ID links the object descriptor to a particular scene node. By contrast, the ES_IDs within the object descriptor will link the necessary elementary streams, which together constitute a single entity from the perspective of the scene description.

Object descriptors are conveyed by a specific elementary stream, which is referred to as the object descriptor stream. In order to initialise an MPEG-4 presentation, it is necessary to identify the scene description and the associated object descriptors' elementary streams. A specific object descriptor known as the initial object descriptor is defined for this purpose. This initial object descriptor is usually transmitted during the session initialisation stage, before any elementary stream channels are set up. It also contains information such as the so-called profile and level requirements for the particular MPEG-4 multimedia presentation. Specifically, the profile [227] requirements refer to the choice of tools necessary for decoding the multimedia information in the MPEG-4 presentation, while the level [227] requirements refer to parameters related to the implementation complexity assumed by the presentation, such as the maximum affordable coded multimedia bitrate and the maximum decoder buffer size.

In addition to its role as the link between the scene description and elementary streams, the object descriptor framework provides a mechanism for describing the relations between elementary streams within the same object descriptor. This allows the establishment of hierarchical relations between the elementary streams, in order to reflect the scalable nature of the multimedia encoding process. For example, two distinct elementary streams

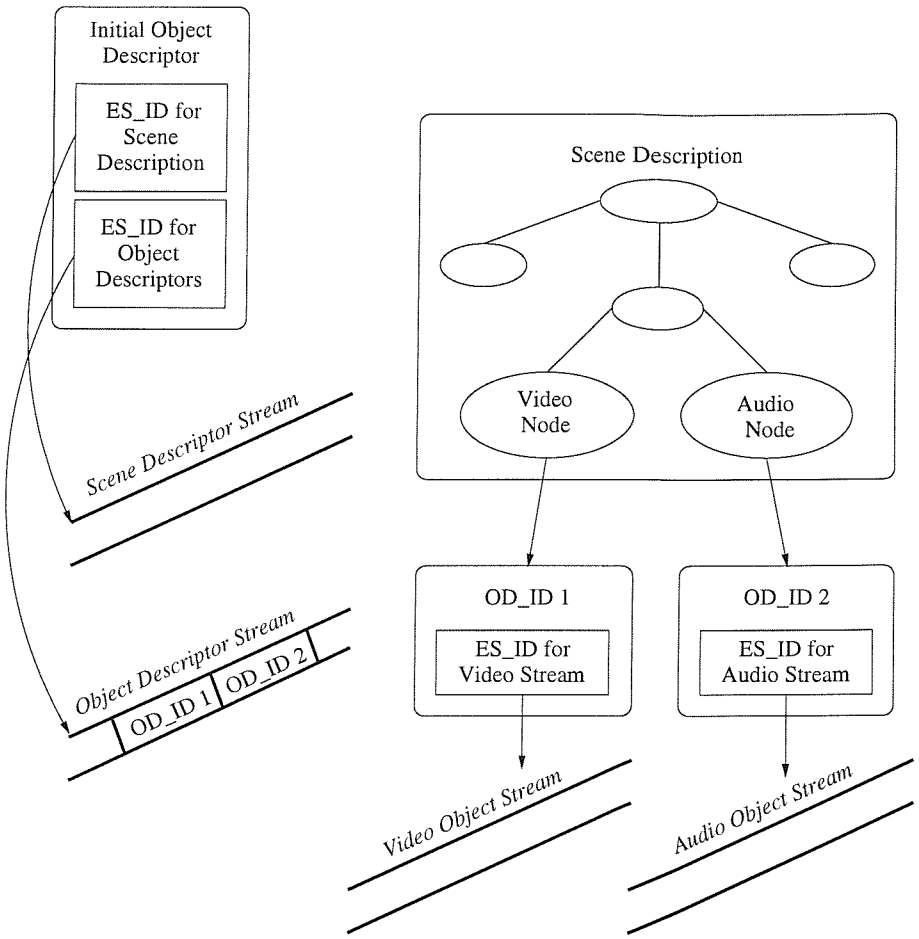


Figure 5.2: Example showing the relationship between scene graphs, object descriptors and elementary streams

corresponding to a low bitrate and a high bitrate video stream can be associated with a particular object descriptor. An MPEG-4 terminal may choose to decode both or only one of the elementary video streams, depending on its capability.

Specific descriptors for textual information based elementary streams can also be embedded into the object descriptor. This textual information can be utilised for example in keyword search processes invoked by search engines, for the textual deccription of multimedia contents or for content rating information. Furthermore, Intellectual Property Right Management and Protection (IPMP) descriptors can be included in the object descriptor. This mechanism provides conditional access to particular contents items.

Again, Figure 5.2 illustrates the relationship between scene graphs, object descriptors and elementary streams in forming an MPEG-4 multimedia presentation. Above, we have provided a rudimentary survey of the facilities supported within the object descriptor framework. The interested reader is referred to the treatise by Herpel and Eleftheriadis [99] as well as to the MPEG-4 systems specification [217] for further details.

5.3.3 Elementary Stream Management

The elementary stream management module refers to the set of functionalities required to establish synchronisation between elementary streams. This module operates within the

Sync Layer (SL) indicated in Figure 5.1. The Sync Layer is responsible for packaging the elementary streams in terms of so-called access units. An access unit refers to a complete instance of a multimedia MPEG-4 presentation entity, such as a video frame. Timing information is also added to the access unit, in order to facilitate synchronisation with other access units and the resultant packet is known as a SL packet. The sequence of SL packets stored in a particular decoding buffer is termed SL-packetised stream, as shown in Figure 5.1.

The purpose of SL packets is to ease the mapping of access units during adaptation to the delivery layer, since all elementary streams use the same SL packet structure with the relevant synchronisation information attached to the SL packets, irrespective of the delivery mechanism used. This method facilitates the synchronous decoding and composition of the corresponding access units. Synchronisation can be established with the aid of clock references and time stamps. The system's decoder model, which we will discuss in the next section, is responsible for synchronising the receiver's clock reference to the transmitter's clock reference.

As noted earlier, the MPEG-4 systems specification is defined to be independent of the delivery technology used. As such, a communication interface is defined for the mapping of the SL-packetised stream to the relevant delivery channels. The delivery technologies used may be different, but the applications are unaware of the underlying delivery mechanisms, since the MPEG-4 applications access the delivery mechanisms through a common interface known as Delivery Multimedia Integration (DMIF) application interface. This DMIF application interface will be discussed further in Section 5.3.5. However, an exception to this approach is that the MPEG-4 systems specification defined a simple multiplexing tool known as FlexMux, which exhibits a simple packet syntax. The FlexMux tool was also indicated in Figure 5.1. Its aim is to create low bitrate and low delay streams, which are then mapped to the delivery layer frame structure. This is desirable in cases where the management load, overhead or delay to set up a different transport channel for each elementary stream is too high.

5.3.4 System Decoder Model

The System Decoder Model (SDM) defines a system buffer model and a system timing model in order to describe the behaviour of a receiving MPEG-4 terminal. Figure 5.3 shows the system decoder model. The building blocks of the SDM are the DMIF application interface, the elementary stream decoder buffers, the audio-visual object decoders, the composition buffers and the compositor.

The core elements that form the processing unit in the SDM are the access units, which were introduced in Section 5.3.3. The DMIF application interface is responsible for conveying the respective access units to their decoding buffers. The decoding buffers will store these access units until they are decoded. Following their decoding, the media decoder will remove them instantaneously from the decoding buffers. Once decoded, they will be known as composition units and they are placed into the composition buffers. The size of the buffers required at the decoder can be determined from the previously defined

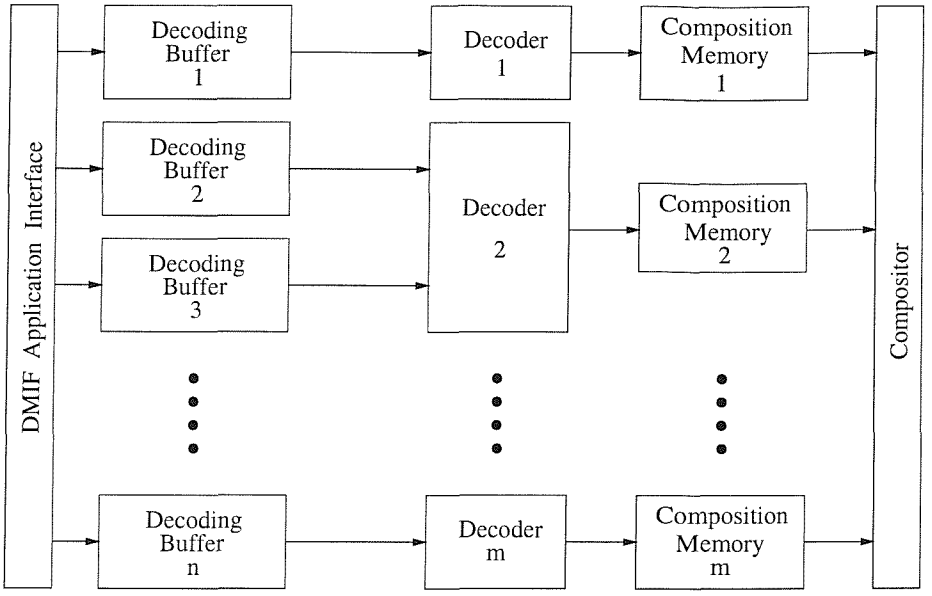


Figure 5.3: System decoder model

profile and level selected, which are indicated by the initial object descriptor during the communication setup phase, as mentioned in Section 5.3.2.

It was alluded to earlier that the time stamp attached to an audio-visual object is used to assist in synchronising its decoding in relation to other audio-visual objects. Firstly, we observe that the encoder used to encode the audio-visual object and the decoder used to recover the audio-visual object may not be operating at the same clock speed. Each of the audio-visual objects at the encoder will be "stamped" with the time of its encoding according to the encoder's system clock. Hence, the terminology Object Time Base (OTB) is used. However, the MPEG-4 decoder's clock, which is referred to as System Time Base (STB), will not be in synchronous relationship with the encoder's time base. Therefore, a time stamp known as Object Clock Reference (OCR) is sent to the decoder for aiding the synchronisation of the encoder and decoder clocks. The Object Clock Reference is the time the encoder clock shows when this time stamp is created. Hence, the OTB of the audio-visual object will have to be mapped to the decoder's STB for synchronised decoding. The associated mapping can be achieved using the following equation:

$$t_{SDT} = \frac{\Delta t_{STB}}{\Delta t_{OTB}}(t_{ODT} - t_{OTB-Start}) + t_{STB-Start}, \tag{5.1}$$

where:

t_{SDT} is the decoder's system decoding time measured using the decoder's clock

t_{STB} is the decoder's System Time Base

t_{OTB} is the encoder's Object Time Base

t_{ODT} is the object's decoding time measured using the encoder's clock

$t_{OTB-Start}$ is the value of the first Object Clock Reference time stamp of the audio-visual object

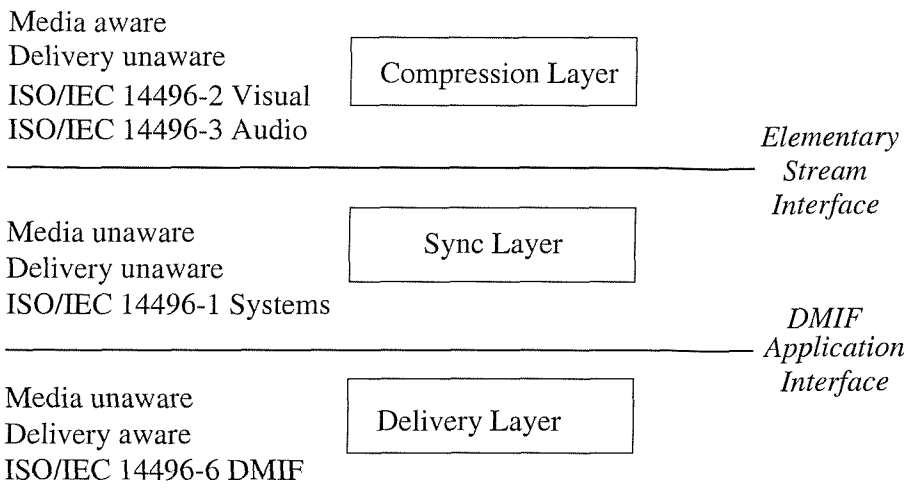


Figure 5.4: MPEG-4 specifications showing the different types of multimedia information processed at each layer.

$t_{STB-Start}$ is the value of the decoder's System Time Base, when the first Object Clock Reference time stamp of the audio-visual object is received.

In short, Equation 5.1 aims to rescale the object's decoding time based on the encoder's clock to a time based on the decoder's clock.

5.3.5 MPEG-4 Delivery Multimedia Integration Framework

At its conception, MPEG-4 was targeted for employment in multiple operating scenarios and it was designed for transmitting multimedia information using different delivery technologies. In order to create a flexible system specifications which can satisfy the above requirements, the MPEG-4 system specifications [217] were defined to be independent of the delivery mechanisms employed. On the other hand, functions which are related to the delivery mechanisms were included into the MPEG-4 delivery multimedia integration framework known as DMIF. The DMIF Application Interface (DAI) is the enabling mechanism for the MPEG-4 systems tools to access functions provided by the delivery technology. By adopting the DMIF application interface, an MPEG-4 terminal can access, present and synchronise multimedia content transmitted using different delivery mechanisms. Figure 5.4 shows the different types of multimedia information processed at every layer of the MPEG-4 specifications.

The DMIF reference architecture is shown in Figure 5.5, which illustrates three different types of operating scenarios, namely broadcast to multiple receivers, local storage and transport through some form of personal communications network. A distinct feature of the reference architecture is that the three different operating scenarios are modelled in a uniform manner. The building blocks of the architecture are the originating application, the originating DMIF, the target DMIF and the target application. The originating DMIF and the target DMIF form a session level service. The originating application is the actual MPEG-4 multimedia application in the terminal. The originating application interacts with the target application through the DMIF. For the remote communications scenario, the originating and target applications reside on separate hosts. Hence, the two

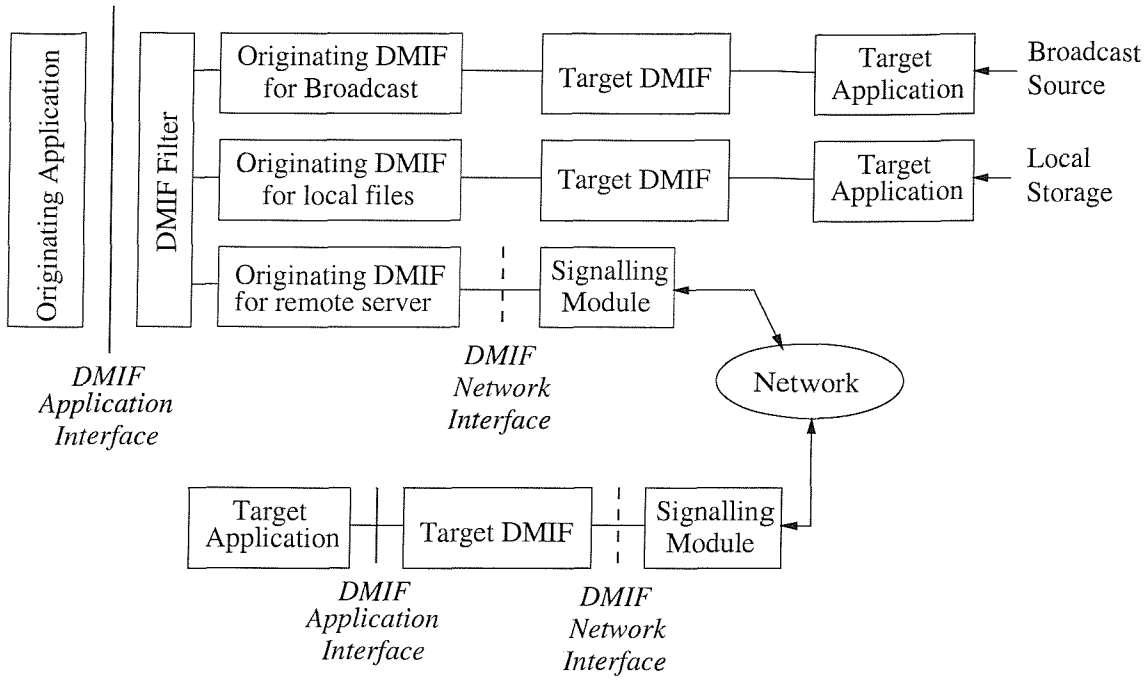


Figure 5.5: DMIF reference architecture

hosts communicate through signalling protocols specific to the communications networks employed. For the local storage and broadcast scenarios, the target application is only a conceptual model as shown in Figure 5.5 in order to maintain the uniformity of the reference architecture for the three different operating scenarios.

The DMIF application interface links the MPEG-4 application and the delivery technology. It contains primitives for the MPEG-4 application to request the setup of a new service session or to delete existing sessions, to setup or delete communication channels and to control the corresponding transfer of multimedia information through these channels. The DMIF filter plays a key role in identifying and activating the appropriate DMIF. The originating application will issue a request for a new session and for providing the resource locator for the DMIF interface. The DMIF filter will then examine this resource locator in order to activate the correct DMIF for the originating application.

The DMIF signalling protocol forms a generic session level protocol designed for supporting multimedia information streaming. The protocol stack includes the provision for activating the FlexMux tool, which was shown in Figure 5.1 and discussed briefly in Section 5.3.3, in order to optimise the employment of network resources. Once an MPEG-4 multimedia session has been initiated, several streams are selected and requested. The DMIF signalling protocol creates one channel for each requested MPEG-4 multimedia stream. The FlexMux tool can be invoked to multiplex these channels, in order to conserve network resources. However, the signalling protocol is not responsible for controlling the data streaming process. The application carries out this task.

Since the DMIF signalling protocol is a generic protocol, a mechanism known as resource descriptor was introduced by the MPEG-4 standard with the aim of identifying different transport mechanisms, such as the Internet Protocol (IP), MPEG-2 transport

streams [67, 228] and the Asynchronous Transfer Mode (ATM) transport protocol. Furthermore, due to the different signalling protocols employed by these transport protocols, a mapping functionality is incorporated into the DMIF signalling protocol, in order to associate the corresponding DMIF signalling protocol with the signalling protocols of the various transport protocols. This function is performed at the DMIF network interface by the signalling module, as shown in Figure 5.5.

5.3.6 Summary of MPEG-4 System Architecture

Let us now summarise the MPEG-4 system architecture by providing the reader with a summary of the sequence of events, which takes place when an MPEG-4 application is activated by the user. This overview will assist the reader in visualising the inter-linking of the system tools we have discussed thus far in this chapter.

An MPEG-4 application commences with the selection of the service required by the user. The sequence of events that follow are as listed below:

- The originating application initiates a request to start a new service session with the aid of the DMIF application interface.
- The DMIF filter of Figure 5.5 will examine the resource locator passed to it by the originating application through the DMIF application interface. This resource locator will be used to identify the target DMIF that provides the required service. It then contacts the corresponding target DMIF by utilising the DMIF signalling protocol as seen in Figure 5.5. The target DMIF will contact the target application in response to this event. The same session initialisation process is applied regardless of the specific operating scenarios, i.e. whether it is a broadcast, local storage or a remote interactive application. However, in remote applications a network session is established through the DMIF network interface.
- Once the service session has been successfully setup, the initial object descriptor of the requested service is transmitted. The originating application then uses this service session in order to create connections for transporting streaming data.

Above, we have briefly discussed the functionalities of the tools provided by the MPEG-4 systems and DMIF specifications. Since our treatment has been rudimentary, we refer to [5, 217, 220, 224] for further information. Details concerning the MPEG-4 systems specifications can be found in [217], as well as in the tutorial papers by Avaro *et al.* [98] and Herpel and Eleftheriadis [99]. The MPEG-4 DMIF specifications were detailed in [220].

5.4 MPEG-4 Natural Video Coding

The central concept underlying the MPEG-4 standard is that of audio-visual objects, which provide direct access to the scene contents. There are many types of multimedia information sources, such as natural still images, stereoscopic images, computer generated images, text, video and etc. In this section, we limit our discussions to natural video objects.

With our intuitive feel for video objects composing a scene, the MPEG-4 video specifications [5] define a hierarchy of classes which facilitate the description of the relationships

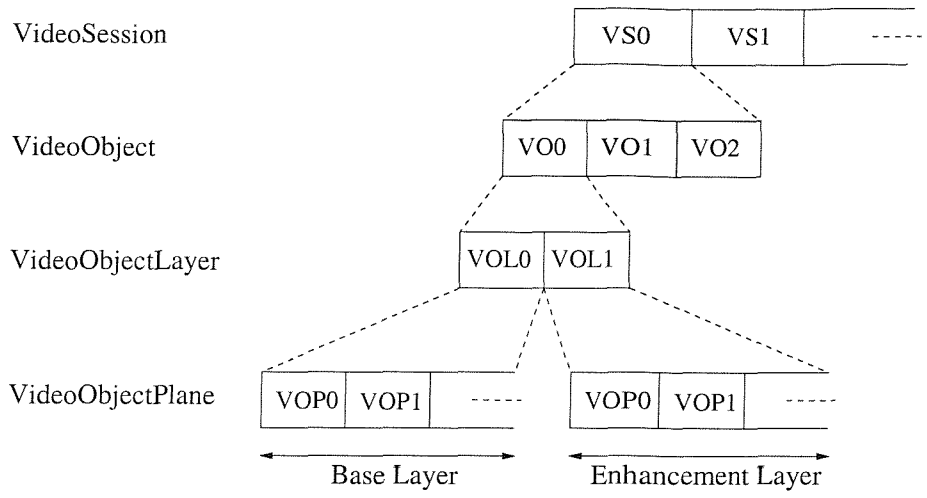


Figure 5.6: Relationship between video objects, video object layers and video object planes (adapted from Ebrahimi [32] © Elsevier Science 1997).

among video objects. This hierarchy is constituted by four elements, namely the Video Session (VS), the Video Object (VO), the Video Object Layer (VOL) and the Video Object Plane (VOP). In order to augment our discussions, we consider an example. Let us commence by viewing a particular video scene at some point in time. At the start of our observation, there are three video objects in the scene, namely a walking man, a moving car and the background scene. As time elapses, the man is seen walking towards the viewer. The car appears from behind the man, moves past the man and eventually disappears from the viewer's domain. Before we proceed, the reader is reminded that a video sequence comprises frames of still images, which are captured at certain instances in time in a sequential manner. In this context, a video session consists of the frames from the first frame until the frame captured at the instant, where the car is last seen. The next video session will commence with the walking man and the background scene only, until other video objects appear on the scene. The man, the car and the background scene are considered three distinct video objects, since they have certain semantics attached to them. Since we are dealing with video objects, the video object plane refers to consecutive snapshots of the particular object in time. The notion of the video object layer is created for introducing the concept of spatial and temporal scalability to a video object. A spatial scalability enhancement is where the spatial resolution of an object is increased. By contrast, a temporal scalability enhancement is where a video object is displayed at a higher frame rate, in order to enable the object to be viewed following a smoother motion trajectory. In both scalability cases, there will be a base layer and an enhancement layer representation of the object. The concept of video object layer is used to represent these two layers. The first video object layer will be the base layer and the second video object layer will be the enhancement layer. Figure 5.6 portrays the relationship between a video session, video objects, video object layers and video object planes.

There are several ways of utilising the video object planes. The most common one is to let each VOP contains the encoded video data for a particular time sample of a video

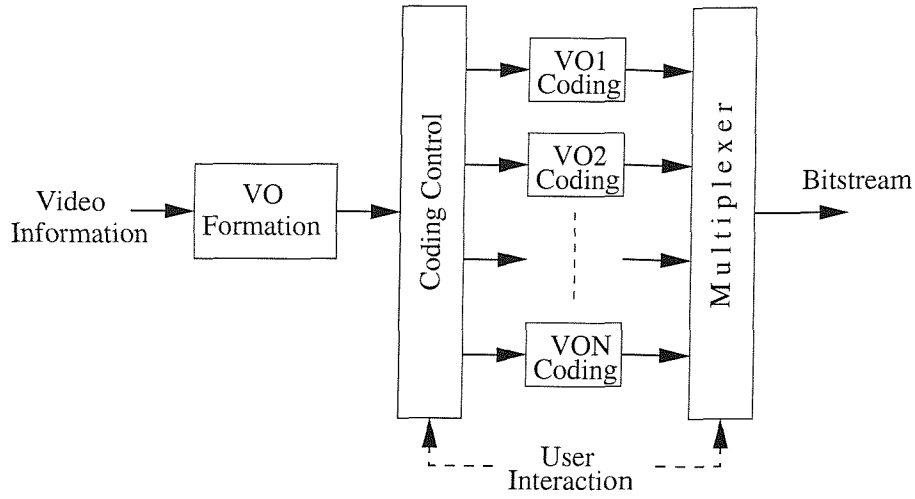


Figure 5.7: General block diagram of the MPEG-4 video encoder (adapted from Ebrahimi [32] © Elsevier Science 1997).

object. A rectangular video frame is a special case of a VOP, which is designed to hold arbitrarily shaped video object planes. Hence, the VOP will contain shape information, motion parameters and texture data. The VOP can also be used to code a video sprite, which is similar to a video frame but it appears persistently over time. The sprite's behaviour can be modified by changing its brightness and by warping it, in order to allow for spatial deformability.

The general structure of the MPEG-4 video encoder is shown in Figure 5.7. In Section 5.2, it was mentioned that content-based interactivity is one of the major requirements in MPEG-4. This is evident in the block diagram of the video encoder's structure, where a scene is composed of different Video Objects (VOs). The coding control module is needed, since different video objects may be characterised with the aid of different parameters and encoded in various coding modes. Here, a video object is not restricted to natural video objects, but it may entail synthetic video [95, 226] as well. Hence, it is expected that natural and synthetic video may use alternative coding methods. These associated coding configuration information will be attached to the respective objects for informing the receiver of the methods required for the decoding of the respective objects. Since interactivity is also a feature in MPEG-4, the end-user is allowed to influence the coding strategy with the aid of the coding control. One possibility is to distribute the total bitrate allocation budget across various objects, such that objects of specific interest are represented at a higher resolution and at a higher bitrate. Furthermore, the composition process can be adjusted in order to suit the end-user's viewing preference.

The general structure of the MPEG-4 video decoder is shown in Figure 5.8, which obeys an inverse structure with respect to that of the encoder. An important difference is the presence of a compositor, which will appropriately arrange the decoded objects based on their spatio-temporal relationships prior to displaying them. Earlier, it was noted that the end-user can interact with the encoder. In cases where there is no upstream control channel from the decoder to the encoder, the end-user may still interact with the decoder

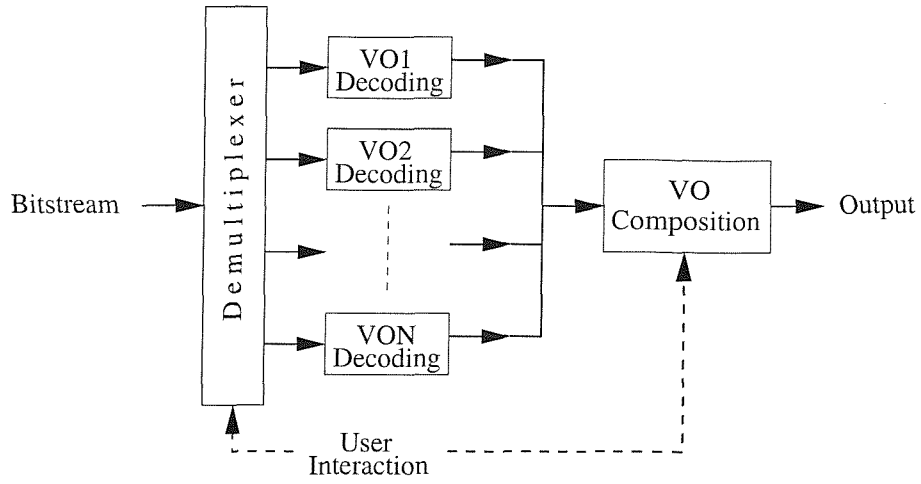


Figure 5.8: General block diagram of the MPEG-4 video decoder (adapted from Ebrahimi [32] © Elsevier Science 1997).

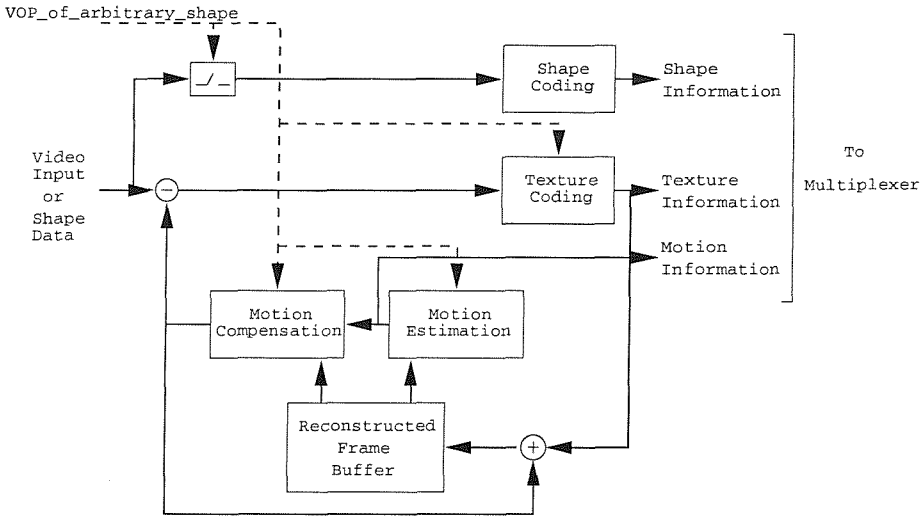


Figure 5.9: Video object plane encoder structure (adapted from Ebrahimi [32] © Elsevier Science 1997).

by either controlling the compositor or the demultiplexer of Figure 5.8, so as to change the relative spatial relationship of the objects at the compositor or to decide which object is decoded from the specific bitstream concerned at the demultiplexer.

Since each video object layer consists of a series of video frames known as video object planes, the MPEG-4 standard has defined a video object plane encoder, which is similar to those in the MPEG-1 [3], MPEG-2 [4], H.261 [1] and H.263 [2] standards. The only difference is the addition of a shape coding module in the MPEG-4 version. Figure 5.9 shows the video object plane encoder which incorporates the shape coding module. In the figure, there is a flag known as VOP_of_arbitary_shape. Its purpose is to indicate the presence or absence of shape information in the bitstream. When standard rectangular images are coded, the encoder structure shown in the figure will have the same structure as those found in the MPEG-1 [3], MPEG-2 [4], H.261 [1] and H.263 [2] standards. This structure is also known as the hybrid DPCM/transform coding technique [46]. The first

video object plane (VOP) will be encoded in intra-frame coded mode, i.e. it is coded without any reference to any other VOPs. Subsequent VOPs will be inter-frame coded, which are known as P-VOPs as they are coded with respect to previously coded VOPs. The third prediction mode is that of the bidirectionally predicted VOP (B-VOP), which forms its prediction of the current VOP on the basis of both previous and future VOPs.

After coding the shape information, the VOPs will be partitioned into macroblocks of size 16-by-16 pixels. The previously coded video frame is stored in the reconstructed frame buffer of both the encoder and that of the decoder. Motion estimation is then performed on a macroblock basis, in order to produce a motion vector which can be used to motion compensate the previously decoded frame. The prediction error is then the difference between the current macroblock of interest and the motion compensated macroblock of the previous reconstructed frame. The prediction error is then Discrete Cosine Transformed (DCT) on an 8x8 block basis. These transformed prediction error blocks will be quantised and run-length encoded using procedures similar to those in the established H.263 standard [2]. The quantization stepsize can be varied to allow the achievement of certain target bitrate and to prevent buffer overflow or underflow. The rate control process which is responsible for optimising the bit allocation of the encoded video frame is not discussed here [77, 78, 80–83, 229–231].

Having explored the general principles behind the VOP encoder structure, we will now take a closer look at each component of the encoder structure, including the shape coding module, the motion estimation and compensation module as well as the texture coding module.

5.4.1 Shape Coding

The MPEG-4 codec utilises two types of shape information, which are known as binary shape information and grey scale shape information. The binary shape information gives us the information in the form of binary labels which indicate the silhouette of the object. Hence, we are able to infer which pixels belong to the video object at any given time. The shape of a VOP is enclosed by a rectangular window having a size in multiples of 16 pixels in both the horizontal and vertical directions. The size of the bounding rectangle is defined such that there is a minimum number of 16-by-16 pixel macroblocks covering it. The pixels in the bounding box which do not belong to the object will be set to "0", while the pixels that belong to the object will be set to "255". The macroblocks within the bounding rectangle are known as Binary Alpha Blocks (BAB). As for the grey scale shape information, the difference is that each corresponding pixel can have a range of values from "0" to "255", which indicates the degree of transparency of a pixel [32, 232]. A "0" value means a completely transparent pixel, while the value "255" implies an opaque pixel. Figure 5.10 shows an example of a video object plane with its corresponding bounding box, which is partitioned into 16x16 macroblocks.

As there are two types of shape information, there are also two corresponding shape coding methods. They are binary shape coding and grey scale shape coding, which are the next topics of our discussion.

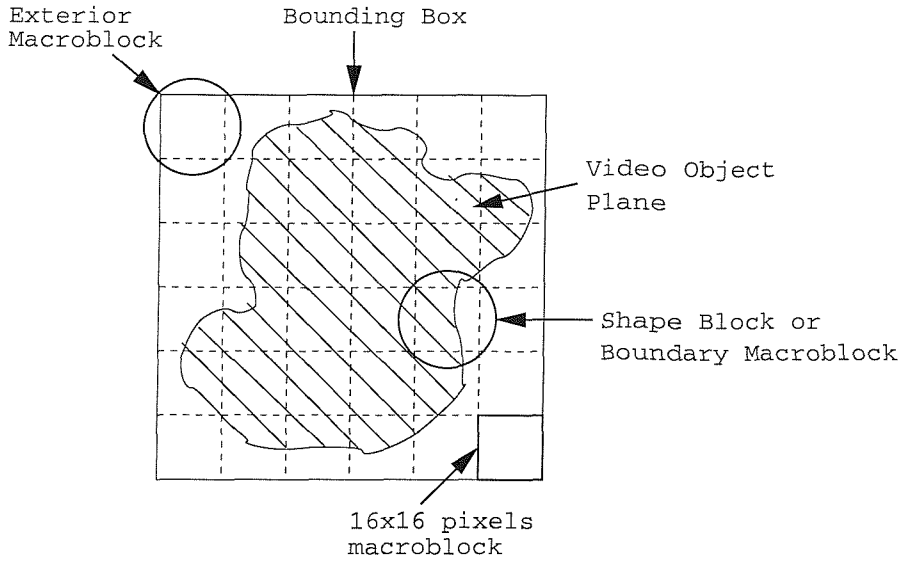


Figure 5.10: Example showing video object plane with its corresponding bounding box (adapted from Ebrahimi [32] © Elsevier Science 1997).

5.4.1.1 Binary Shape Coding

The shape compression algorithm provides two categories for coding a BAB based on the so-called Context Based Arithmetic Encoding (CAE) scheme [233], namely the intra-coded CAEs or IntraCAEs and inter-coded CAEs or InterCAEs. In the IntraCAE mode, the BAB is encoded without any reference to previously encoded BABs. By contrast, the BAB is encoded with reference to previously encoded BABs in the InterCAE mode. The motion vectors describing the motion translation of the BAB in the InterCAE case can be obtained by searching for the best matching position of the current BAB with respect to the previously encoded BAB based on a certain similarity metric, such as for example the minimum sum of absolute difference. The motion vectors found are differentially encoded.

The encoding of the BAB which belongs to an intra-coded VOP can be further classified as follows:

- If the macroblock concerned is deemed to be opaque, shape coding is not required. However, the corresponding texture information will be encoded.
- If the macroblock is classified as transparent, its shape and texture will not be encoded.
- If the macroblock is a shape macroblock, as shown in Figure 5.10, the shape information is encoded in intra mode (IntraCAE).

Similarly, the encoding of the BAB which belongs to an inter-coded VOP can be further divided into the following categories:

- If the macroblock is considered to be opaque, shape coding is not required. However, the corresponding texture information will be encoded.
- Provided that the macroblock is deemed transparent, its shape and texture will not be encoded.
- This BAB is encoded using the IntraCAE mode, if motion estimation fails to provide a suitable match for the current BAB on the basis of the previously encoded BAB.

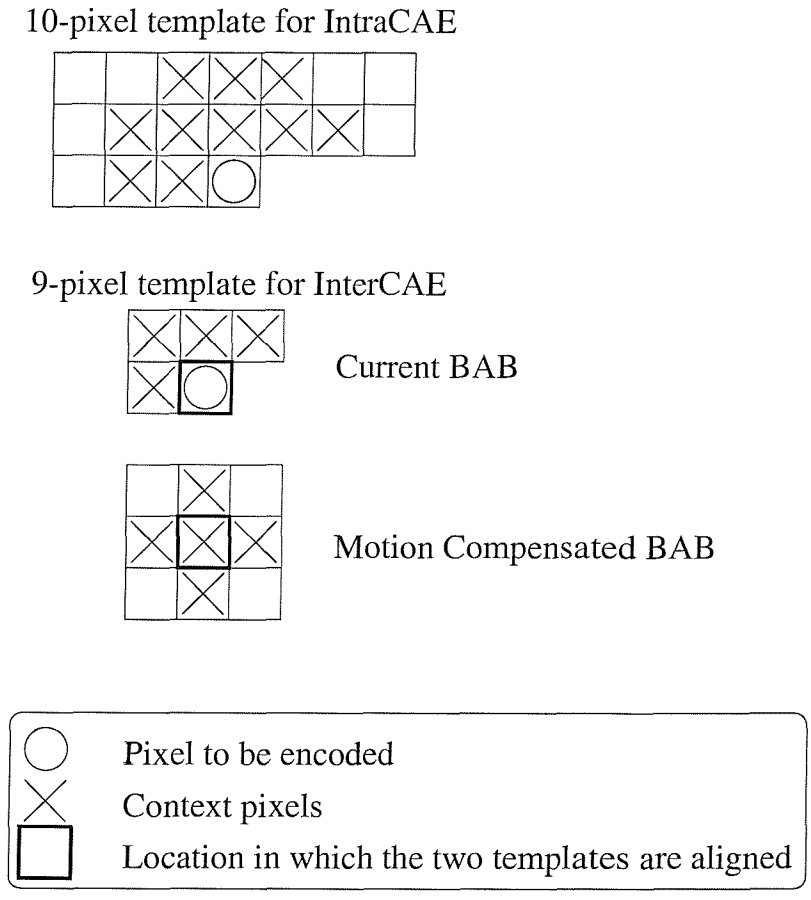


Figure 5.11: Templates for computing the context number to be used in the context based arithmetic encoder.

- The motion vector difference is zero and the BAB is not updated.
- The motion vector difference is zero, but the BAB is updated since the motion prediction error is non-zero. The InterCAE mode is employed for encoding the BAB motion prediction error.
- The motion vector difference is non-zero, but the BAB is not updated since the motion prediction error is close to zero.
- The motion vector difference is non-zero and the BAB is updated. The InterCAE is employed for encoding the BAB motion prediction error.

Reiterating, the context based arithmetic encoding algorithm [5,233] is used to encode the BAB, where each pixel is encoded in a scanline order as follows:

- A parameter known as context number is computed based on the templates shown in Figure 5.11. In an intra-coded BAB the context information is based on pixels, which belong to the BAB currently being encoded. By contrast, for inter-coded BABs, four of the context pixels originate from the BAB currently being encoded and the other five context pixels belong to the previously encoded motion compensated BAB, which also acts as the prediction BAB for the current BAB. The context

number $C(i,j)$ is computed as follows [233]:

$$C(i,j) = \sum_{k=0}^{n} c_k(i,j)2^k, \tag{5.2}$$

where (i,j) refers to the location of the pixel under consideration and $c_k(i,j)$ refers to each context pixel adjacent to the pixel at location (i,j).

- The context number serves as the index, addressing the predefined probability table [5] which outputs the probability of the pixel under consideration to be either belonging to the video object concerned or not. A constant probability distribution is employed instead of an adaptive arithmetic coding scheme in order to reduce the associated memory requirement.
- The corresponding probability value is used to assist the arithmetic encoder for codeword assignment.

Above, we have provided a brief outline of the binary shape coding method employed by the MPEG-4 encoder. Further related details can be found in the MPEG-4 video specifications [5].

5.4.1.2 Grey Scale Shape Coding

As mentioned previously, the grey scale BAB comprises a binary mask, also known as support information, and the associated transparency values for each pixel within the confines of the bounding box specified by the binary mask. Hence, two encoding stages are required by the grey scale shape coding. The first encoding stage corresponds to the binary shape coding required for generating the support information, as discussed in the previous section. The second stage encodes the 8-bit pixel transparency values within the shape concerned using technique similar to that used for compressing the arbitrarily shaped texture data, employing a block-based motion compensated DCT assisted coding scheme. Hence, this is a lossy compression scheme.

5.4.2 Motion Estimation and Compensation

Any video sequence can be viewed as a set of sequential snapshots of still images of a scene. Therefore, consecutive snapshots are correlated. It is this form of predictability or redundancy that the motion prediction mechanism is exploiting. In its basic form, we can simply use the previous frame for predicting the current frame. However, the drawback is that the motion which occurs in between the time the frames are recorded is not taken into account. Therefore, typically more sophisticated motion estimation and compensation techniques are introduced for exploiting and removing the temporal redundancy inherent in the video sequence. The prediction mechanism can be divided into two main categories. The first one is known as forward prediction, whereby the previous encoded picture is used as the reference picture. As mentioned before, the video object plane encoded in this manner is referred to as the P-VOP. The second mode is known as bidirectional prediction. Here, both the previous and future encoded pictures are used for forming the motion prediction. The video object plane encoded in this manner is termed as the B-VOP. At the commencement of the encoder's operation, the prediction

buffer is empty. Hence, the first VOP is encoded independently of other VOPs. This VOP is known as an intra-frame coded VOP or I-VOP. The major difference in MPEG-4, with respect to H.263 for example, regarding motion estimation and compensation is the introduction of video objects. Specifically, in MPEG-4 the motion estimation has to be employed for arbitrarily shaped video object planes instead of using rectangular video frames.

Similarly to the shape coding technique introduced in Section 5.4.1, the video object plane is encapsulated by a bounding box which is also divided into macroblocks of 16-by-16 pixels for luminance macroblocks and 8-by-8 pixels for chrominance blocks. When a macroblock is completely outside the video object plane, no motion estimation is employed. This macroblock is also labelled as an exterior macroblock, as shown in Figure 5.10. By contrast, when a macroblock belongs completely to the video object plane, motion estimation is carried out using standard block matching techniques on the basis of 16-by-16 macroblocks for producing a motion vector. Alternatively, each 8-by-8 pixel block can have its own motion estimation process, resulting in four motion vectors per macroblock.

The third case to consider is when a macroblock is partially filled with pixels that belong to a VOP. We shall refer to this macroblock as a boundary macroblock, as shown in Figure 5.10. In order to facilitate the employment of standard block matching algorithms here, the boundary macroblock of the reference VOP is padded by replicating boundary pixel values towards the exterior of the reference VOP. This replication process is performed in two stages, namely horizontal repetitive padding and vertical repetitive padding. By contrast, the exterior macroblock is padded by a procedure known as extended padding.

Let us illustrate the repetitive padding procedures with the assistance of a luminance boundary block, represented in matrix form as follows:

$$f(x,y) = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 47 & 46 \\ 45 & 46 & 0 & 0 & 0 & 48 & 49 & 49 \\ 44 & 45 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 44 & 45 & 45 & 0 & 0 & 0 & 0 & 0 \\ 45 & 45 & 46 & 46 & 0 & 0 & 48 & 48 \\ 47 & 45 & 46 & 47 & 47 & 0 & 48 & 48 \\ 46 & 44 & 45 & 45 & 46 & 46 & 46 & 47 \end{pmatrix}. \tag{5.3}$$

For horizontal repetitive padding, each boundary pixel value of a reference VOP is copied to the left or right of the reference VOP boundary. In our example, we assume that the pixels having zero value are to be considered outside the VOP boundary concerned. This exterior VOP region is also known as the transparent region. If two boundary pixel values exist that fill the same transparent region, the two values are averaged according to the equation shown below:

$$f'(x,y) = \lfloor \frac{f(x_l,y) + f(x_r,y)}{2} \rfloor, \tag{5.4}$$

where the symbol $\lfloor p \rfloor$ means rounding to the nearest integer lower or equals to p. The parameter x_l refers to the location of the nearest pixel value at the VOP boundary at the left of the current location x. Likewise, x_r refers to the location of the nearest pixel value at the VOP boundary at the right of the current location x. After horizontal repetitive padding, the resultant matrix $f'(x, y)$ is as shown below:

$$f'(x,y) = \begin{pmatrix} 47 & 47 & 47 & 47 & 47 & 47 & 47 & 46 \\ 45 & 46 & 47 & 47 & 47 & 48 & 49 & 49 \\ 44 & 45 & 45 & 45 & 45 & 45 & 45 & 45 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 44 & 45 & 45 & 45 & 45 & 45 & 45 & 45 \\ 45 & 45 & 46 & 46 & 47 & 47 & 48 & 48 \\ 47 & 45 & 46 & 47 & 47 & 47 & 48 & 48 \\ 46 & 44 & 45 & 45 & 46 & 46 & 46 & 47 \end{pmatrix}. \tag{5.5}$$

The process of vertical repetitive padding ensues similarly to the horizontal repetitive padding, but the process takes place in the vertical direction. The vertical repetitive padding is responsible for padding the remaining unfilled transparent pixels in the boundary block concerned. The resultant matrix after vertical repetitive padding is shown below for our example:

$$f'(x,y) = \begin{pmatrix} 47 & 47 & 47 & 47 & 47 & 47 & 47 & 46 \\ 45 & 46 & 47 & 47 & 47 & 48 & 49 & 49 \\ 44 & 45 & 45 & 45 & 45 & 45 & 45 & 45 \\ 44 & 45 & 45 & 45 & 45 & 45 & 45 & 45 \\ 44 & 45 & 45 & 45 & 45 & 45 & 45 & 45 \\ 45 & 45 & 46 & 46 & 47 & 47 & 48 & 48 \\ 47 & 45 & 46 & 47 & 47 & 47 & 48 & 48 \\ 46 & 44 & 45 & 45 & 46 & 46 & 46 & 47 \end{pmatrix}. \tag{5.6}$$

The third padding process, which is referred to as extended padding, is used to pad the remaining exterior macroblocks. For exterior macroblocks, which are adjacent to the boundary macroblocks, pixel values from the border of one of the boundary macroblocks are used for padding. The selection of the boundary macroblock concerned is based on the positions of the boundary macroblocks in relation to the exterior macroblock, as shown in Figure 5.12. The order of priority is from the largest index to the smallest index. The remaining transparent macroblocks are filled with pixel values of magnitude $2^{bits\_per\_pixel-1}$ and for 8-bit pixel resolution, this is equal to 128.

Once the reference VOP boundary macroblocks are padded, standard block matching based motion estimation can be invoked for the current macroblock. The sum of absolute difference between the pixels in the current macroblock and the pixels at the same position in the reference macroblock can be used as a similarity measure for locating the best matching region. If the current macroblock is a boundary macroblock, only the pixels that belong to the VOP will be used to compute the sum of absolute difference.

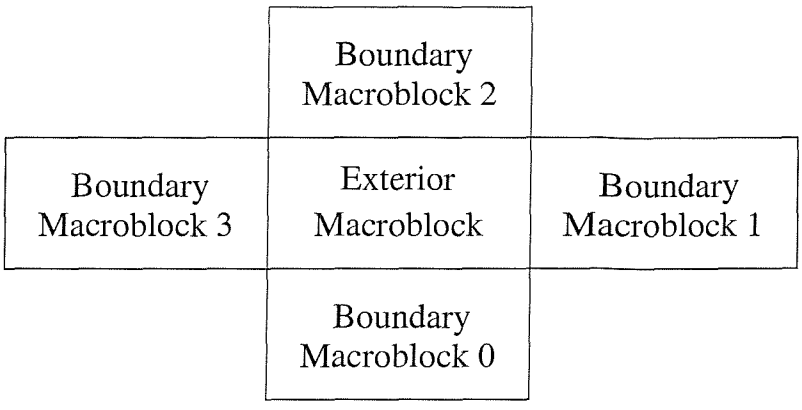


Figure 5.12: The index of the boundary macroblocks indicates their priorities in terms of being selected for padding process. The order of priority is from the largest number to the smallest number.

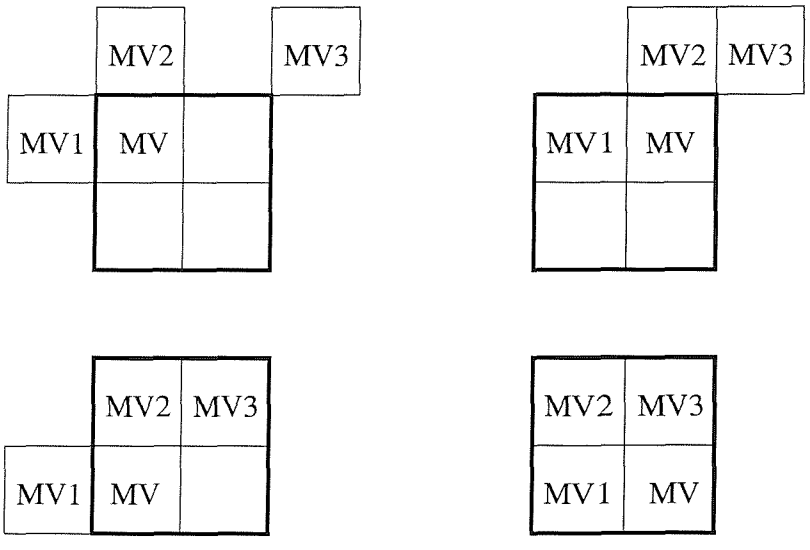


Figure 5.13: Candidate predictors for determining the motion vector MV of the current block. Specifically, MV is predicted from the median of MV1, MV2 and MV3 which are the motion vectors of the surrounding blocks.

The motion vectors obtained are encoded differentially with respect to the median values of the motion vectors of the three previously encoded macroblocks, as shown in Figure 5.13. Specifically, the prediction of MV is formed from the median of the three candidate motion vectors, namely that of MV1, MV2 and MV3. The selection of the blocks used for predicting the motion vector MV depends on the portion of the block in the macroblock concerned, as seen in Figure 5.13. The differential motion vector values are then variable-length encoded. If any of the candidate motion vectors belongs to a transparent block, this MV predictor block will be labelled as invalid. If only one of the candidate motion vector predictors is invalid, it will be set to zero. If two of the candidate motion vector predictors are invalid, they are set to the value of the third candidate motion vector predictor. If all three candidate motion vector predictors are invalid, they are set to zero.

Overlapped block motion compensation [5, 60] is also supported in MPEG-4. The motion vectors are acquired by the standard block-matching method. However, the motion compensation process differs, because during the compensation process the motion vectors of neighbouring blocks are also taken into account apart from the motion vector of the current block. Each motion vector will provide an estimate of the pixel value, and the pixel estimates are then weighted, in order to form the final prediction value.

Since the motion prediction will not be a perfect one, motion prediction errors will be generated. The process of encoding this prediction error is the topic of the next section.

5.4.3 Texture Coding

The term texture as used in this section may refer to either the texture of the actual pixel values or to that of the motion compensated residual, which is the result of imperfect motion prediction. The texture coding process consists of a number of consecutive steps, namely Discrete Cosine Transformation (DCT) [62,64], DCT coefficient quantisation and variable-length DCT codeword assignment. This process is similar to that of other block coding based schemes, such as H.261, H.263, MPEG-1 and MPEG-2. The 16-by-16 pixel macroblocks are tesselated in order to form the bounding box that hosts the arbitrarily shaped video object plane.

The first step in texture coding is to perform DCT on the actual pixel values in the case of intra-frame coded macroblocks or on the motion compensated residual in the case of inter-frame coded macroblocks, as shown in Figure 5.9. The DCT process is performed on each 8-by-8 block of a macroblock, since the DCT exhibits a high energy compaction property in the frequency domain. Hence, after the DCT process, the signal energy will be concentrated in the low-frequency range. The DCT coefficient quantisation process is used for removing the low energy, high-frequency components. For intra-frame coded macroblocks, the average signal energy of the quantised DCT coefficients can be further reduced by employing DCT coefficient prediction from neighbouring blocks, as shown in Figure 5.14. The adaptive selection of DC and AC DCT coefficient prediction direction is determined with the aid of comparing the horizontal and vertical DC gradients of the current block and its neighbouring blocks. If the DC gradient between block A and block B is higher than that between block B and block C, the AC and DC coefficients of block A are chosen for predicting the coefficients of the current block and vice versa. It should be noted that only the AC DCT coefficients at the boundary of the block are used for prediction, as indicated by the shaded region in Figure 5.14.

The resultant two-dimensional 8-by-8 pixel DCT coefficient matrix is scanned into a one-dimensional DCT coefficient string such that it typically contains a long chain of zeros followed by a non-zero value. This enables compression through the use of variable-length encoding. For inter-frame coded blocks, the well established zig-zag scan, shown in Figure 5.15, is employed. For intra-frame coded blocks, however, the encoder may select zig-zag scanning, alternate horizontal scanning or alternate vertical scanning. The selection is based on the DC coefficient prediction direction and on the activation of the AC prediction mode. If the AC prediction mode is inactive, only zig-zag scanning is used. Otherwise, either the alternate horizontal scanning or alternate vertical scanning approach of Figure

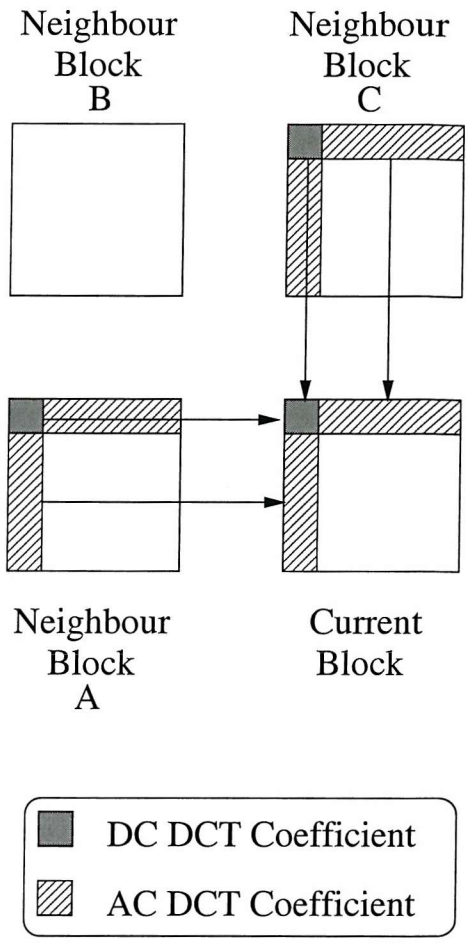


Figure 5.14: DCT coefficients prediction of current block from neighbouring blocks.

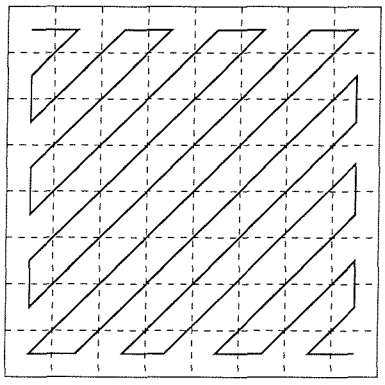
5.15 is chosen, depending on the DC prediction direction. If the DC coefficient is predicted from the adjacent horizontal block, alternate vertical scan is invoked for the current block in an effort to maximise the length of zero strings before runlength coding.

Macroblocks that contain arbitrary-shape texture data at the VOP boundaries have to be padded into rectangular macroblocks prior to discrete cosine transformation. For inter-frame coded macroblocks, the blocks are padded with zero values. For intra-frame coded blocks, a technique known as low pass extrapolation [97] is used to pad the blocks, which is performed in two stages as follows:

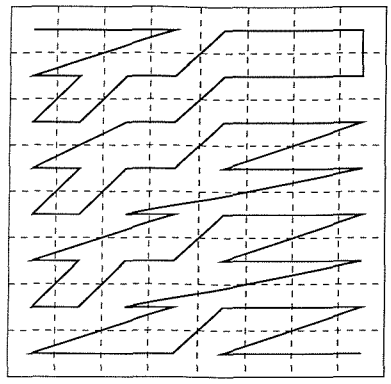
1. The mean of the pixels belonging to the boundary block is computed and used as the padding value for the transparent region.
2. An averaging operation based on the following equation is performed for all the pixels that lie outside the VOP:

$$f(x,y)_{x,y \in VOP} = \frac{f(x,y-1) + f(x-1,y) + f(x,y+1) + f(x+1,y)}{4}. \quad (5.7)$$

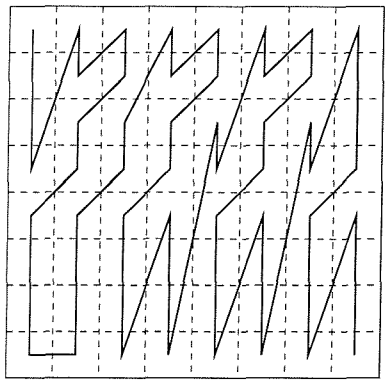
If any of the pixels considered in Equation 5.7 lies outside the block concerned, it is excluded from the filtering operation and the denominator is adjusted accordingly.



Zig-zag Scan



Alternate Horizontal Scan



Alternate Vertical Scan

Figure 5.15: The three different modes for scanning DCT coefficients prior to variable-length encoding.

5.4.4 Scalable Coding

At the beginning of this chapter it was stated that one of the MPEG-4 standard's objectives is to enable content-based interactivity in the context of audio-visual objects. Potential end-user requests are for a particular visual object to be displayed at a higher spatial and/or temporal resolution. A higher temporal resolution will result in a smoother motion perception. Both enhancements will result in a more pleasing perception for the user.

Before we proceed further, the notion of bitstream scalability will be introduced. First of all, it should be noted that the encoding of audio-visual objects will result in a set of bitstreams, corresponding to each individual object. If spatial or temporal scalability is utilised, there will be an extra bitstream, which is the so-called enhancement layer. This enhancement layer can be multiplexed with the base layer bitstream for forming the final bitstream, which will be transmitted to the receiver. Now, depending on the complexity of the decoder, it may decode only the base layer for reconstructing the object or to include the decoding of the enhancement layer as well. Although the enhancement layer and the base layer are multiplexed for transmission, it is not necessary to decode both layers for recovering the object. This is the essence of bitstream scalability. It is stated in [32] that bitstream scalability is the property of a bitstream that permits the decoding of

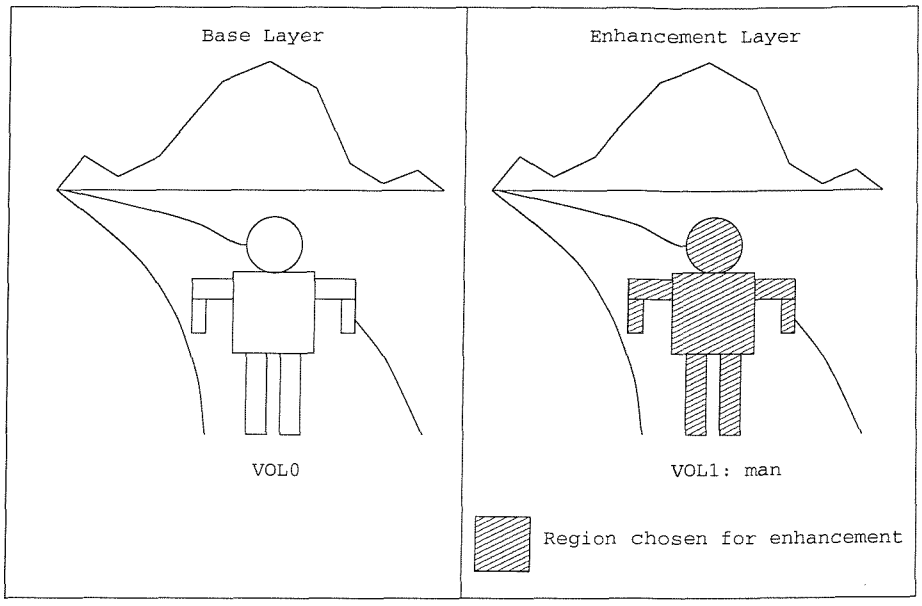


Figure 5.16: Example showing the first type of enhancement, where only a subset of the base layer is enhanced either spatially or temporally.

appropriate subsets of a bitstream for generating complete pictures of a certain resolution or quality commensurate with the proportion of the bitstream decoded. This fact has important hardware implications. A low complexity decoder can still decode the base layer for reconstructing a reasonable picture quality, while a higher complexity decoder may decode the additional enhancement layer as well, in order to attain a better picture quality.

The MPEG-4 video specifications [5] define two types of enhancements. The first one entails enhancements of portions of the base layer. The second type allows the enhancement of the entire base layer. Figure 5.16 shows an example of employing enhancements to limited portions of the base layer. In this example, there are two distinct objects in the scene, i.e. the man and the background. Referring to the data structure defined in Section 5.4, the overall scene of the base layer is treated as one video object. At the base layer, this video object has only one video object layer (VOL0). When enhancement is applied to the image of the man only, there will be an extra video object layer (VOL1) exclusively created for the man. When an enhancement of the second type is invoked, the VOL1 will represent the whole scene, similarly to VOL0.

The general encoding and decoding structure of a two-layer scalability processing model is shown in Figure 5.17. At the source, there are two encoders, one for the base layer and one for the enhancement layer. The bitstreams generated by both encoders are multiplexed and transmitted to the receiver. The receiver has a similar structure to that of the encoders. The bitstreams received are demultiplexed and sent to the decoders for processing. The functions of the scalability preprocessor, midprocessor and scalability postprocessor will be made more explicit during our further discussions on spatial and temporal scalability. These are our next topics of discussion.

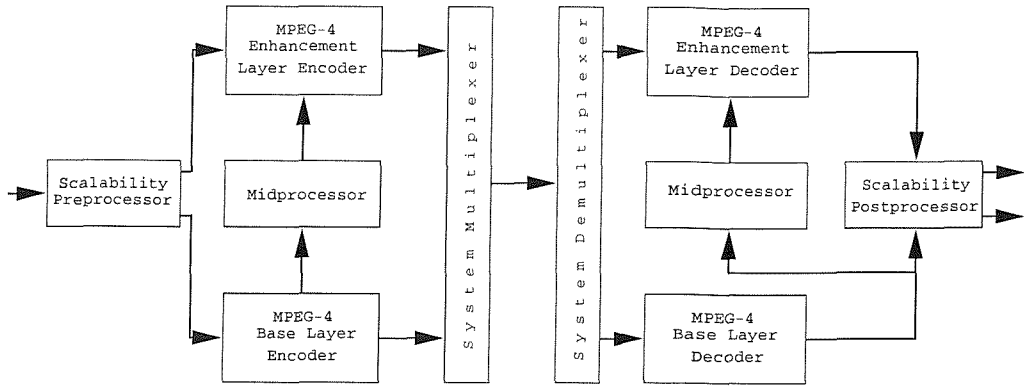


Figure 5.17: General encoding and decoding structure for a two-layer scalability processing model utilised in the MPEG-4 Video specification [5].

5.4.4.1 Spatial Scalability

In terms of spatial scalability, the viewer may have the option of increasing the resolution of the video object seen. However, this mode is only applicable to rectangular frame based pictures [5]. The discussion that ensues is made with reference to Figure 5.17. The video object planes, VOPs, form the input of the scalability preprocessor. This preprocessor downsamples the input VOPs for forming a reduced resolution base layer, which is sent to the base layer encoder for encoding. After encoding, the reconstructed base layer is fed to the midprocessor for upsampling, before being passed to the enhancement layer encoder. The other input of the enhancement layer is constituted by the higher resolution VOPs received from the preprocessor. After encoding, the base layer and enhancement layer bitstreams are multiplexed and sent to the receiver. It should be noted that the higher resolution VOPs and the upsampled base layer VOPs input to the enhancement layer encoder represent the same VOPs. At the enhancement layer, the upsampled base layer VOP acts as a prediction of the incoming VOP. At the decoder, the reverse process is invoked. The only difference is the presence of the scalability postprocessor. This postprocessor upsamples the base layer VOPs and incorporates the enhancement layer's decoded information prior to displaying it.

The VOPs of the base layer obey the encoding techniques described earlier. Hence, the base layer can be intra-frame coded (I-VOP) or predictive inter-frame coded (P-VOP and B-VOP). As for the enhancement layer, P-VOPs or B-VOPs are commonly used, although I-VOPs can also be employed. A possible relationship among the corresponding I-VOP, P-VOP and B-VOP of the base layer and enhancement layer is shown in Figure 5.18. Since the decoding of the enhancement layer is dependent on the upsampling of the corresponding base layer information, the corresponding base layer VOP must be encoded before the encoding of the enhancement layer VOP can take place.

In this subsection, we have examined briefly the spatial scalability option offered by the MPEG-4 codec. We shall now proceed to consider its temporal scalability feature.

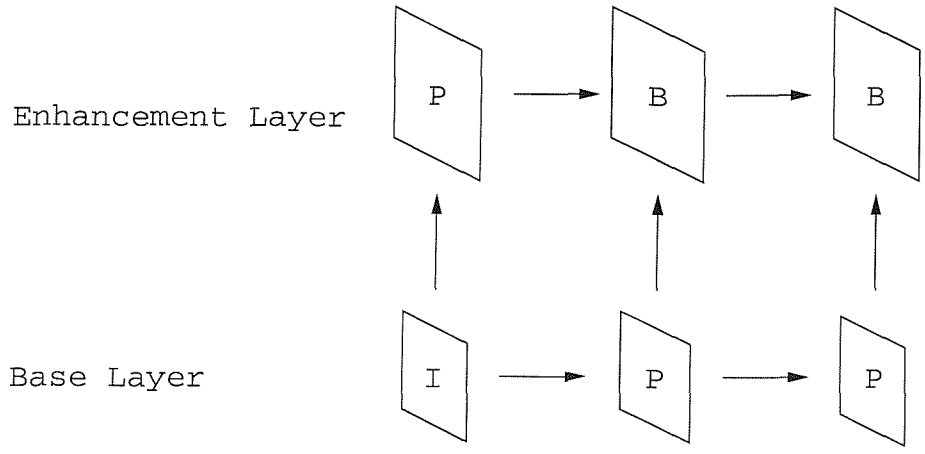


Figure 5.18: A possible relationship among the I-VOP, P-VOP and B-VOP between the base layer and enhancement layer.

5.4.4.2 Temporal Scalability

The temporal scalability mode is applicable to encoding arbitrarily shaped objects as well as rectangular shaped VOPs. With reference to Figure 5.17, the encoder's scalability preprocessor temporally demultiplexes the input video object into two groups of VOPs. One of the associated VOP groups is fed to the base layer, while the other group to the enhancement encoder. In this case, the encoder's midprocessor has no function and hence transparently conveys the reconstructed base layer VOPs to the enhancement encoder for use in its predictions. At the decoder, the inverse process is employed. Specifically, the decoder's postprocessor outputs the base layer without any conversion. As for the decoder's enhancement layer, the postprocessor temporally multiplexes the base layer and enhancement layer, in order to produce the higher temporal resolution enhancement layer VOPs.

In order to illustrate the two types of enhancements mentioned in Section 5.4.4, we consider two examples which have been adapted from [32]. The first example portrays the first type of enhancement, i.e. the enhancement of a fraction of the base layer, as portrayed in Figure 5.19 in terms of temporal scalability. In this example, VOL0 represents the background and a particular object in the scene, while VOL1 represents exclusively the particular object considered in the enhancement layer. Hence, VOL0 is the base layer in this case, which has a lower frame rate. VOL1 is coded to enable for example a particular high speed object in the scene to achieve an increased display rate.

The second example is shown in Figure 5.20, where the background represented by the rectangular frame and the arbitrarily shaped object are two separate entities. The background frame only exists in the base layer. By contrast, the arbitrarily shaped object has a base layer and an enhancement layer for a potentially higher temporal scalability. In this case, the scalability is related to the whole of the base layer constituted by the arbitrarily shaped object.

Having considered the issues of temporal scalability in this section, we provide a brief summary of the MPEG-4 video coding issues, before characterising the performance of

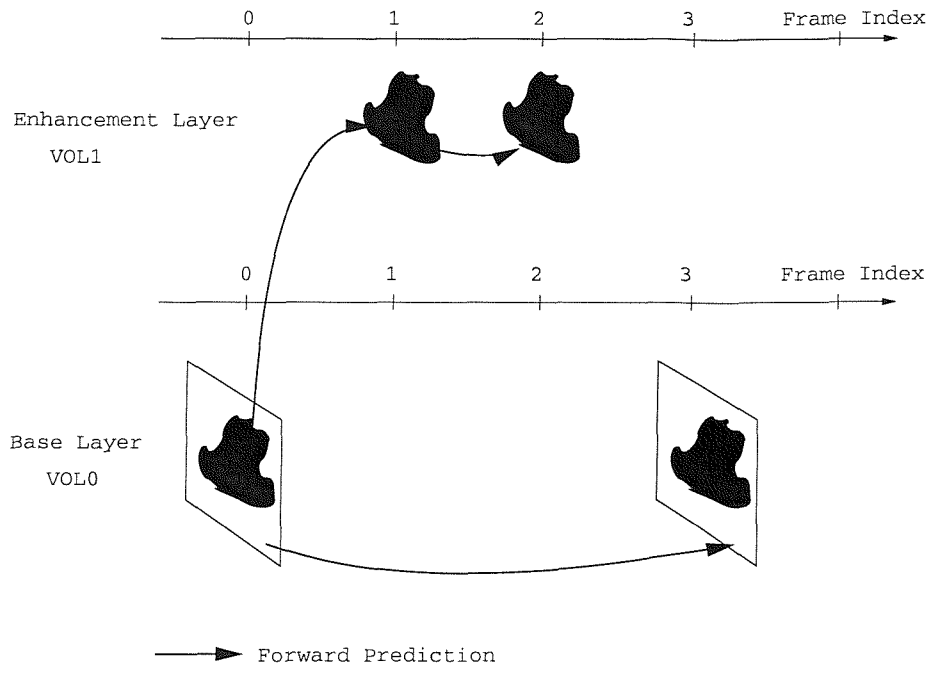


Figure 5.19: Example showing the temporal scalability of a fraction of the base layer (adapted from Ebrahimi [32] © Elsevier Science 1997).

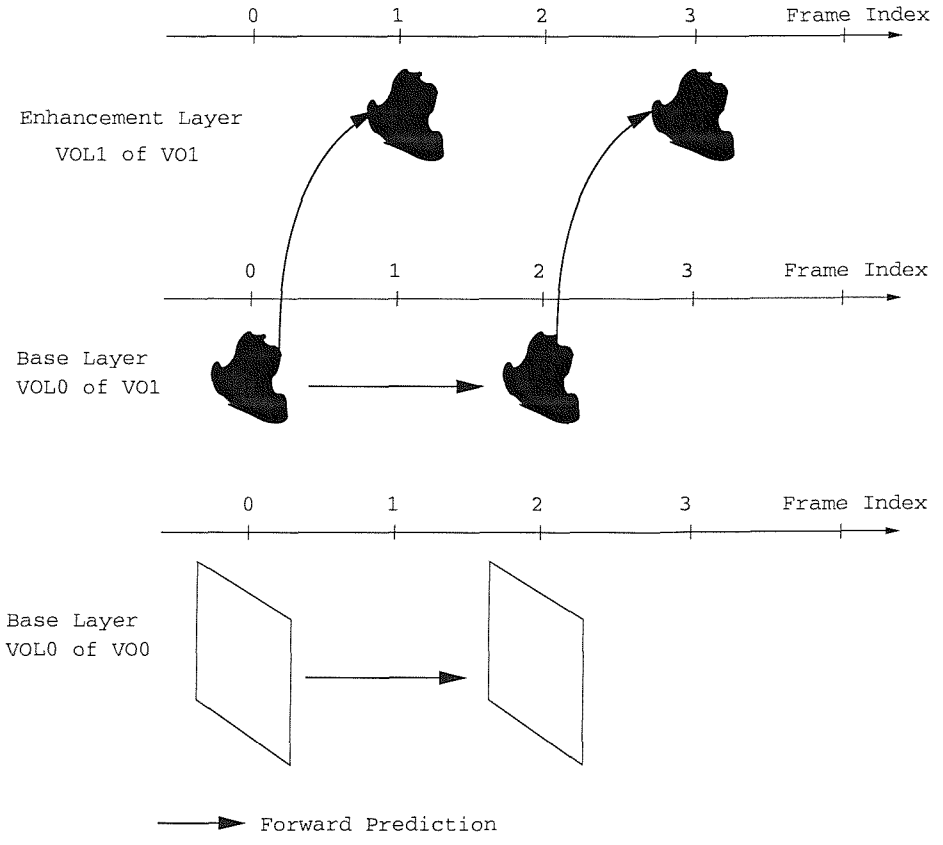


Figure 5.20: Example showing the temporal scalability of the entire base layer (adapted from Ebrahimi [32] © Elsevier Science 1997).

Figure 5.21: First frame of the QCIF "Miss America" sequence.

the codec.

5.4.5 Summary of MPEG-4 Natural Video Coding

The MPEG-4 visual standard [5] treats the coding of arbitrary shaped video objects as its main aim. This allows convenient manipulation of the video objects within the frames. The MPEG-4 video encoder is seen as an evolution from the existing video coding standards, such as H.263 and MPEG-2, which encode only rectangular frame-based video. In order to support arbitrary-shape object encoding, the MPEG-4 Visual standard has specified the encoding of shape information, which was discussed in Section 5.4.1. The texture coding tools, though similar to those used in existing video coding standards, have to be modified in order to accommodate the encoding of arbitrary shaped texture data. The necessary modifications were summarised in Section 5.4.3. The standard has also specified several enhancements in order to increase the achievable compression efficiency, such as AC and DC DCT coefficient predictions, as well as three alternative DCT coefficient scanning techniques prior to the variable-length codeword assignments. As in previous video coding standards, the MPEG-4 visual standard [5] has also offered scalability tools in the form of both temporal and spatial scalability modes. These scalability modes have also been extended in order to account for the object-based coding schemes adopted by the standard.

5.5 Performance of the MPEG-4 Codec

The investigations conducted in this section are aimed at characterising the performance of the MPEG-4 codec employing a range of video sequences. The simulations were carried out using colour video sequences, which were sampled at 10 frames/s and 30 frames/s. The video sequences used were "Miss America", "News" and "Foreman" of dimensions 176-by-144 pixels. This video representation format is known as the Quarter Common Intermediate Format (QCIF). The first frame of these three sequences is shown in Figures 5.21 to 5.23.

Initially, the MPEG-4 encoder was set to operate without activating the rate control. The quantiser step size was set to the finest and coarsest levels, in order to gauge the lower and upper bound of the coded bitrate and that of the compression ratio attainable for the video sequences considered. Explicitly, Table 5.1 presents a summary of the MPEG-4

Figure 5.22: First frame of the QCIF "News" sequence.

Figure 5.23: First frame of the QCIF "Foreman" sequence.

codec's performance, when the quantiser step size was set to the finest and coarsest levels for the "Miss America", "News" and "Foreman" QCIF video sequences. As expected, the codec exhibited the highest compression ratio for the "Miss America" sequence, followed by the "News" and Foreman" sequences, respectively. The high compression ratio of the "Miss America" sequence is a consequence of its low spatial detail and limited amount of motion activity. The MPEG-4 encoder is capable of achieving a coded bitrate of 6.85 kbit/s, when the coarsest quantiser is used for encoding the "Miss America" sequence sampled at 10 frames/s. At 30 frames/s, the lowest achievable average coded bitrate was 13.42 kbit/s, which corresponds to approximately twice the lowest achievable coded bitrate experienced at 10 frames/s case. The "Foreman" sequence which exhibits a fair amount of spatial detail, but only a limited amount of motion activity requires a minimum coded bitrate of 30.89 kbit/s and 52.85 kbit/s for scanning rates of 10 frames/s and 30 frames/s, respectively.

In our next experiment, the MPEG-4 encoder's rate control [230] was activated. Figure 5.24 and Figure 5.25 show the performance of the video codec in terms of the achievable PSNR image quality, against bitrate and compression ratio, respectively. The MPEG-4 codec's performance is observed to scale quasi-linearly over the operating range. In order to increase the video scanning rate from 10 frames/s to 30 frames/s, the video encoder is required to approximately double its bitrate in order to maintain the same image quality for the "Miss America" and "News" sequences. However, the "Foreman"

Sequence	Type	Frame Rate (fps)	Quantiser Step Size	Coded Bitrate (Kbps)	Average Luminance PSNR (dB)	Compression Ratio
Miss America	QCIF	10	1	496.59	47.41	6.12
			31	6.85	33.87	443.78
News	QCIF	10	1	456.10	47.19	6.67
			31	13.09	26.81	232.27
Foreman	QCIF	10	1	776.41	47.34	3.92
			31	30.89	27.77	98.45
Miss America	QCIF	30	1	1442.40	47.04	6.33
			31	13.42	33.73	680.08
News	QCIF	30	1	1002.93	47.06	9.10
			31	21.63	26.95	421.83
Foreman	QCIF	30	1	1754.14	47.21	5.20
			31	52.85	28.44	172.64

Table 5.1: Upper and lower bounds of the MPEG-4 codec's performance for various QCIF resolution video sequences, where the video encoder employs either the finest or the coarsest quantiser, in order to obtain the highest and lowest quality encoded video, respectively. The rate control was disabled in this experiment.

sequence required only about 50 % bitrate increase, in order to maintain the same image quality when the video scanning rate is tripled. This is attributed to the fact that upon reducing the frame scanning rate to 10 frames/s, the motion prediction between the 100 ms spaced video frames becomes less efficient and hence the achievable compression ratio is reduced. Viewing this phenomenon from a different angle, we observe that we can obtain the same PSNR image quality at a factor three lower frame scanning rate at about half the bitrate, when the associated slight jerkiness is acceptable. This fact implies a reduction of the bandwidth required for transmitting the encoded video. It is also observed from Figure 5.25 that when maintaining the same PSNR image quality in comparison to the original 30 frames/s sequence, there is an approximately factor two compression ratio loss, when the video frame scanning rate is decreased from 30 frames/s to 10 frames/s. This is a result of reduced predictibility of the 100 ms spaced frames. It is interesting to observe in Figure 5.25 that as the compression ratio is decreased, the image quality expressed in terms of the PSNR attained by the video codec for the "Miss America" sequence at 10 frame/s and 30 frame/s converges, while this is not observed for the other two sequences. This can be explained as follows. The "Miss America" sequence is more amenable to compression compared to the other two sequences. Since it contains only a small amount of motion in conjunction with limited spatial detail, the difference between two consecutive frames is limited even for the 10 frames/s scenario. Hence at low compression ratios, the PSNR image quality converges, since even the low frame scanning rate 10 frames/s "Miss America" sequence is smoothly evolving and therefore it can be encoded at a high quality.

We continued our experiments using three different Common Intermediate Format

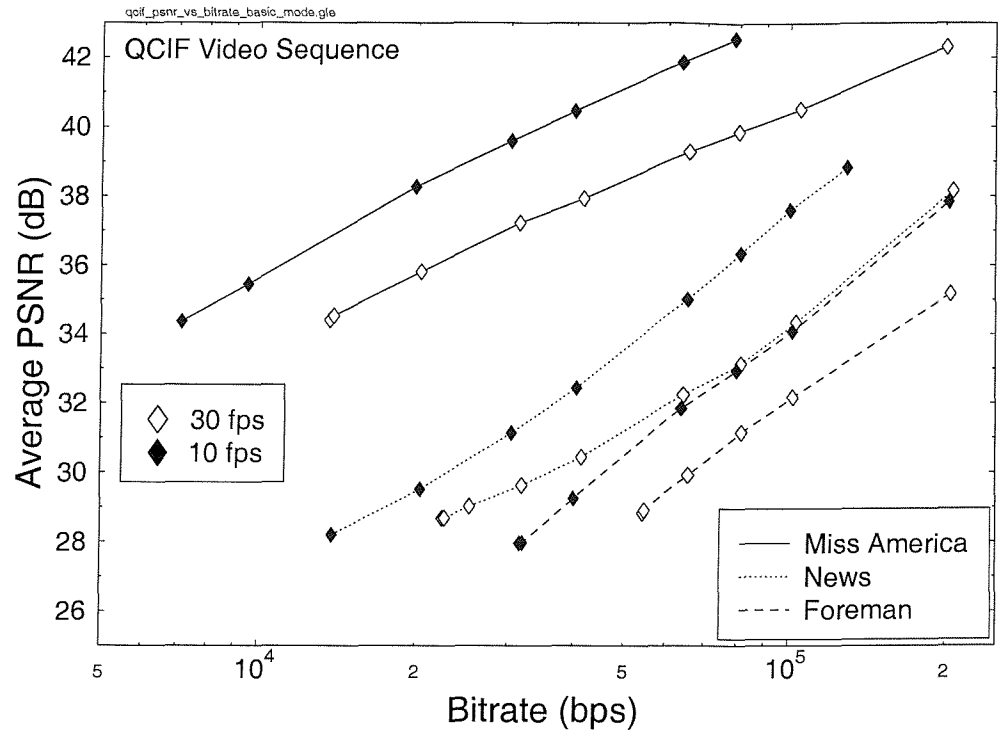


Figure 5.24: Average PSNR image quality versus coded bitrate performance of the MPEG-4 codec for the "Miss America", "News" and "Foreman" video sequences at 10 frames/s and 30 frames/s.

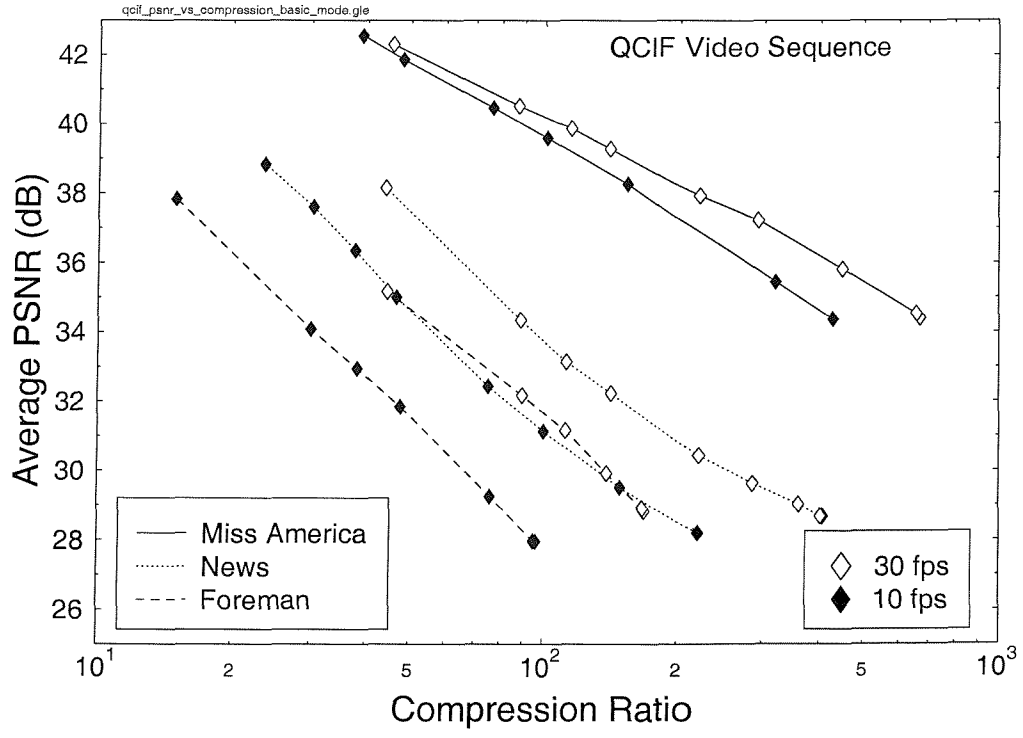


Figure 5.25: Average PSNR image quality versus compression ratio performance of the MPEG-4 codec for the "Miss America", "News" and "Foreman" video sequences at 10 frames/s and 30 frames/s.

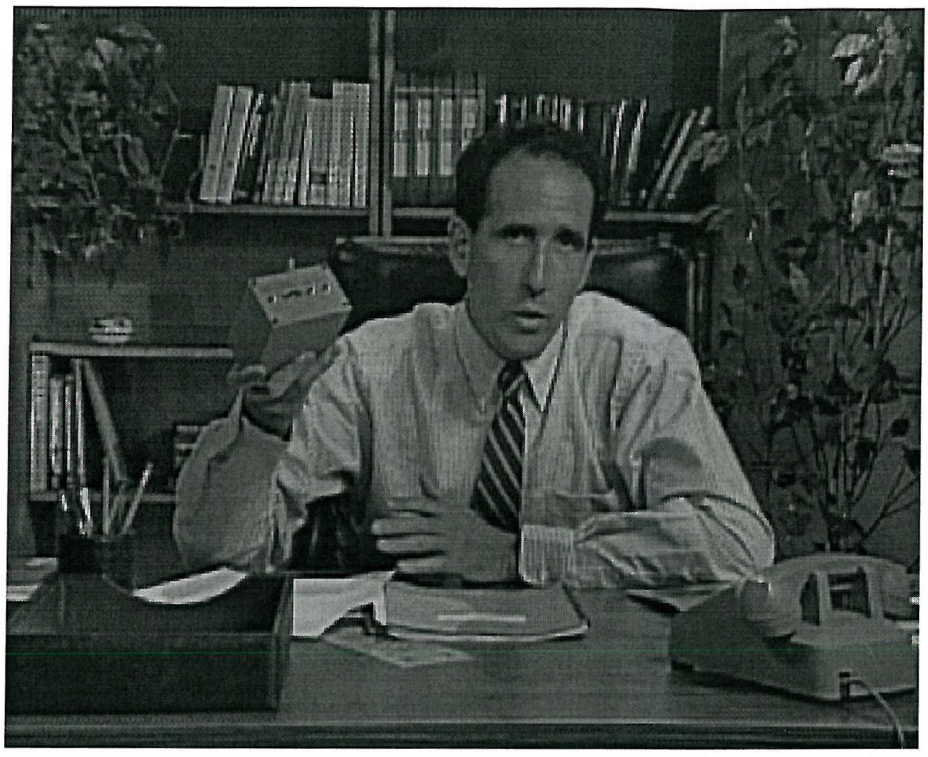


Figure 5.26: First frame of the CIF "Salesman" sequence.

(CIF) video sequences of dimension 352-by-288 pixels, namely the "Miss America", "Salesman" and "Football" video clips, which are sampled at both 10 frames/s and 30 frames/s. The "Miss America" sequence is as shown in Figure 5.21 but at CIF resolution, while the CIF "Salesman" and "Football" sequences are shown in Figure 5.26 and 5.27, respectively. Table 5.2 presents a summary of the MPEG-4 codec's performance when the rate control module is switched off. As before, the finest and coarsest quantiser step sizes are selected in order to enable us to obtain the upper and lower bound of the coded bitrate and compression ratio attainable by the video codec. Again, the "Miss America" sequence exhibits the highest compression ratio of the three sequences considered. Similarly to the "Miss America" sequence, the "Salesman" sequence is also amenable to compression. This property is due to their moderate motion activity and spatial detail. By contrast, the "Football" sequence exhibits a low compression ratio due to its high motion activity and increased spatial detail. Hence a significantly higher coded bitrate is required, even when the coarsest quantiser step size is employed. Specifically, the "Football" sequence requires approximately seven and eight times higher bitrate than the "Miss America" sequence for the 10 frames/s and 30 frames/s scenarios, respectively. Incidentally, the CIF resolution "Miss America" sequence requires an approximately four times higher bitrate than the QCIF resolution "Miss America" sequence. This corresponds to the factor four increase in resolution.

As in the context of QCIF sequences, we then invoked the MPEG-4 encoder's rate control module for encoding the CIF sequences. The associated range of attained coded bitrates and compression ratios is shown in Figure 5.28 and Figure 5.29 respectively. The image quality is seen to vary in a quasi-linear fashion as a function of the coded bitrate

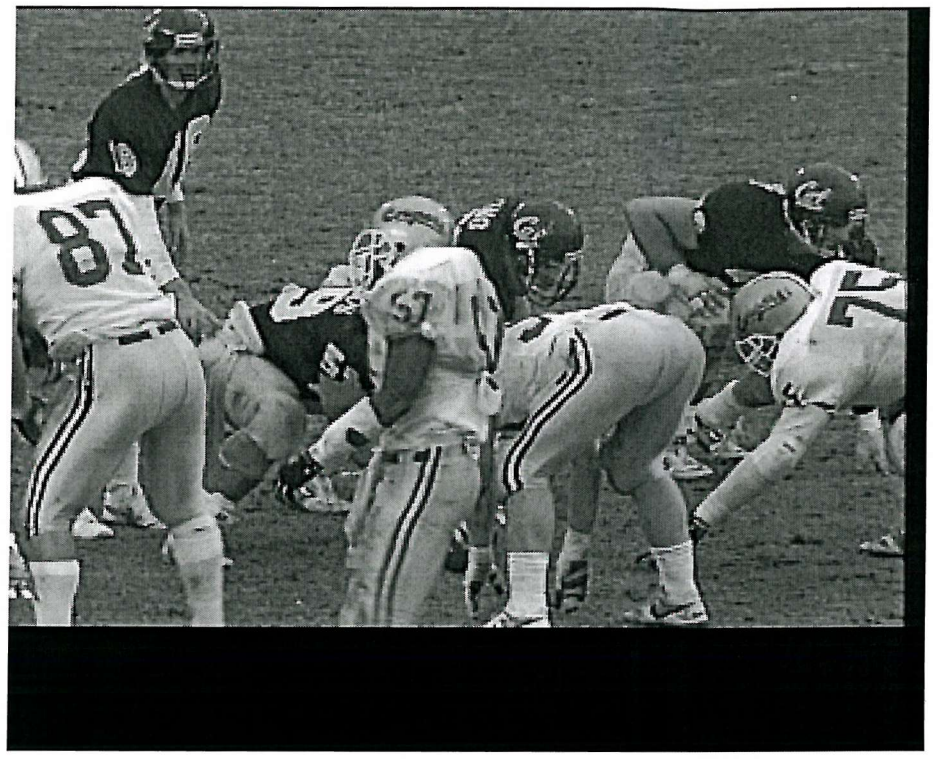


Figure 5.27: First frame of the CIF "Football" sequence.

Sequence	Type	Frame Rate (fps)	Quantiser Step Size	Coded Bitrate (Kbps)	Average Luminance PSNR (dB)	Compression Ratio
Miss America	CIF	10	1	2489.84	47.07	4.89
			31	28.86	35.63	421.56
Salesman	CIF	10	1	2873.03	46.78	4.23
			31	31.80	28.49	382.56
Football	CIF	10	1	4671.66	47.52	2.60
			31	208.77	27.54	58.27
Miss America	CIF	30	1	7324.51	47.05	4.98
			31	53.97	35.61	676.28
Salesman	CIF	30	1	8196.47	46.75	4.45
			31	58.33	28.59	625.68
Football	CIF	30	1	12553.65	47.58	2.91
			31	451.98	27.37	80.75

Table 5.2: Upper and lower bounds of the MPEG-4 codec's performance for various CIF resolution video sequences, where the video encoder employs either the finest or the coarsest quantiser, in order to obtain the highest and lowest quality encoded video, respectively. The rate control was disabled in this experiment.

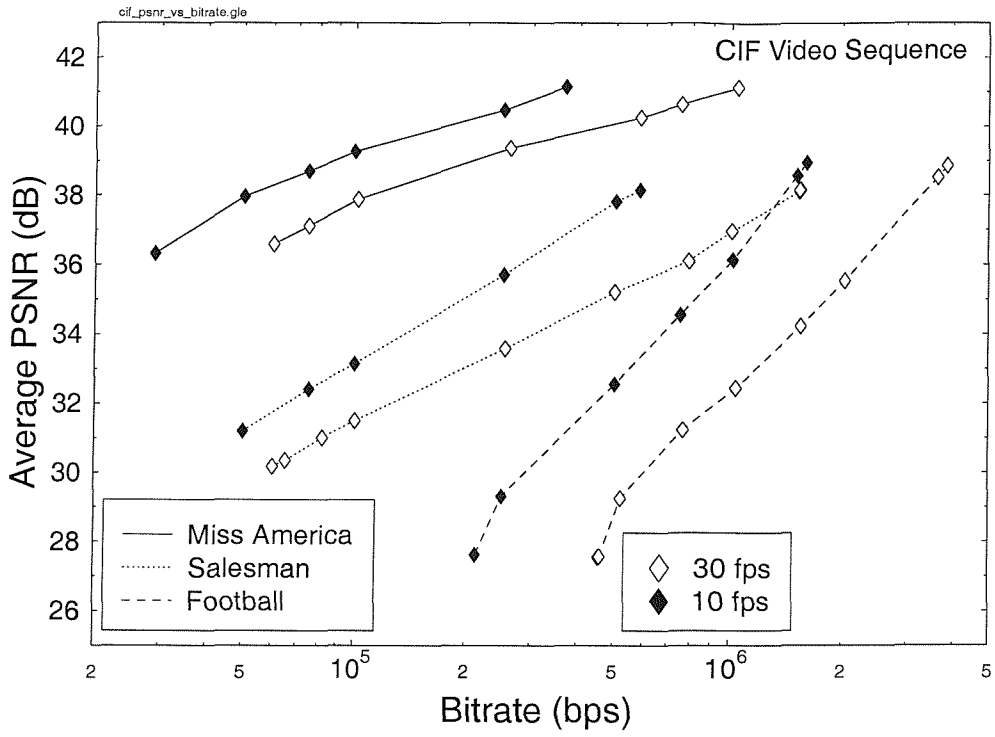


Figure 5.28: Average PSNR image quality versus coded bitrate performance of the MPEG-4 codec for the CIF resolution "Miss America", "Salesman" and "Football" video sequences at 10 frames/s and 30 frames/s.

and compression ratio. In reducing the frame scanning rate from 30 frames/s to 10 frames/s, the coded bitrate achieved is reduced by approximately a factor of two for the same image quality for all three CIF sequences. Overall at a given PSNR, the operating bitrate range of the MPEG-4 codec spans an order of magnitude for these CIF sequences. From Figure 5.29, we note that the "Miss America" clip is significantly more amenable to compression at a given PSNR than the "Salesman" sequence. Another observation, which is valid in the context of both sequences, is that their PSNR appears similar at both scanning rates, although different for the two sequences. This observation is based on the fact that the difference between two consecutive frames is limited for both sequences. However, again, since the "Salesman" sequence contains more spatial detail than the "Miss America" sequence, the PSNR image quality attained is lower for the "Salesman" sequence at the same compression ratio. On the other hand, the "Football" sequence exhibits more significant differences between consecutive frames, since it contains higher motion activity than the other two CIF sequences. This is reflected in the difference between their compression ratios achieved, when the frame scanning rate is decreased by a factor of three.

Lastly, we characterised the performance of the MPEG-4 codec using the "Stefan" sequence, which has a resolution of 720-by-480 pixels. From Table 5.3, we observe that the "Stefan" sequence has a low compression ratio due to its high spatial detail and medium amount of motion activity. When the rate control module is disabled and the

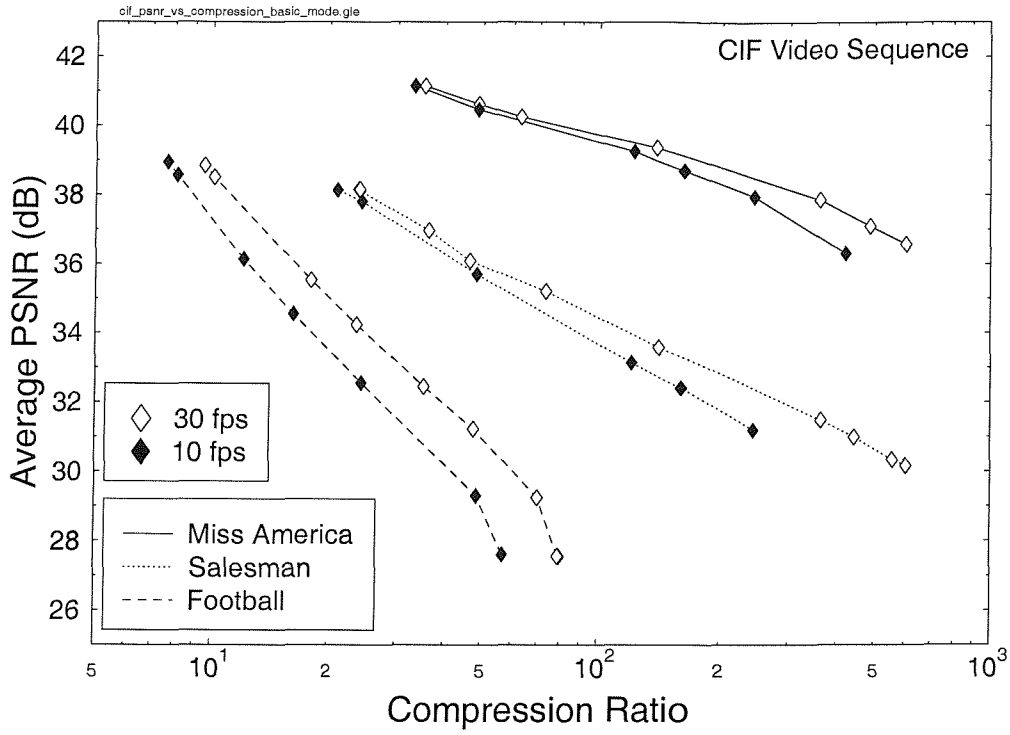


Figure 5.29: Average PSNR image quality versus compression ratio performance of the MPEG-4 codec for the CIF resolution "Miss America", "Salesman" and "Football" video sequences at 10 frames/s and 30 frames/s.

quantiser step size is set to the finest level, the coded bitrate attained is as high as 52 Mbit/s. The operating range of the MPEG-4 codec for this sequence is portrayed in Figure 5.30 and Figure 5.31.

Sequence	Type	Frame Rate (fps)	Quantiser Step Size	Coded Bitrate (Mbps)	Average Luminance PSNR (dB)	Compression Ratio
Stefan	720x480	30	1	52.05	46.97	2.39
			31	2.41	25.45	51.62

Table 5.3: Upper and lower bounds of the MPEG-4 codec performance for "Stefan" sequence at resolution of 720-by-480 pixels. The video encoder employs either the finest or the coarsest quantiser in order to obtain the best and worst quality encoded video. The rate control is disabled in this experiment.

In summary, we note that the MPEG-4 video codec has a wide operating range. For a low activity video sequence, such as the QCIF "Miss America" clip, the video codec can achieve a coded bitrate as low as 6.85 kbit/s. The operating range can also be extended to high bitrates for CIF and larger video sequences. Therefore, the MPEG-4 codec is suitable not only for low bitrate applications, such as videoconferencing over wireless networks, but also can be targeted at video broadcast appplications, such as high definition television broadcast. In the next section, we will concentrate our discussions on the MPEG-4 codec's error resilience features.

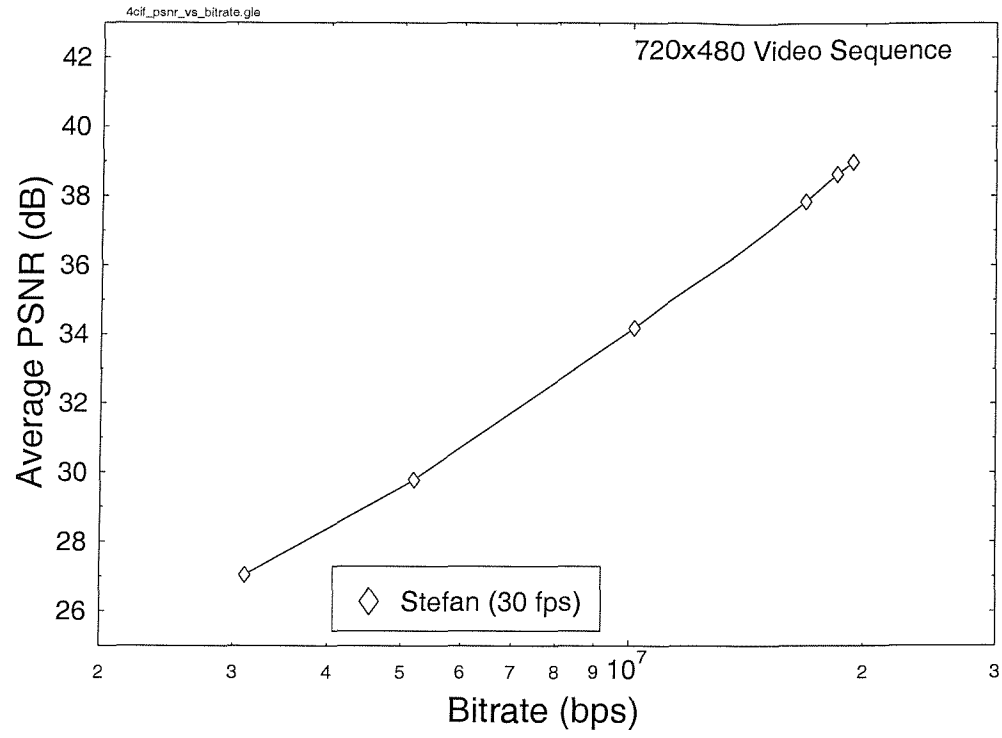


Figure 5.30: Average PSNR image quality versus coded bitrate performance of the MPEG-4 codec for the 720-by-480 resolution "Stefan" sequence at 30 frames/s.

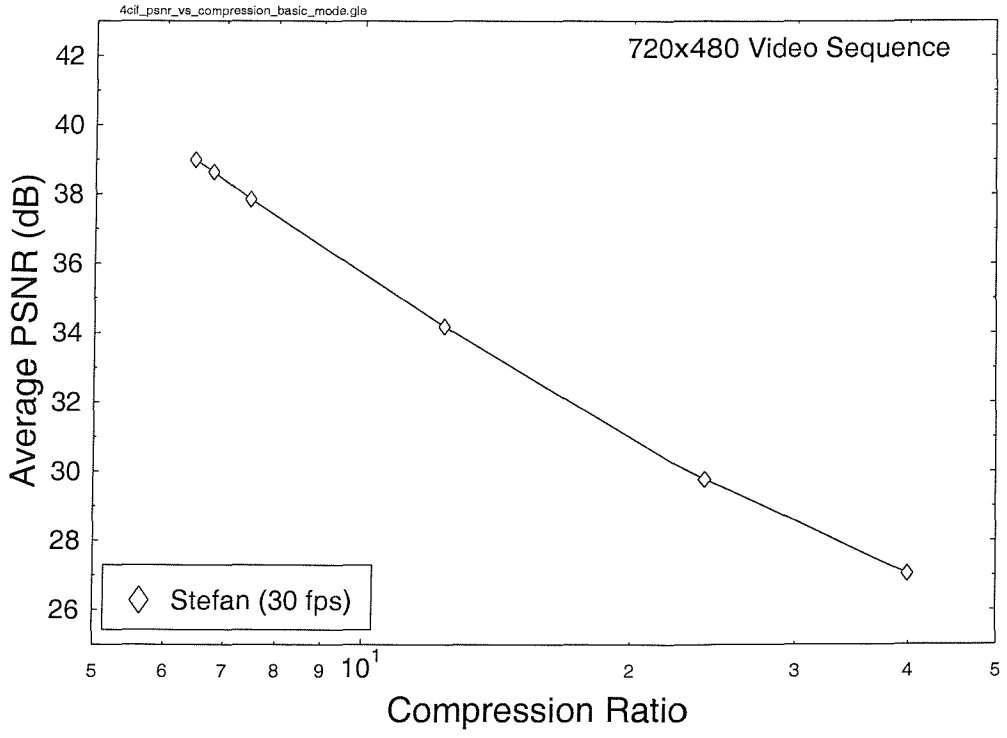


Figure 5.31: Average PSNR image quality versus compression ratio performance of the MPEG-4 codec for the 720-by-480 resolution "Stefan" sequence at 30 frames/s.

5.6 MPEG-4 Error Resilience Features

One of the implications of universal access requirement in MPEG-4, which was discussed in Section 5.2.2, is the possibility of communicating MPEG-4 encoded multimedia signals through heterogeneous communication networks. The propagation conditions of mobile channels are particularly severe due to Rayleigh fading, as argued in Section 1.3. In general, forward error correction codes, such as convolutional codes, Reed-Solomon codes, turbo codes and etc., are used to counteract the effects of channel errors. However, the channel decoders may not be able to correct all the channel errors. This has motivated the MPEG-4 committee to include error resilient features into the MPEG-4 video decoder in order to mitigate the effects of the residual errors after channel decoding. The error resilient features introduced into the MPEG-4 video decoder comprise:

- error detection and localisation
- resynchronisation
- data recovery.

The video decoder can be utilised to detect errors in the video bitstream with the aid of semantic and syntactic checks. Examples of error detection performed by the video decoder are

- more than 64 DCT coefficients are decoded per 8-by-8 pixel block
- decoded motion vectors are outside the allowed range
- invalid variable-length codewords
- wrong coding mode detection for macroblock, such as intra-frame coded pictures containing inter-frame coded blocks.

If the decoder identifies a decoding error, it will curtail decoding and attempts to resynchronise at the next valid decoding start point. At this stage, the decoder may manage to isolate the transmission error to be between two resynchronisation points and may regain synchronisation with the encoder. In the next three sections, we will outline the approaches adopted by the MPEG-4 committee in providing resilience against transmission errors.

5.6.1 Resynchronisation

In the event of a decoding error, the MPEG-4 video decoder regains synchronisation by searching for the next resynchronisation marker in the bitstream. The resynchronisation marker is a unique codeword. Hence, the probability that the decoder confuses it with other codewords is low. This technique is also employed in other video coding standards, such as MPEG-2, H.261 and H.263. In the H.261 and H.263 standards, each image is divided into rows of macroblocks, with each row spanning from the left boundary of the image to its right boundary. Each row of macroblocks is also known as group of blocks (GOB). The resynchronisation marker is inserted at the start of the GOB. Hence, resynchronisation can only occur at the left edge of the image, provided that the resynchronisation marker is not corrupted by channel errors. The smallest region in which transmission errors can be isolated is one row of macroblocks. In the MPEG-2 standard, the group of blocks term is also known as a slice. However, a slice does not have to

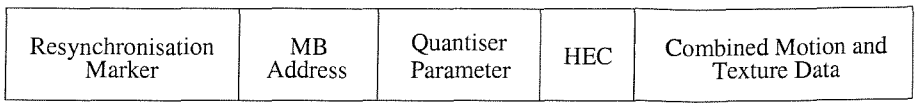


Figure 5.32: MPEG-4 video packet structure employing a resynchronisation marker and the Header Extension Code (HEC), in order to assist the decoder in recovering from its errors.

be constrained to occupy a whole row of macroblocks. One row of macroblocks can be represented by more than one slice.

MPEG-4 provides a similar slice synchronisation scheme as well. It does not restrict the slice to an entire row of macroblocks. Instead, it divides the image into video packets, each packet containing a similar number of bits. When there is significant motion activity in a region of the image, more bits will be used to encode the macroblocks in this region. On the other hand, a low activity image region, such as the still background, can be coded with less bits per macroblock. Hence, the video packet will contain more macroblocks in a low activity region compared to a high activity region. When residual channel decoder errors exist, the video decoder can localise the errors in a high activity region to a smaller number of affected macroblocks. This enables the optional error concealment scheme to operate more effectively.

Apart from searching for the next resynchronisation marker, the decoder also has to remove any data dependencies between data that belongs to different video packets. This is a consequence of differentially encoding information, such as the DC DCT coefficients and motion vectors. The following information is inserted at the start of each video packet, in addition to the resynchronisation marker:

- macroblock address of the first macroblock in the video packet, which indicates the spatial location of this macroblock in the image
- quantisation parameter, which denotes the default quantisation parameter used to quantise the DCT coefficients of the macroblocks within this video packet.

The video decoder will also reset the motion vector predictors, which we discussed in Section 5.4.2, so that prediction is made from within the video packet boundary only. Figure 5.32 portrays the organisation of data within a video packet. Up to now, we have indirectly assumed that the picture header information has not been corrupted. The picture header contains information such as time stamp and picture coding type, which assists the video decoder in successfully decoding the received video data. However, this information can also be corrupted during transmission. Hence, an optional section within the video packet, known as Header Extension Code (HEC), is introduced, which contains information similar to that carried in picture header, in order to allow the information in each video packet to be independently decoded. Furthermore, by comparing the information contained within the video packet header and the picture header, the video decoder can determine the validity of the picture header data. The use of the HEC can reduce the number of dropped video frames, when the picture header is corrupted.

Resynchronisation Marker	MB Address	Quantiser Parameter	HEC	Motion Data	MBM	Texture Data

Figure 5.33: Video packet structure in MPEG-4 employing data partitioning. The term HEC refers to the header extension code and MBM denotes to motion boundary marker.

5.6.2 Data Partitioning

Typically, after the video decoder has detected a decoding error, it is incapable of locating the exact position of the error. This is a consequence of the extensive use of variable-length codewords, in order to achieve a high compression ratio. Hence, the data between the detection of the error and the next resynchronisation point are discarded. Furthermore, since the motion and texture data are encoded jointly within a macroblock, they are both lost in case of transmission errors. The data partitioning scheme used in MPEG-4 was incorporated for reducing the amount of lost data by separating the motion and texture data within the transmitted video packet, as seen in Figure 5.33. The motion data of all the macroblocks belonging to the same video packet are also grouped together. Similarly, the texture data of all the macroblocks belonging to the same video packet are grouped together. The motion section and the texture section are separated by a unique Motion Boundary Marker (MBM). The MBM indicates the end of the motion data and the start of texture data segment in Figure 5.33. The specific bit pattern of the MBM was obtained with the aid of computer search based on generating every possible combination of codewords from the variable-length code tables used for encoding the motion data [103]. The aim of this search was to find a MBM, which is unique, and hence it is not emulated by any combination of codewords from this code table. Again, Figure 5.33 shows the structure of the video packet employing data partitioning. The role of the HEC in Figure 5.33 was highlighted in Section 5.6.1.

If the video decoder detects any error while decoding the motion data, all the macroblocks in the current video packet will be skipped. This is because the effects of motion vector errors propagate for a long time due to the misaligned local and remote reconstruction frame buffers. However, if only the texture data is corrupted, the video decoder will retain the motion data but discards all the texture data. If the MBM was correctly detected, the video decoder has a high confidence that the motion data of Figure 5.33 is correct. Furthermore, if the whole video packet is successfully decoded, but the next resynchronisation marker is missing, the video decoder will discard all the texture data belonging to the current video packet, which preceeded the lost marker. This action is favoured because the video decoder is confident of the correctness of the motion data, since the MBM is detected, but the texture data is likely to be corrupted.

5.6.3 Reversible Variable Length Codes

In the previous section we noted that the variable-length encoded texture data is discarded until the next resynchronisation point is found, when a decoding error is detected. The data is removed, irrespective of whether it has actually been corrupted, since the decoder loses synchronisation when an earlier error renders the subsequent variable-length

codewords undecodable. It would be beneficial if we could reduce the likelihood of the texture data being discarded. Reversible Variable Length Codes (RVLC) [104, 105] were adopted by the MPEG-4 codec with this motivation in mind. These RVLCs have the specific property that they are decodable either in the forward or backward direction. When the video decoder detects an error while decoding in the forward direction, it will halt the forward decoding process and starts decoding in the reverse direction, once the next resynchronisation point was found. Hence, some of the texture data at the end of the packet seen in Figure 5.33, which would normally be discarded can be salvaged. In order for the RVLC scheme to be effective, all the data that is encoded using RVLC must occur together. Hence, this scheme is usually invoked together with the data partitioning scheme.

Although the MPEG-4 standard does not specify the behaviour of the video decoder, when the RVLC scheme is utilised, Talluri [85] has provided an attractive strategy for decoding the video bitstream employing RVLCs. His suggestion is repeated here for the reader's convenience. Consider this example with the assistance of Figure 5.34, where the variables used are as follows:

L:	Total number of bits used for encoding the DCT coefficients in a video packet
N:	Total number of macroblocks in a video packet
L1:	Number of bits which can be decoded in the forward direction
L2:	Number of bits which can be decoded in the backward direction
N1:	Number of macroblocks which can be completely decoded during forward decoding
N2:	Number of macroblocks which can be completely decoded during backward decoding
f_mb(S):	Number of decoded macroblocks when S bits can be decoded in the forward direction
b_mb(S):	Number of decoded macroblocks when S bits can be decoded in the backward direction
T:	Threshold (Recommended value for MPEG-4 is 90).

The bitstream is decoded in the forward direction first. If no decoding error is detected, the decoding process is successful and the next video packet will be decoded. If a decoding error occurs in the forward decoding process, the decoder will search up to the end of the current video packet and commences backward decoding, until an error is detected. The next step is to decide on the amount of information to discard. There are four possibilities to be considered, which are as follows:

- **(L1 + L2 < L) and (N1 + N2 < N):** The f_mb(L1 - T) number of macroblocks from the start and b_mb(L2 - T) number of macroblocks from the end of the video packet are retained.
- **(L1 + L2 < L) and (N1 + N2 ≥ N):** The (N - N2 - 1) number of macroblocks from the start and the (N - N1 - 1) number of macroblocks from the end of the video packet are retained.

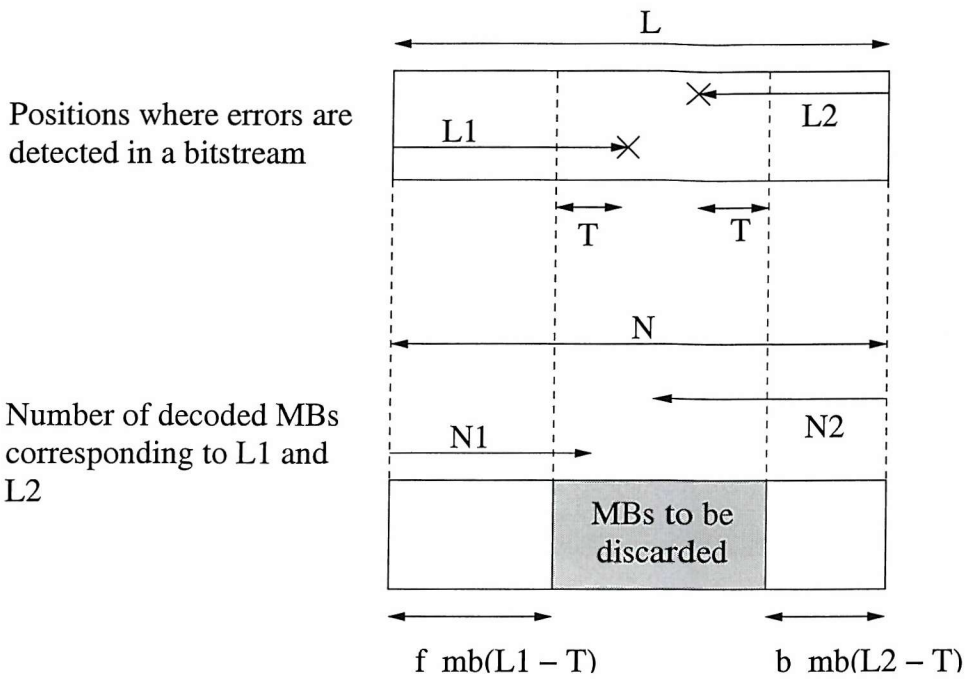


Figure 5.34: Example of a possible decoding strategy for the MPEG-4 bitstream employing reversible variable-length codes.

- **(L1 + L2 ≥ L) and (N1 + N2 < N):** The (N - b_mb(L2)) number of macroblocks from the start and the (N - f_mb(L1)) number of macroblocks from the end of the video packet are retained.
- **(L1 + L2 ≥ L) and (N1 + N2 ≥ N):** The $min\{$(N - b_mb(L2)), (N - N2 - 1)$\}$ number of macroblocks from the start and the $min\{$(N - f_mb(L1)), (N - N1 - 1)$\}$ number of macroblocks from the end of the video packet are retained. The operator $min\{a, b\}$ returns either the value a or b, depending on which one has the smaller value.

In our ensuing section, we will characterise the performance of the MPEG-4 error resilience tools within the context of a DVB terrestrial transmission scenario.

5.6.4 Performance of the MPEG-4 Error Resilience Tools Within DVB Terrestrial Transmission Scenario

In this section we present the results of our study of the MPEG-4 error resilience tools within the context of a DVB terrestrial transmission scenario. The DVB terrestrial system has been discussed in Chapter 2. The "Salesman" and "Stefan" video sequences are used in our experiments. The "Salesman" clip has a dimension of 352-by-288 pixels, while the "Stefan" sequence has a dimension of 720-by-480 pixels. Figure 5.26 and Figure 5.35 show the first frame of the "Salesman" and "Stefan" video sequences, respectively.

Both video sequences are encoded using the MPEG-4 video encoder and the coding parameters are summarised in Tables 5.4 and 5.5. For the encoded video sequences employing only resynchronisation markers but no other error resilience enhancements, direct copying of the affected image region from the previously reconstructed video frame is performed in the event of a decoding error. On the other hand, this concealment

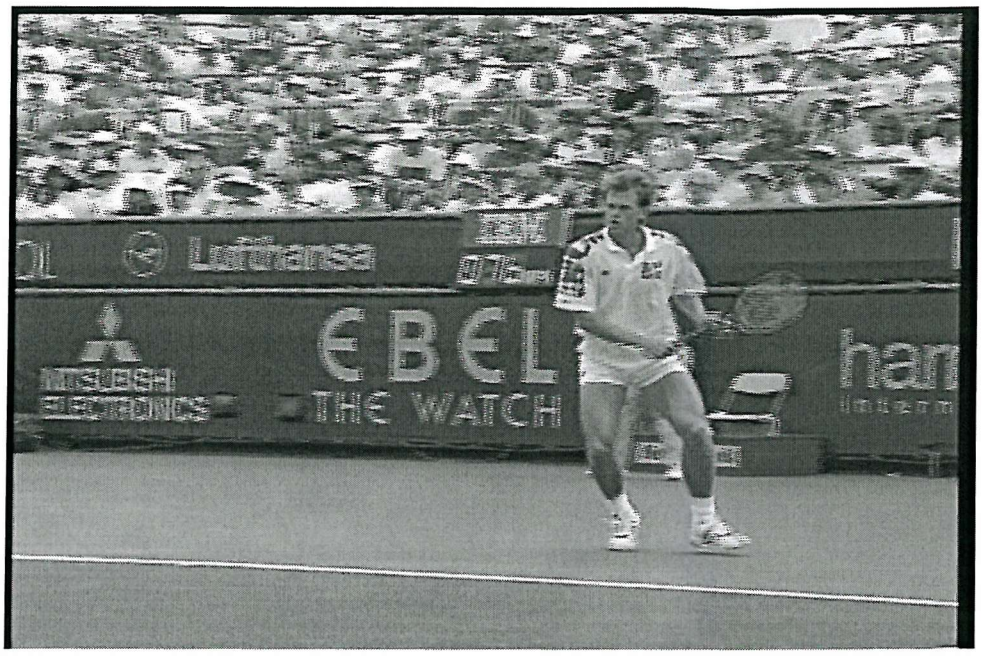


Figure 5.35: First frame of the 720-by-480 pixel "Stefan" video sequence.

method is invoked only when the specific part of the video packet, which contains motion information is corrupted. However, when data partitioning is used, then the lost section of the video frame is replenished from the previous frame, even if only the texture information was corrupted. If only the texture portion of the video packet is corrupted, the motion information is still utilised, while the texture information is removed.

"Salesman" Coding Parameters	
Image Resolution	352 × 288
Target Bitrate	1 Mbps
Total Frames	150
Frame Rate	30 fps
Motion Vector Search Range	16 × 16 pixels
Intra-Frame Refresh Period	15 or 30 frames
Video Packet Size	1500 bits

Table 5.4: Coding parameters for the CIF "Salesman" video sequence.

"Stefan" Coding Parameters	
Image Resolution	720 × 480
Target Bitrate	5 Mbps
Total Frames	120
Frame Rate	30 fps
Motion Vector Search Range	16 × 16 pixels
Intra-Frame Refresh Period	15 frames
Video Packet Size	1500 bits

Table 5.5: Coding parameters for the "Stefan" video sequence.

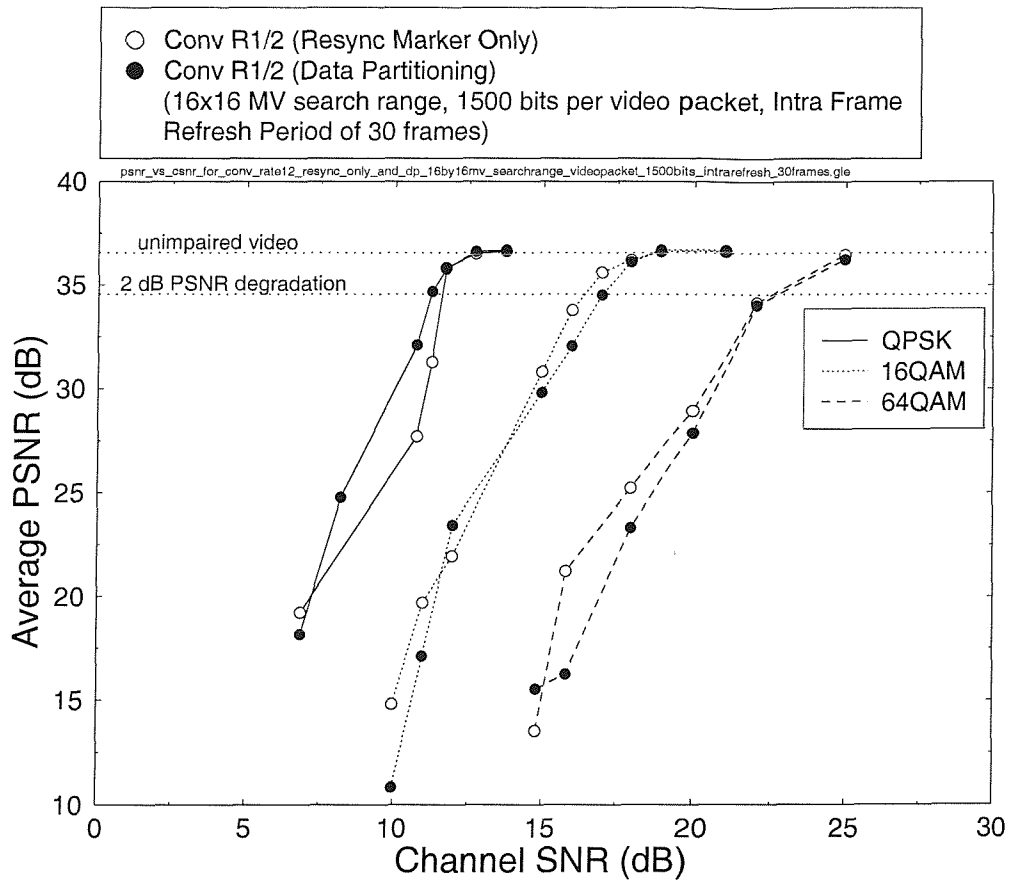


Figure 5.36: PSNR versus channel SNR performance for the MPEG-4 encoded "Salesman" sequence with intra-frame refresh updates every 30 frames. The encoded video employs low-complexity resynchronisation marker assisted error recovery. The DVB-T system employs rate 1/2 convolutional code as its inner code, and an outer Reed-Solomon code to form a concatenated coding scheme. The QPSK, 16-QAM and 64-QAM based transmissions were used.

Figure 5.36 shows a comparison of the performance of the data partitioned scheme and the low-complexity resynchronisation marker assisted scheme within a DVB terrestrial transmission scenario employing a rate 1/2 convolutional code as its inner code. The intra-frame refresh period used for this experiment was 30 frames. The "Salesman" video sequence was used. We note that the data partitioned scheme outperforms the low-complexity resynchronisation marker assisted scheme in the transmission scenario, which employs Quadrature Phase Shift Keying (QPSK) modulation. At the channel SNR of 11.2 dB, the data partitioned scheme outperformed the low-complexity resynchronisation marker assisted scheme by 3.4 dB in terms of the achievable PSNR. The 2 dB PSNR degradation threshold was selected as the maximum tolerable objective quality drop, because the viewer will begin to notice the degradation of the decoded video quality at this level. However, when employing the 16-QAM and 64-QAM modulation modes, respectively, the low-complexity resynchronisation marker assisted scheme consistently performed better than the data partitioned scheme. This observation can be explained as follows. Each video packet which utilises the data partitioning tool can be divided into

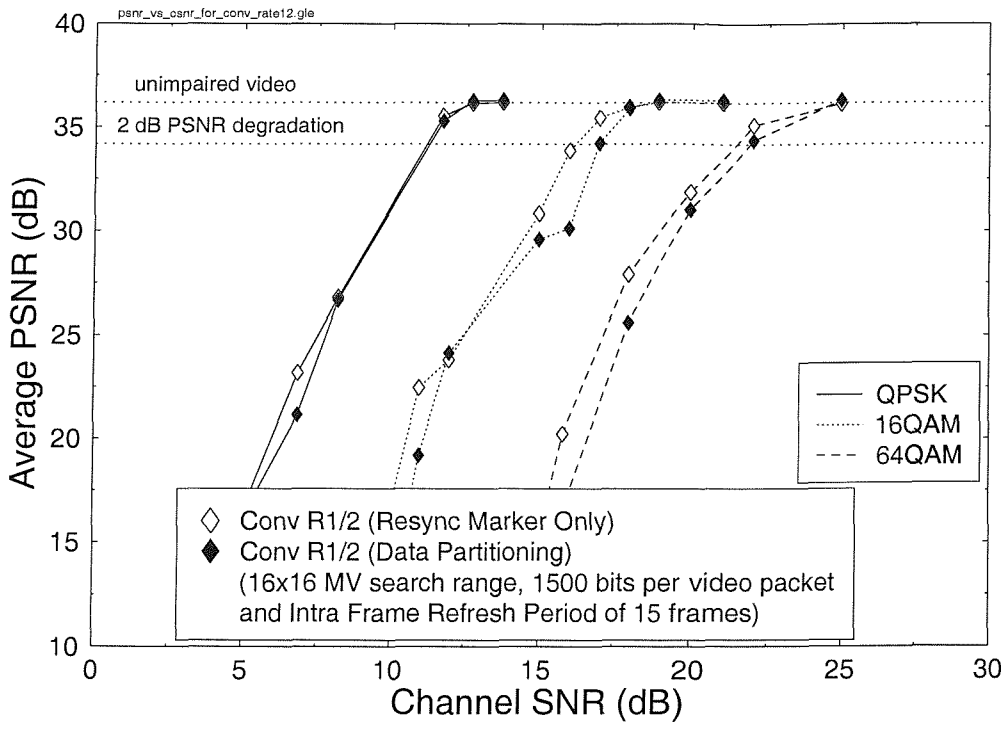


Figure 5.37: PSNR versus channel SNR performance for the MPEG-4 encoded "Salesman" sequence with intra-frame refresh update every 15 frames. The encoded video employs low-complexity resynchronisation marker assisted and data partitioning based error resilience tools. The DVB-T system employs rate 1/2 convolutional code as its inner code, and an outer Reed-Solomon code to form a concatenated coding scheme. The QPSK, 16-QAM and 64-QAM based transmissions were used.

two main sections, namely motion information and texture information as suggested by the packetisation scheme seen in Figure 5.33. The effectiveness of the data partitioned scheme relies on the fact that channel errors will affect the texture information more than motion information, since the accumulated probability of bit errors increases towards the end of the transmission packet. If this condition holds, the data partitioned scheme will provide better error resilience, than the resynchronisation assisted scheme, since the motion information can be used to provide better error concealment, especially in the image areas where high motion activity is registered. By contrast, if errors occur towards the beginning of the video packet, corrupting the motion information, this results in erroneous motion vector predictions and prolonged video degradations. It should also be noted that in our experiments the inter-frame coded pictures do not contain any intra-frame coded macroblocks. Intra-frame coded macroblocks are restricted to intra-frame coded frames only. The previously mentioned efficiency criterion of the data partitioned scheme is met by the QPSK modulated transmission system in this particular scenario. For the 16-QAM and 64-QAM systems, however, this criterion is not met, because the modulation modes in these cases are less robust than QPSK modulation, when employing the same type of channel codes. Hence the probability of motion data errors is increased, which result in prolonged motion prediction errors.

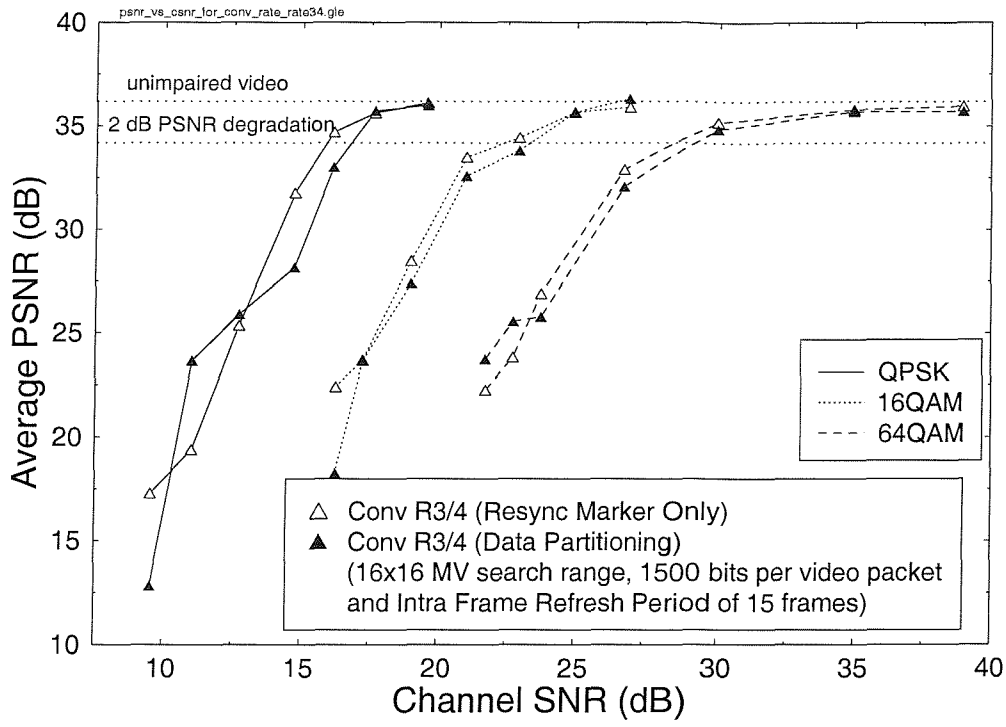


Figure 5.38: PSNR versus channel SNR performance for the MPEG-4 encoded "Salesman" sequence with intra-frame refresh update every 15 frames. The encoded video employs low-complexity resynchronisation marker assisted and data partitioning based error resilience tools. The DVB-T system employs rate 3/4 convolutional code as its inner code, and an outer Reed-Solomon code to form a concatenated coding scheme. The QPSK, 16-QAM and 64-QAM based transmissions were used.

The same set of experiments were then repeated for the "Salesman" video sequence. However, the intra-frame refresh period was reduced from 30 frames to 15 frames. The comparison of the performance between the low-complexity resynchronisation marker assisted scheme and the data partitioned scheme is shown in Figure 5.37. With more frequent intra-frame updates, the error propagation can be curtailed every 15-frame intra-frame update period. This observation is made in Figure 5.37, where the performance of the low-complexity resynchronisation marker assisted scheme is similar or better than that of the data partitioned scheme.

We have also made similar observations for the identically encoded "Salesman" video sequence, when transmitted over a DVB terrestrial system employing rate 3/4 and rate 7/8 convolutional codes, respectively. The associated results are shown in Figure 5.38 and Figure 5.39. The intra-frame refresh period was also set to 15 frames.

Figure 5.40 and Figure 5.41 show the results of our experiments using the "Stefan" video sequence of Figure 5.35. The intra-frame refresh period was maintained at 15 frames. For the case where each video packet size was set to 1500 bits, the low-complexity resynchronisation marker assisted scheme outperformed the data partitioned scheme by approximately 3 to 4 dB in terms of PSNR. The PSNR improvement for the "Stefan" sequence was higher than for the "Salesman" sequence, because the "Stefan" sequence

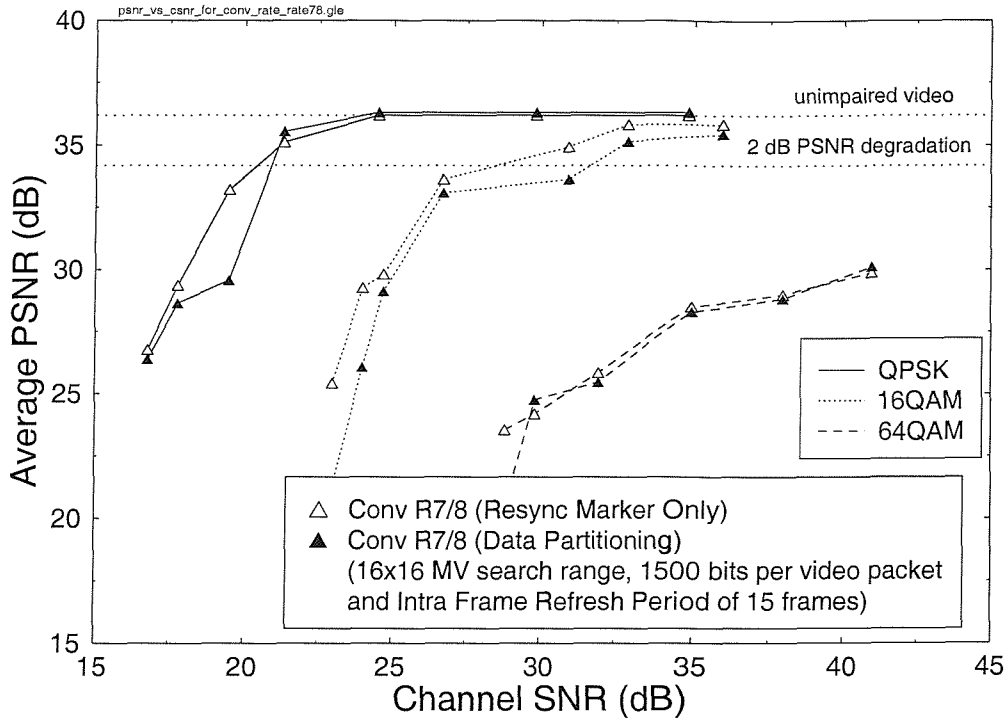


Figure 5.39: PSNR versus channel SNR performance for the MPEG-4 encoded "Salesman" sequence with intra-frame refresh update every 15 frames. The encoded video employs low-complexity resynchronisation marker assisted and data partitioning based error resilience tools. The DVB-T system employs rate 7/8 convolutional code as its inner code, and an outer Reed-Solomon code to form a concatenated coding scheme. The QPSK, 16-QAM and 64-QAM based transmissions were used.

exhibits a higher motion activity than the "Salesman" sequence. Since the data partitioned scheme relies on most of the encoded motion information to be correctly decoded in order for it to be effective, its performance deteriorates for the "Stefan" sequence. A higher motion activity would translate into more motion information being encoded into the video bitstream. As a result, the motion information has a higher probability of being corrupted by channel errors. Furthermore, the direct copying based concealment algorithm, which we employed in our experiments, will perform poorly and hence the concealed region will show obvious discontinuities with respect to other decoded regions of the picture.

Overall, we found that the low-complexity resynchronisation marker based scheme outperformed the data partitioned scheme within the investigated DVB transmission scenario. The data partitioned scheme relies on the high probability of correct motion information in the encoded video bitstream in order for it to be effective. Within a mobile DVB transmission scenario, this condition is not guaranteed. Hence, the data partitioned scheme has inflicted further deterioration of the decoded video quality in this transmission scenario, due to the error propagation engendered by the corrupted motion information.

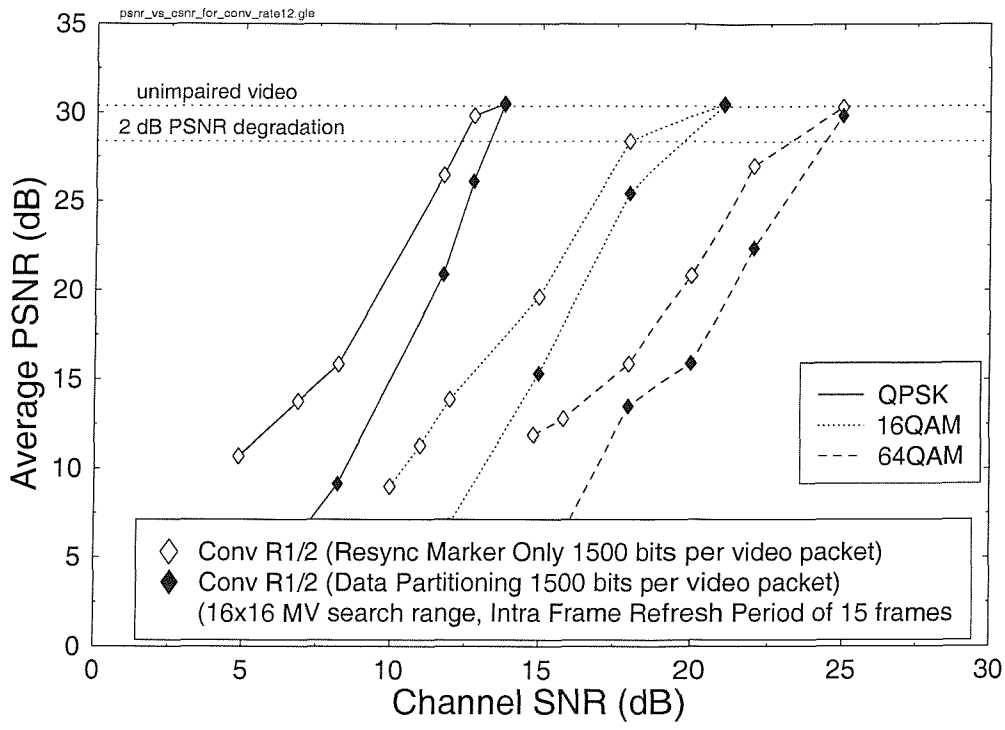


Figure 5.40: PSNR versus channel SNR performance for the MPEG-4 encoded "Stefan" sequence with intra-frame refresh update every 15 frames. The encoded video employs low-complexity resynchronisation marker assisted and data partitioning based error resilience tools. The DVB-T system employs rate 1/2 convolutional code as its inner code, and an outer Reed-Solomon code to form a concatenated coding scheme. The QPSK, 16-QAM and 64-QAM based transmissions were used.

5.6.5 Packet Based MPEG-4 Video Data Transmission

The MPEG-4 error resilience tools were designed with the aim of enabling the video decoder to cope with residual channel errors left uncorrected by the channel decoder. However, the error resilience tools provided have several drawbacks:

- If a valid codeword is corrupted to another valid codeword, there is a possibility that the video decoder fails to detect it, because the semantics and syntax of the bitstream are correct.
- The data partitioning approach aims to utilise the motion information salvaged from an erroneous video packet. Research has shown that error concealment using motion information achieves a better reconstructed picture quality than the simple direct copying approach [184]. However, this motion information assisted scheme is only effective, if the following conditions apply:
  1. There is no frame loss since this would results in motion discontinuity.
  2. The correct motion information is always available. This implies that errors affect only the texture information. Otherwise, the use of the recovered motion information will lead to error propagation.
- If the video frame region, which is used for prediction was actually formed with the aid of error concealment from the previous reconstructed picture, the video

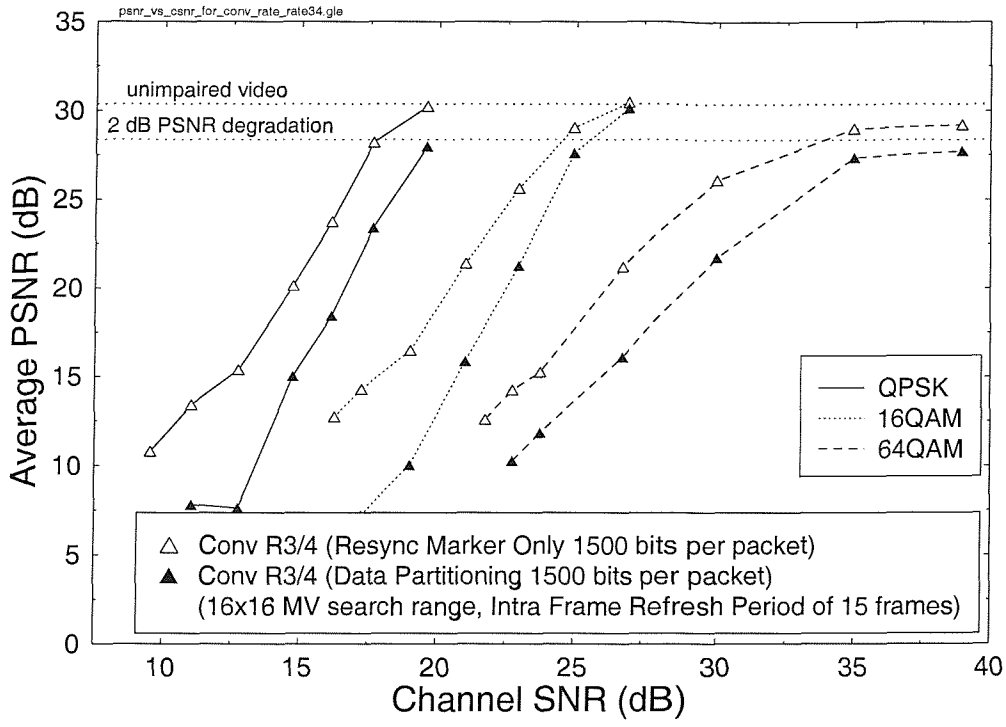


Figure 5.41: PSNR versus channel SNR performance for the MPEG-4 encoded "Stefan" sequence with intra-frame refresh update every 15 frames. The encoded video employs low-complexity resynchronisation marker assisted and data partitioning based error resilience tools. The DVB-T system employs rate 3/4 convolutional code as its inner code, and an outer Reed-Solomon code to form a concatenated coding scheme. The QPSK, 16-QAM and 64-QAM based transmissions were used.

decoder faces the problem of adding prediction errors for the current macroblock to the wrong predicted region. This will result in a phenomenon referred to as colour bleeding in the current reconstructed picture.

Let us now discuss a low-complexity packetisation scheme, which we have proposed in order to circumvent the first problem mentioned earlier. We have also adopted the content-based intra-coded macroblock update scheme proposed by Chen and Hwang [110] in order to limit the error propagation problem caused by using an error impaired region from the previous reconstructed picture for motion prediction.

As seen in Figure 5.42, the transmission packet comprises a packet header and the packet's payload. The packet header contains two indices. The first index points to the start of the first independently decodable, i.e. self-contained macroblock within the transmission packet. The second index points to the end of the last self-contained, i.e. complete decodable macroblock within the transmission packet. Again, Figure 5.42 shows our low-complexity transmission packet structure. These transmission packets are protected by channel codes prior to transmission. If any of these packets contain errors, which could not be corrected by the channel decoder, the packets are dropped. Since the next transmission packet may contains a fraction of a macroblock that can be successfully decoded, the video decoder will always be presented with the error-free bitstream

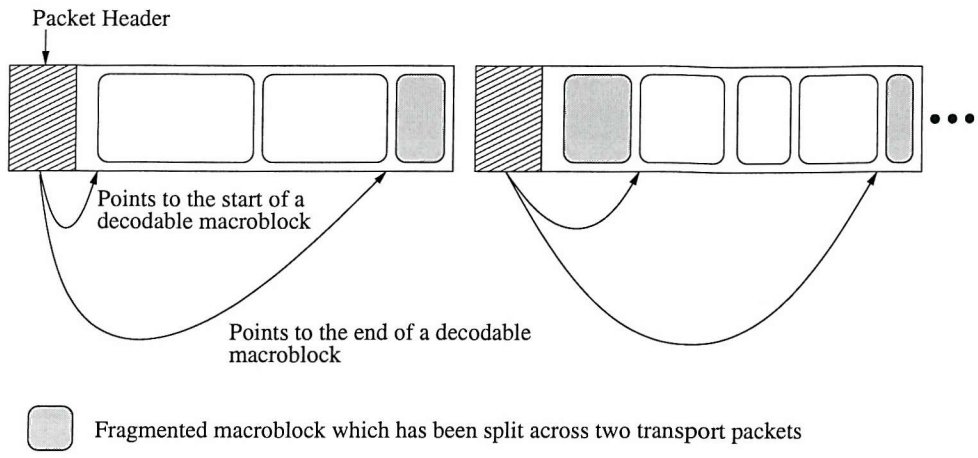


Figure 5.42: Low-complexity transport packet structure for MPEG-4 video data.

portion, in order to attempt its video decoding. Hence, the problem of presenting the video decoder with undetectable errors is avoided.

The MPEG-4 video packet's data can be mapped to the transmission packet as follows. The first index of the transmission packet will always point to the start of the MPEG-4 video packet in order to allow subsequent video data to be independently decoded from the video data of the previous transmission packet. The MPEG-4 encoder will attempt to fit as many complete macroblocks' data into one transmission packet, as possible. The last macroblock which cannot be completely fitted into the current transmission packet will be split across two transmission packets. This so-called partial macroblock cannot be decoded, if part of its data is lost due to transmission packet dropping. Hence, the first index of the transport packet will always point to the start of a MPEG-4 video packet. In our DVB scenario, our transmission packet size is defined to be 187 bytes in order to fit one transmission packet into one DVB transmission packet, which can carry 187-byte of data. Our transmission packet header is allocated 22 bits and the packet payload is allocated a total of 1474 bits.

However, our proposed packetisation scheme alone is incapable of increasing the robustness of the video decoder. The problem of using the wrong video frame region for motion prediction due to repetition-based error concealment still exists because packet dropping will result in the region of the picture carried by the packet concerned being concealed by the video decoder. In order to mitigate this problem, the adaptive intra-coded macroblock update scheme proposed by Chen and Hwang [110] was adopted, which will be highlighted during our forthcoming discourse.

Figure 5.43 will be used to assist us in highlighting the adaptive intra-coded macroblock update scheme of Chen and Hwang [110]. Frame k refers to the current frame to be encoded and Frame $(k-1)$ and Frame $(k-2)$ are the previous reconstructed frames. The current macroblock to be encoded in Frame k is denoted as $MB_k(x,y)$, where (x,y) refers to its location within the frame. Basically, the video encoder has a choice of encoding $MB_k(x,y)$ in either intra-coded mode or inter-coded mode. First, motion estimation is performed for $MB_k(x,y)$ and its best match in Frame $(k-1)$ is labelled as $MB_{k-1}(x',y')$.

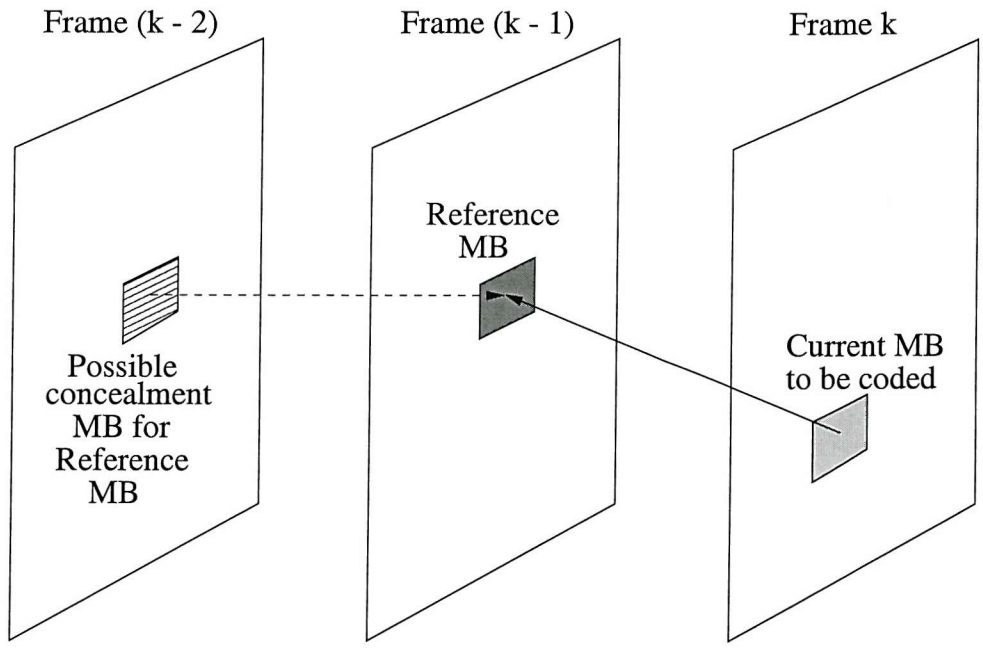


Figure 5.43: Relationship between the current macroblock to be encoded, the reference macroblock and the potential concealment macroblock to be used for concealing the reference macroblock, if it is lost due to transmission packet dropping.

However, $MB_{k-1}(x', y')$ may get lost due to transmission packet dropping as a result of channel errors. Consequently, the video decoder may have to conceal the lost macroblock with the aid of $MB_{k-2}(x', y')$, if the direct copying based method of error concealment is employed. This action would result in error propagation, if the wrong macroblock, namely the concealment based macroblock was used for prediction. Chen and Hwang showed that a measure of the mismatch between the correct macroblock $MB_{k-1}(x', y')$ used for prediction and the incorrect macroblock $MB_{k-2}(x', y')$, which would be used for motion prediction in case of concealment, can be invoked by the encoder to select either the intra coding mode or the inter coding mode for the current macroblock $MB_k(x, y)$. The measure of mismatch between the above two macroblocks used by the encoder is the average of the absolute pixel value differences between the above two macroblocks, namely the correct prediction macroblock and the incorrect prediction macroblock. If the mismatch is above a certain threshold, the current macroblock is encoded in intra-coded mode in order to ensure that even if transmission packet dropping and concealment occurs, no dramatic video quality degradation is perceived. Otherwise, it is encoded in inter-coded mode. The authors of [110] have shown experimentally that a mean absolute difference threshold value of 10 is adequate in providing a trade-off between compression efficiency and error resilience. Intuitively, we note that if a macroblock has a high correlation with the macroblock used to conceal it in the event of transmission packet loss, the mismatch measure will register a small value. In this case, the video decoder achieves a good concealment, and the fact of using concealment will not be perceived by the viewer. Hence, the detrimental effects of error propagation are minimum. By contrast, if the

mismatch between the corresponding macroblocks is high, then the effects of error propagation will be noticeable. Since we are studying a broadcast system, such as DVB, there is no feedback channel from the video decoder informing the video encoder of this problem. Hence, intra-coded macroblocks are used to curtail the error propagation problem. In summary, the above adaptive intra-coded macroblock update scheme proposed by Chen and Hwang [110] was employed in our system for controlling the coding mode selection process by the video encoder in an intelligent fashion. Due to the dependence of this scheme on the content of the video in selecting the macroblock coding mode, it is also referred to as Content-Based Error Resilient Coding (CBERC) scheme.

We will discuss the performance of this scheme in the context of DVB terrestrial transmission in the next section. We will also compare its performance to the previously discussed MPEG-4 error resilience tools.

5.6.6 Performance of the Packet Based MPEG-4 Video Data Transmission

Below we characterise the performance of our proposed packetisation scheme within the context of the DVB terrestrial transmission scenario, which we discussed in Chapter 2. A comparison will also be made between the proposed scheme and the MPEG-4 error resilience tools. The "Salesman" and "Stefan" video sequences are used again in our experiments here. The coding parameters used for both video sequences were shown in Tables 5.4 and 5.5. Figures 5.26 and 5.35 showed the first frame of the "Salesman" and "Stefan" video sequences, respectively. Figure 5.44 portrays the transmission packet loss ratio versus channel SNR characteristics for the DVB terrestrial transmission scenario, showing that as the channel SNR decreases, the transport packet loss ratio increases.

Let us first concentrate on the performance of our proposed packetisation scheme in comparison to the low-complexity MPEG-4 resynchronisation marker assisted scheme, where the previously used rate 1/2 convolutional code was employed. The intra-frame refresh period was 30 frames. For the system that utilises QPSK modulation in Figure 5.45, it is interesting to note that the proposed scheme outperformed the MPEG-4 resynchronisation assisted scheme by 3.3 dB in terms of the average video PSNR. Again, a maximum 2 dB average PSNR degradation was selected, because at this threshold the viewer will begin to notice the transmision error induced artifacts in the reconstructed video. Figure 5.46 shows the comparison between the reconstructed Frame 31 of the "Salesman" video sequence, when the low-complexity resynchronisation marker assisted scheme is invoked and when the proposed packetisation and adaptive intra-coded macroblock update based scheme is used. The artifacts seen in Figure 5.46(a) are the result of channel induced corruption to the quantiser parameter, which was not detected by the video decoder, because the corruption changes one valid codeword to another valid codeword. However, the same problem is not faced by the adaptive intra-coded macroblock update scheme, because the video decoder is presented with only correct information to decode. By contrast, in the 16-QAM system of Figure 5.45 the performance of both error resilience enhancement schemes is comparable. The use of packetisation and adaptive intra-coded macroblock updates do not always guarantee a higher reconstructed video PSNR. In order for the adaptive intra-coded macroblock update scheme to be effective

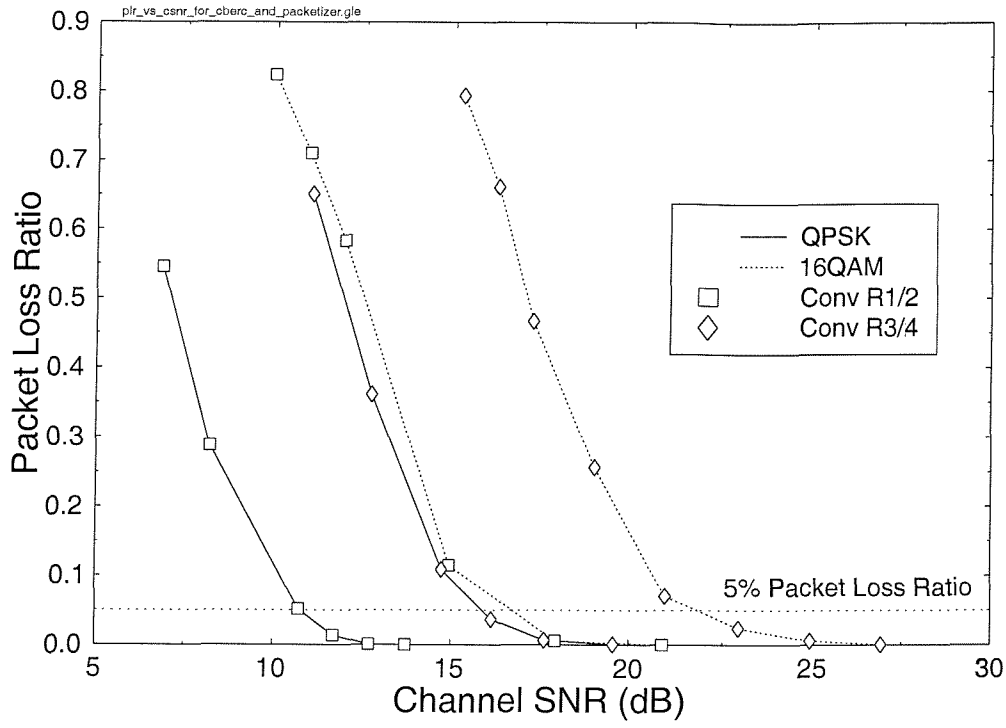


Figure 5.44: Transport packet loss ratio versus channel SNR for the DVB terrestrial transmission scenario, using rate 1/2 and rate 3/4 convolutional codes employed by the DVB terrestrial system. The convolutional code is concatenated to an outer Reed-Solomon code.

in curtailing error propagation, the corresponding intra-coded macroblock updates have to reach the video decoder safely. If the dropped transmission packets contain these intra-coded macroblock updates, the video decoder will be unable to mitigate the error propagation and the quality of the reconstructed video will deteriorate.

The same experiment was repeated, but with the rate 1/2 convolutional code changed to a rate 3/4 convolutional code. The packetisation and adaptive intra-coded macroblock update scheme consistently outperformed the low-complexity resynchronisation marker assisted scheme for both the QPSK and 16-QAM scenarios. The PSNR improvement is approximately 1.6 dB and 1.9 dB for the QPSK and 16-QAM schemes, respectively, as seen in Figure 5.47.

At the next stage of our experiments, the intra-frame refresh period was reduced from 30 frames to 15 frames. Figures 5.48 and 5.49 show the performance of the packetisation plus the adaptive intra-coded macroblock update scheme against that of the resynchronisation marker assisted scheme for the "Salesman" video sequence in this particular scenario. We note that in conjunction with more frequent intra-frame updates, the advantage of the adaptive intra-coded macroblock update scheme is diminished.

Figures 5.50 and 5.51 show our comparison of the performance between the resynchronisation marker assisted scheme and the packetisation plus adaptive intra-coded macroblock update scheme for the "Stefan" video sequence. The packetisation plus adaptive intra-coded macroblock update scheme shows consistent improvements in all the cases

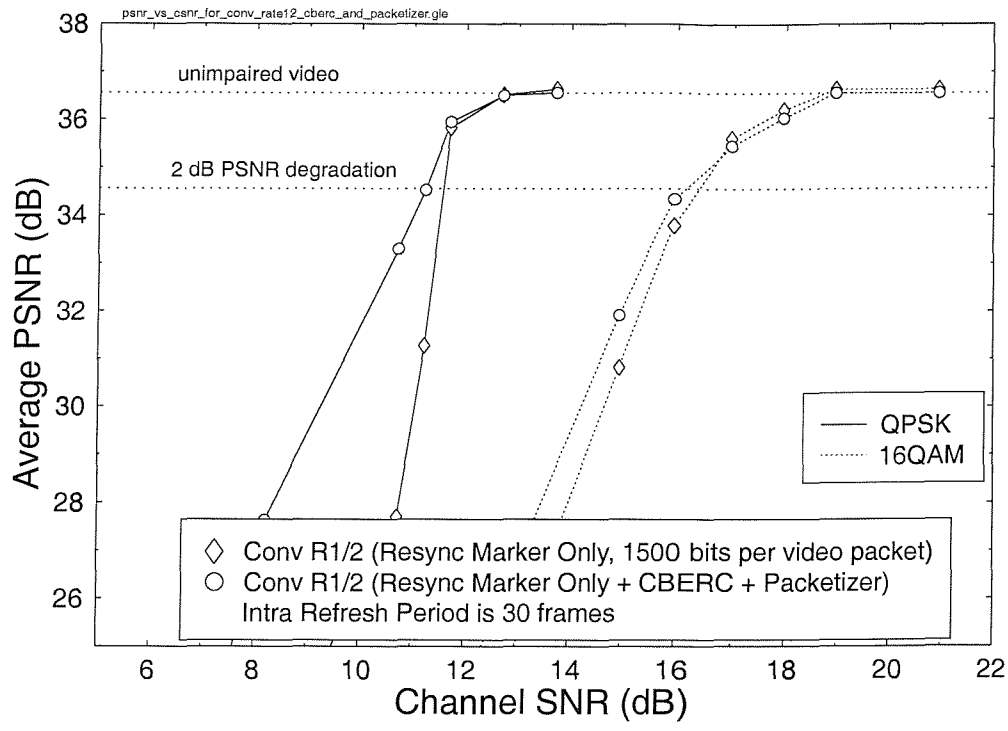


Figure 5.45: Average video PSNR versus channel SNR performance for the MPEG-4 encoded "Salesman" video sequence transmitted using the DVB terrestrial system employing rate 1/2 convolutional code as its inner code. The modulation types are QPSK and 16-QAM. The intra-frame refresh period is 30 frames and CBERC denotes content-based error resilient coding.

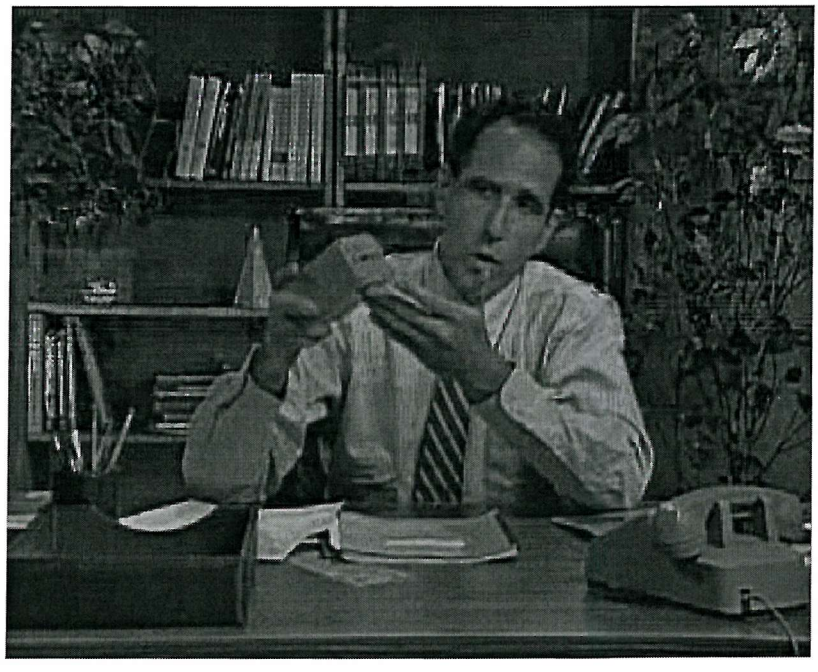
portrayed. It should be noted that the unimpaired video and the 2 dB PSNR degradation threshold shown in Figures 5.50 and 5.51 refer to the "Stefan" video sequence, employing the low-complexity resynchronisation marker assisted scheme. The unimpaired PSNR video quality of the packetisation plus adaptive intra-coded macroblock update scheme is approximately 1 dB lower in this case. This is a consequence of sending more intra-coded macroblocks. Intra-coded macroblocks are encoded using more bits than inter-coded macroblocks. Hence, when intra-coded macroblocks are transmitted, there will be less bits available for encoding other macroblocks, when a constant bitrate encoding scenario is used.

Overall, we conclude that the introduction of the proposed packetisation enhancement at the system layer combined with the adaptive intra-coded macroblock update scheme can improve the error resilience of the MPEG-4 video decoder. However, the bitrate efficiency of the proposed scheme is limited by the associated higher proportion of intra-coded macroblocks.

5.7 Summary

The MPEG-4 standard was contrived for enabling multimedia communication. Its requirements and functionalities include the ability to interact with the encoded media, achieving a high compression ratio and the support of coding tools for enabling universal

(a) Resynchronisation marker only



(b) Packetisation and adaptive intra-coded macroblock update

Figure 5.46: Comparison of the reconstructed Frame 31 of the "Salesman" video sequence when utilising (a) the resynchronisation marker assisted scheme and (b) the proposed packetisation scheme as well as adaptive intra-coded macroblock update for increasing the video decoder's resilience to channel errors.

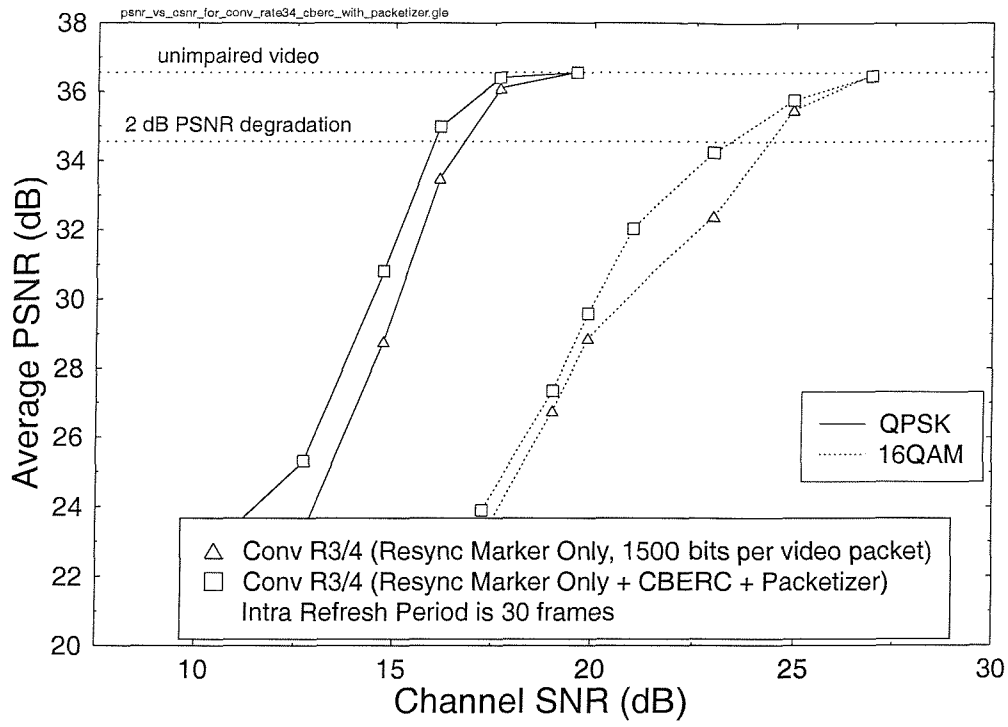


Figure 5.47: Average video PSNR versus channel SNR performance for the MPEG-4 encoded "Salesman" video sequence transmitted using the DVB terrestrial system employing rate 3/4 convolutional code as its inner code. The modulation types are QPSK and 16-QAM. The intra-frame refresh period is 30 frames and CBERC denotes content-based error resilient coding.

access of the encoded media. The MPEG-4 system's architecture was defined such that these requirements are met. The architecture comprises:

- **Object Descriptor Framework:** This module is responsible for providing the necessary tools for an MPEG-4 terminal for describing all the objects that exist within an MPEG-4 presentation.
- **Binary Format for Scenes:** This module complements the object descriptor framework in providing means for the multimedia content creator to supply composition information for the MPEG-4 terminal.
- **Elementary Stream Management:** This module manages the streaming of encoded object data to the MPEG-4 terminal.
- **System Decoder Model:** It defines a system buffer model and a system timing model in order to model the behaviour of a receiving MPEG-4 terminal. This is important in overcoming the problem of buffer underflow and overflow, as well as in providing a synchronised presentation of MPEG-4 encoded objects.
- **Delivery Multimedia Integration Framework:** This framework provides the necessary abstraction for any MPEG-4 application from the underlying communication networks. This concept is essential for MPEG-4 systems to interoperate with each other through heterogeneous networks.

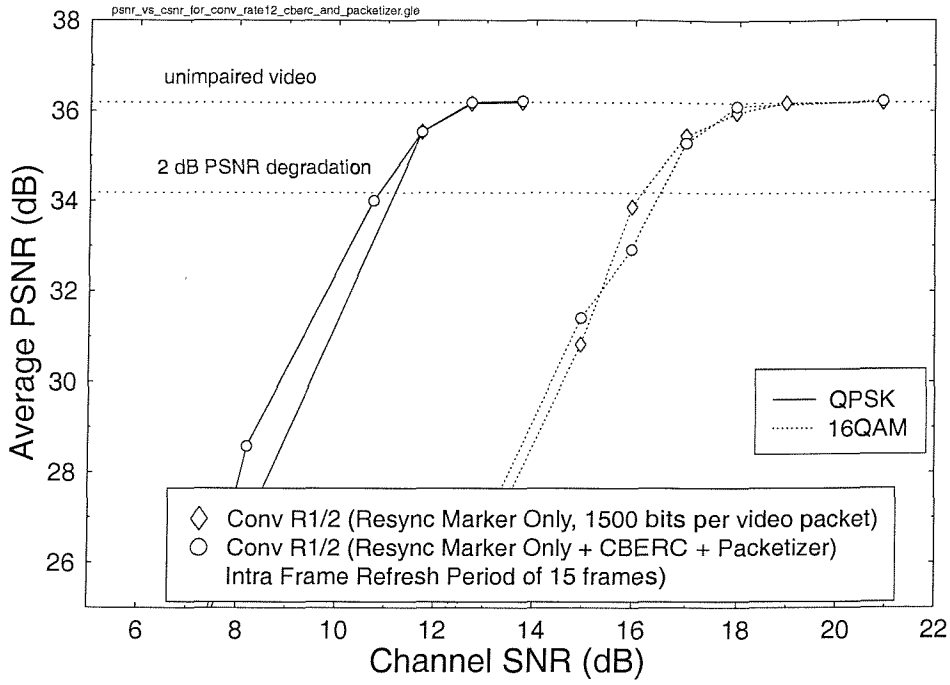


Figure 5.48: Average video PSNR versus channel SNR performance for the MPEG-4 encoded "Salesman" video sequence transmitted using the DVB terrestrial system employing rate 1/2 convolutional code as its inner code. The modulation modes are QPSK and 16-QAM. The intra-frame refresh period is 15 frames. CBERC denotes content-based error resilient coding.

The system architecture section of the standard provides us with an overview of the operation of a typical MPEG-4 terminal. We elaborated on the tools available for coding natural video in Section 5.4. In order to support object-based coding, shape coding tools are made available in the MPEG-4 Visual specification [5]. There are two types of shape coding tools, namely binary shape coding and grey scale shape coding, which were the topic of Sections 5.4.1.1 and 5.4.1.2, respectively. The pixel padding algorithm was introduced into the motion estimation and DCT modules of the video codec in order to facilitate the encoding of arbitrary shaped video objects. Apart from the enhancements of the MPEG-4 video codec incorporated for supporting arbitrary shaped object coding, the philosophy of the codec is similar in principle to that of other standard block-based motion compensated DCT video codecs, such as H.261 [1], H.263 [2], MPEG-1 [3] and MPEG-2 [4]. The scalability tools of the codec include spatial scalability and temporal scalability, which allow the video sequence to be encoded such that MPEG-4 terminals having different decoding capabilities can decode the encoded video sequence, albeit with different qualities. This is one of the requirements for universal access of MPEG-4 encoded media, which we discussed in Section 5.2.2. Furthermore in Section 5.5, we observed that the MPEG-4 has a wide operating range, from very low bitrate to very high bitrate scenarios. This ensures that the codec can be used in a multitude of applications, from mobile video conferencing to high definition television broadcast.

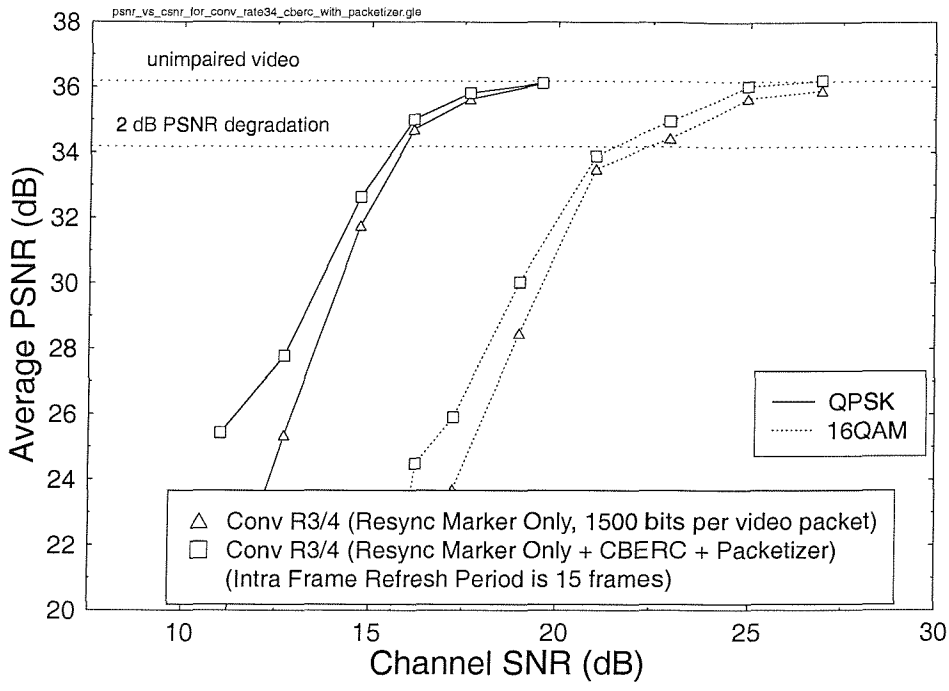


Figure 5.49: Average video PSNR versus channel SNR performance for the MPEG-4 encoded "Salesman" video sequence transmitted using the DVB terrestrial system employing rate 3/4 convolutional code as its inner code. The modulation modes are QPSK and 16-QAM. The intra-frame refresh period is 15 frames. CBERC denotes content-based error resilient coding.

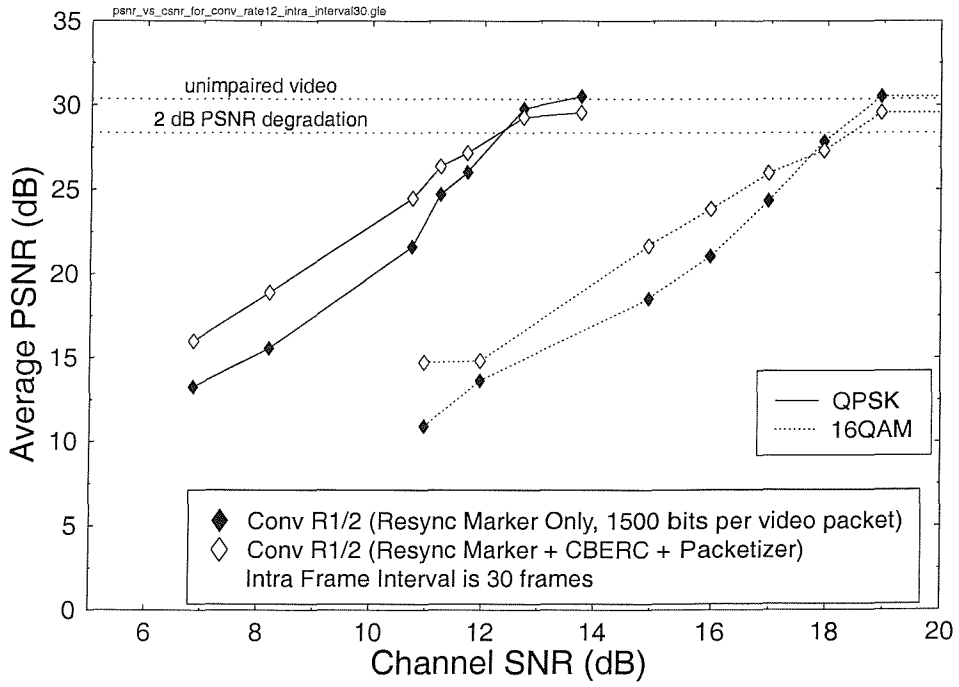


Figure 5.50: Average video PSNR versus channel SNR performance for the MPEG-4 encoded "Stefan" video sequence transmitted using the DVB terrestrial system employing rate 1/2 convolutional code as its inner code. The modulation modes are QPSK and 16-QAM. The intra-frame refresh period is 30 frames. CBERC denotes content-based error resilient coding.

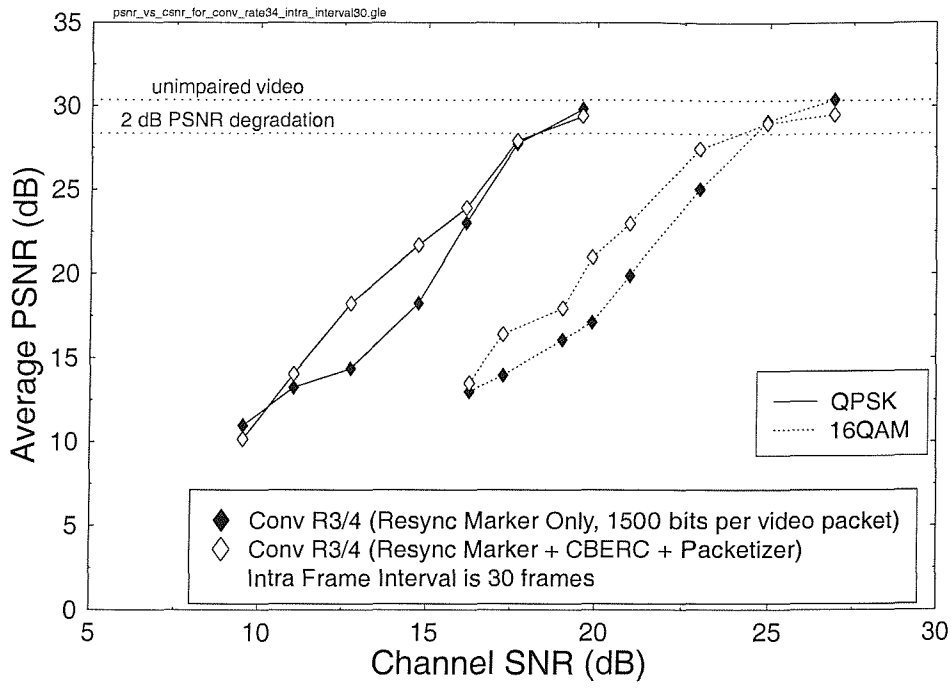


Figure 5.51: Average video PSNR versus channel SNR performance for the MPEG-4 encoded "Stefan" video sequence transmitted using the DVB terrestrial system employing rate 3/4 convolutional code as its inner code. The modulation modes are QPSK and 16-QAM. The intra-frame refresh period is 30 frames. CBERC denotes content-based error resilient coding.

Another requirement for universal access is the need to transmit MPEG-4 encoded media over heterogeneous networks, such as the Internet, wireless communication networks and broadcast networks. The propagation conditions over wireless communication networks are particularly harsh. Hence, the MPEG-4 committee has proposed error resilience tools to be incorporated into the MPEG-4 visual specification. These tools are resynchronisation markers, data partitioning and reversible variable-length codes. We have provided experimental results for characterising the resynchronisation marker assisted scheme and the data partitioning scheme in Section 5.6.4 in the context of DVB terrestrial transmission scenarios. We have also invoked a packetisation plus adaptive intra-coded macroblock update scheme, which can further improve the error resilience of the MPEG-4 video codec. The proposed scheme was shown to perform well up to a transport packet loss ratio of approximately 5 %.

Chapter 6

Conclusions and Suggestions for Future Work

6.1 Summary

In this thesis, we have discussed a range of issues related to digital video transmissions to mobile and fixed receivers. In Chapter 1, we have highlighted the basics of video compression in the context of standard video codecs, such as H.263, MPEG-2 and MPEG-4. The associated error resilience issues related to these video codecs were also considered. Specifically, following our introduction to the problems associated with video transmission, we embarked on quantifying the sensitivity of the MPEG-2 parameters to channel errors in Chapter 2. We have also presented our results on the performance of the terrestrial DVB scheme employing both non-hierarchical and hierarchical transmission scenarios. Furthermore, we have characterised the performance of the satellite DVB scheme, which was presented in Chapter 3. In order to further enhance the performance of the DVB systems, we have also proposed the replacement of the DVB convolutional inner code with a convolutional constituent code based turbo code. In Chapter 4, we studied coded modulation schemes. Finally, we have also studied the recently ratified MPEG-4 standard in Chapter 5.

In order to elaborate further, our discussions on video compression fundamentals in Chapter 1 has shown that variable-length codes and predictive coding techniques are employed by the family of state-of-the-art video codecs to achieve high compression ratios. However, such techniques are sensitive against transmission errors. Specifically, the corruption of variable-length codes will in general lead to the loss of data until the recovery of the next resynchronisation codeword within the video bitstream. The employment of predictive techniques also means that the transmission errors will be propagated, if the synchronisation markers are corrupted. 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-2 bitstream parameters to channel errors. Our experiments have shown that the variable-length codes, which are used to encode the DCT coefficients are the most susceptible to channel errors. This has suggested the use of unequal error protection schemes for protecting the MPEG-2 bitstream. The video

bitstream was split into two classes, namely a high-priority class and a low-priority class. The MPEG-2 data partitioning scheme was invoked in this case. The DVB terrestrial system operating in its hierarchical mode was used to multiplex both the high-priority and low-priority video data prior to modulation for transmission. The DVB system specifies the mapping of the high-priority and low-priority data to each modulation symbol. This in turn imposes a restriction on the proportion of video data that can be inserted into the high-priority partition. Our simulations showed that in order to satisfy the above restriction imposed by the transmitter as well as for preventing the video buffer from overflowing or underflowing, a substantial amount of video data would be relegated to the low-priority partition. Hence, in the event of errors in the low-priority partition, a high proportion of video data was lost and low reconstructed video quality was recorded. Hence, we found that the non-partitioned reconstructed video quality was better than that of the partitioned scenario. We then endeavoured to enhance the tranceiver's performance further by proposing the replacement of the DVB inner convolutional codec with a more powerful turbo codec. Our simulations demonstrated that the turbo codec was capable of providing a coding gain of approximately 5 dB at a similar codec complexity, as the inner convolutional code employed, when the same modulation mode is employed. Furthermore, if the rate 1/2 turbo code is employed together with 16-QAM modulation, the SNR required in order to achieve a similar bit error rate performance was similar to that of the scheme employing a rate 1/2 convolutional code and QPSK. This implies that the useful bit per symbol transmission capacity can be doubled and a higher video quality can be attained.

Having characterised the performance of the terrestrial DVB system, we continued our study with the satellite-based DVB system. Furthermore, turbo-coded enhancement, similar to that of the terrestrial DVB system, was also introduced. Again, we observed that the rate 1/2 turbo code was capable of providing a coding gain of approximately 4–5 dB. We have also shown that 16-QAM can be invoked at a similar power budget to that of QPSK, when turbo coding is employed. This enhancement translates into doubling of the useful bit per symbol transmission capacity. The doubling of transmission capability can benefit the video codec in two different ways. Namely, the video can be encoded at a higher quality or the number of video channels can be doubled, given a certain bandwidth.

Continuing our efforts towards enhancing the tranceiver performance further, we embarked on the study of coded modulation schemes in various broadcast scenarios in Chapter 4. The coded modulation schemes studied were trellis coded modulation, turbo trellis coded modulation, bit-interleaved coded modulation and iterative decoded bit-interleaved coded modulation. The trellis coded modulation schemes were found capable of matching the performance of the inner convolutional code employed by the DVB standards. The advantage of the trellis coded modulation schemes was that they could achieve this performance at a lower decoding complexity. Furthermore, for the case where a rate 2/3 TCM 8PSK scheme having a code memory of 7 is employed, the associated performance is similar to that of the rate 1/2 convolutional code using a code memory of 6 and a QPSK modulation scheme, as employed in the DVB standard. This became explicit in

the performance curves of Figure 4.4. Hence the corresponding effective throughputs were 2 and 1 bit per symbol, respectively. Therefore, the video can be encoded at a higher quality or the number of video channels can be increased, similarly to the improvements attained by employing turbo codes instead of convolutional codes. In our simulations, it was found that by doubling the coded bitrate for the "Salesman" sequence, the reconstructed PSNR video quality increased by approximately 2 dB. While the TCM codes can perform similarly to the corresponding convolutional codes of the same code rate but at a lower decoding complexity, the BICM codes can achieve an additional coding gain of 3.5 dB for the QPSK and 8PSK scenarios, with respect to the convolutional codes of the same code rate as well as the same decoding complexity as the corresponding TCM codes. The performance of the BICM codes can be further improved, if iterative decoding is employed. Of the four coded modulation schemes, the TTCM schemes provided the best performance in terms of power efficiency. When compared to the turbo-coded DVB-like system, the TTCM schemes have a better spectral efficiency in high-order constellation modes, such as 16-QAM. By contrast, in low-order constellation modes, both schemes possess a similar spectral efficiency.

Chapter 5 provided an overview of the recently standardised MPEG-4 standard. It started by discussing the functionalities and requirements of the MPEG-4 standard. This was followed by the description of the MPEG-4 system architecture that underlies the functionalities supported. Next, we elaborated on the features of the MPEG-4 standard. A notable contribution of the standard is the introduction of object-based video coding techniques. We have also shown that the MPEG-4 video codec is versatile and meets the requirements of a wide range of applications. It can encode video at rates as low as 5–6 kbps for mobile video conferencing applications up to rates as high as several megabits per second at a video quality suitable for digital television broadcast. A feature of particular interest in the standard is the introduction of error resilience tools for transmitting video through hostile channels, such as mobile radio channels. There are two main types of error resilience tools, namely resynchronisation markers and data partitioning. The resynchronisation markers allow the video decoder to resynchronise the decoding process, once an error has occured. The data between the error and the next resynchronisation marker will be lost. In order to more strongly protect the more sensitive video bits, the concept of data partitioning was introduced. In this case, the motion and texture data are separated. The philosophy behind this scheme is that should the texture data be corrupted, the video decoder still has the motion data in order to reconstruct a better image. However, the data partitioning scheme relies on the optimistic assumption that the motion data are received without error. If this condition is not met, the video decoder will not necessarily be able to reconstruct an adequate-quality image, even though the associated texture data is received without errors. This observation is valid because the correct motion data may still point to a corrupted region in the previous reconstructed image and hence error propagation may occur. In our simulations, we found that the resynchronisation marker based scheme typically performs better, than the data partitioning scheme.

We also note that the MPEG-4 video decoder is capable of detecting errors in the

bitstream with the aid of semantic and syntactic checks. However, it fails to detect errors, which may cause the corruption of one valid codeword to another valid codeword. One particular example, which we have shown was the corruption of the quantiser index parameter. The corresponding reconstructed image for this case was shown in Figure 5.46(a). As a remedy of this problem, we have proposed a simple packetisation scheme, where the video decoder will only be supplied with data, which has been received correctly. This scheme will enable us to circumvent the problem mentioned above. Nevertheless, one problem still persists in the current scheme. The portion of the image, where the coded data is dropped, is concealed by using the corresponding portion of the previous reconstructed image. If the concealment does not provide a good match to the actual missing image portion, the viewer will notice it. We have adopted an adaptive intra-coded macroblock update scheme originally proposed by Chen and Hwang [110] as a counter-measure for mitigating the poor concealment problem. It should be noted, however, that other intra-coded macroblock update schemes [109, 113, 115] can also be employed. In our simulations we observed that the above simple packetisation and adaptive intra-coded macroblock update scheme can operate effectively up to an approximately 5 % transmission packet loss ratio.

6.2 Suggestions for Further Work

There are several possible extensions to the work presented in this thesis. In Chapter 2, we characterised the performance of the terrestrial DVB systems, proposing the replacement of the inner convolutional code with a turbo code. It would be beneficial to study the performance of the turbo code, when various types of turbo interleavers are used. The characterisation of the terrestrial DVB system's performance should also be extended to other channel models and Doppler frequencies.

The implementationally complex DVB Reed-Solomon decoder cannot utilise the soft decision inputs provided by the inner decoder. Hence, when we replaced the convolutional codec with the turbo codec, we did not exploit the soft decision outputs provided by the turbo decoder. The feasibility of adapting the Reed-Solomon decoder to accepting soft information bits would certainly be worthwhile future research.

The MPEG-4 audio-visual system has recently been standardised. The system promised to provide the enabling technology for multimedia communication. In this respect, we are hoping to extend our work to encompass the coding of stereoscopic video. The simplest form of stereoscopic video is presented in the form of having a pair of images which are captured at the same sampling instance, where one of the images is presented to the right eye and the other one to the left eye. With the existing MPEG-4 video standard, we are capable of encoding stereoscopic video by invoking the video encoder's temporal scalability mode. The images to be presented to the right eye, for instance, can be encoded as the base layer. The corresponding images for the left eye can be encoded as the enhancement layer. This particular coding scenario would enable us to encode our stereoscopic video without violating the syntax of the MPEG-4 standard. Apart from the block-based motion compensated DCT codec feature of the MPEG-4, we are also interested in exploring the feasibility of employing wavelet-based schemes for

compressing the stereoscopic video. A particular advantage of the wavelet-based scheme is its ability to encode video at various levels of resolutions. This feature is attractive in enabling the encoded video to be viewed by different viewers who may be wearing stereoscopic glasses of varying capabilities. Typical applications in this area are virtual conferencing, three-dimensional television broadcast, immersive computer games within a multiuser environment, distance learning and telemedicine. Furthermore, it would also be interesting to extend our error resilience work to cover stereoscopic video coding.

We have not considered the effects of the network layer on the transmission of video source signals in this thesis. In a real-life scenario, the video source will be multiplexed with other video sources, audio sources, textual data and control information. It is important that the correct source data is routed to the correct receipient. During network congestion or transmission errors, some transmission packets may be dropped. The receiver has to be sufficiently robust in order to be able to still forward the correct source data to the right owner. Furthermore, in communication networks such as the Internet, network congestion will lead to delay jitter. The source decoder has to be adapted in order to cope with the resultant delay jitter. Within wireless networks users are free to roam from one serving cell to another. At the network layer, the routing algorithm has to be able to track the movement of the user and forward the owner's source data correctly and in time, in order to support real-time communication.

Glossary

AAL5	ATM Adaptation Layer 5
ACK	Positive acknowledgement signal in automatic repeat request system (ARQ)
ARQ	Automatic Repeat Request
ATM	Asynchronous Transfer Mode transport protocol
B-G	Benveniste-Goursat Algorithm
B-ISDN	Broadband Integrated Services Digital Network
BAB	Binary Alpha Block
BICM	Bit-Interleaved Coded Modulation
BICM-ID	Iterative Decoded Bit-Interleaved Coded Modulation
BIFS	Binary Format for Scenes
CAE	Context Based Arithmetic Encoding
CBERC	Content Based Error Resilient Coding
CIF	Common Intermediate Format images containing 352 pixels in the horizontal direction and 288 pixels in the vertical direction
CMA	Constant Modulus Algorithm
DAB	Digital Audio Broadcast
DAI	DMIF Application Interface
DCT	Discrete Cosine Transformation
DMIF	Delivery Multimedia Integration Framework
DVB	Digital Video Broadcasting
EREC	Error Resilient Entropy Coding
FEC	Forward Error Correction
GOB	Group of Blocks
HDTV	High Definition Television
HEC	Header Extension Code
IDCT	Inverse Discrete Cosine Transformation
IP	Internet Protocol
IPMP	Intellectual Property Rights Management and Protection
ISD	Independent Segment Decoding
LMS	Least Mean Squares
LogMAP	Logarithmic Maximum A-Posteriori
LOS	line-of-sight
MBM	Motion Boundary Marker

MCER	Motion compensated error residual
MCMA	Modified Constant Modulus Algorithm
NACK	Negative acknowledgement signal in automatic repeat request system (ARQ)
OBMC	Overlapped Block Motion Compensation
OCR	Object Clock Reference
OFDM	Orthogonal Frequency Division Multiplexing
OTB	Object Time Base
PBP	Priority Breakpoint which is used by the data partitioning scheme to indicate the position of an input video bitstream to be split into two output bitstreams
PCESC	Picture Coding Extension Start Code
POTS	Plain Old Telephone Service
PSC	Picture Start Code
PSNR	Peak Signal-to-Noise Ratio
PSP	Per-Survivor Processing Algorithm
QAM	Quadrature Amplitude Modulation
QCIF	Quarter Common Intermediate Format images containing 176 pixels in the horizontal direction and 144 pixels in the vertical direction
QPSK	Quadrature Phase Shift Keying
RCPCC	Rate Compatible Punctured Convolutional Code
RPS	Reference Picture Selection
RS	Reed-Solomon codes
RSC	Recursive Systematic Convolutional Code
RTP	Real-Time Transport Protocol
RVLC	Reversible Variable Length Codes
S-a-G	Stop-and-Go Algorithm
SDM	MPEG-4 System Decoder Model
SL	Sync Layer
SNR	Signal-to-Noise Ratio
TCM	Trellis Coded Modulation
TDD	Time Division Duplex
TS	MPEG-2 Transport System
TTCM	Turbo Trellis Coded Modulation
UDP	User Datagram Protocol
UMTS	Universal Mobile Telecommunication System
VO	Video Object
VOL	Video Object Layer
VOP	Video Object Plane
VRML	Virtual Reality Modelling Language
VS	Video Session

WWW World Wide Web

Bibliography

[1] *ITU-T Recommendation H.261: Video Codec For Audiovisual Services at $p \times 64$ kbits/s*, March 1993.

[2] *ITU-T Recommendation H.263: Video Coding For Low Bitrate Communication*, December 1995. Draft.

[3] *ISO/IEC 11172-2: Coding of moving pictures and associated audio for digital storage media up to about 1.5 Mbit/s*, 1993.

[4] *ISO/IEC 13818-2: Information Technology - Generic Coding of Moving Pictures and Associated Audio Information - Part 2: Video*, March 1995.

[5] *ISO/IEC 14496-2: Information Technology - Coding of Audio-Visual Objects: Visual*, November 1998. Final Draft International Standard, ISO/IEC JTC1/SC29/WG11 N2502.

[6] R. V. Cox, B. G. Haskell, Y. Lecun, B. Shahraray, and L. Rabiner, "On the applications of multimedia processing to communications," *Proceedings of the IEEE*, vol. 86, pp. 755–824, May 1998.

[7] D. Wood, "The DVB project: philosophy and core system," *Electronics & Communication Engineering Journal*, vol. 9, pp. 5–10, February 1997.

[8] G. Drury, "DVB channel coding standards for broadcasting compressed video services," *Electronics & Communication Engineering Journal*, vol. 9, pp. 11–20, February 1997.

[9] D. J. Cutts, "DVB conditional access," *Electronics & Communication Engineering Journal*, vol. 9, pp. 21–27, February 1997.

[10] U. Reimers, "DVB-T: the COFDM-based system for terrestrial television," *Electronics & Communication Engineering Journal*, vol. 9, pp. 28–32, February 1997.

[11] A. Arcidiacono, "Multimedia services and broadcasting via satellite," *Electronics & Communication Engineering Journal*, vol. 9, pp. 33–37, February 1997.

[12] ETSI, *Digital Video Broadcasting (DVB); Framing structure, channel coding and modulation for 11/12 GHz Satellite Services*, August 1997. EN 300 421 V1.1.2.

[13] ETSI, *Digital Video Broadcasting (DVB); Framing structure, channel coding and modulation for digital terrestrial television*, August 1997. EN 300 744 V1.1.2.

[14] M. Civanlar and G. Cash, "A practical system for MPEG-2-based video-on-demand over ATM packet networks and the WWW," *Signal Processing: Image Communication*, vol. 8, pp. 221–227, April 1996.

[15] G. Kerr, "A review of fully interactive video on demand," *Signal Processing: Image Communication*, vol. 8, pp. 173–190, April 1996.

[16] ETSI, *Digital Video Broadcasting (DVB); Framing structure, channel coding and modulation for cable systems*, December 1997. EN 300 429 V1.2.1.

[17] H. Zheng and K. J. R. Liu, "Multimedia services over digital subscriber lines," *IEEE Signal Processing Magazine*, vol. 17, pp. 44–60, July 2000.

[18] M. Jeffrey, "Asynchronous Transfer Mode: the ultimate broadband solution?," *Electronics and Communication Engineering Journal*, vol. 6, pp. 143–151, June 1994.

[19] P. Cuenca, A. Garrido, F. Quiles, and L. Orozco-Barbosa, "An efficient protocol architecture for error resilient MPEG-2 video communications over ATM networks," *IEEE Transactions on Broadcasting*, vol. 45, pp. 129–140, March 1999.

[20] R. Prasad and T. Ojanperä, "An overview of CDMA evolution toward wideband CDMA." IEEE Communications Survey, Fourth Quarter 1998. Available at http://www.comsoc.org/pubs/surveys.

[21] M. Gallagher and W. Webb, "UMTS: the next generation of mobile radio," *IEE Review*, vol. 45, pp. 59–63, March 1999.

[22] M. Shafi, A. Hashimoto, M. Umehira, S. Ogose, and T. Murase, "Wireless communications in the twenty-first century: A perspective," *Proceedings of the IEEE*, vol. 85, pp. 1622–1638, October 1997.

[23] R. Steele and L. Hanzo, eds., *Mobile Radio Communications: Second and Third Generation Cellular and WATM Systems.* John Wiley & Sons, 2 ed., 1999.

[24] T. Ojanperä and R. Prasad, *Wideband CDMA for Third Generation Mobile Communications.* Artech House Publishers, 1998.

[25] R. Prasad, W. Mohr, and W. Konhauser, eds., *Third Generation Mobile Communications Systems.* Artech House Publishers, 2000.

[26] H. Holma and A. Toskala, *WCDMA for UMTS.* John Wiley & Sons, 2000.

[27] R. Koenen, F. Pereira, and L. Chiariglione, "MPEG-4: Context and Objectives," *Signal Processing: Image Communication*, vol. 9, pp. 295–303, May 1997.

[28] P. K. Doenges, T. K. Capin, F. Lavagetto, J. Ostermann, I. S. Pandzic, and E. D. Petajan, "MPEG-4: Audio/video and synthetic graphics/audio for mixed media," *Signal Processing: Image Communication*, vol. 9, pp. 433–463, May 1997.

[29] T. Chen, A. Katsaggelos, and S. Y. Kung, "The Past, Present and Future of Multimedia Signal Processing," *IEEE Signal Processing Magazine*, vol. 14, pp. 28–51, July 1997.

[30] F. Pereira and R. Koenen, "Very low bit-rate audio-visual applications," *Signal Processing: Image Communication*, vol. 9, pp. 55–77, November 1996.

[31] K. Aizawa and T. S. Huang, "Model Based Image Coding: Advanced Video Coding Techniques for Very Low Bit-Rate Applications," in *Proceedings of the IEEE*, vol. 83, pp. 259–271, February 1995.

[32] T. Ebrahimi, "MPEG-4 Video Verification Model: A video encoding/decoding algorithm based on content representation," *Signal Processing: Image Communication*, vol. 9, pp. 367–384, May 1997.

[33] P. Kauff, B. Makai, S. Rauthenberg, G. Gölz, J. L. P. de Lameillieure, and T. Sikora, "Functional coding of video using a shape-adaptive DCT algorithm and an object-based motion prediction toolbox," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 7, pp. 181–196, February 1997.

[34] H. Li, A. Lundmark, and R. Forcheimer, "Image Sequence Coding at Very Low Bitrates: A Review," *IEEE Transactions on Image Processing*, vol. 3, pp. 589–609, September 1994.

[35] R. Talluri, K. Oehler, T. Bannon, J. D. Courtney, A. Das, and J. Liao, "A robust, scalable, object-based video compression technique for very low bit-rate coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 7, pp. 221–233, February 1997.

[36] P. Salembier, F. Marqués, M. Pardàs, J. R. Morros, I. Corset, S. Jeannin, L. Bouchard, F. Meyer, and B. Marcotegui, "Segmentation-Based Video Coding System Allowing the Manipulation of Objects," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 7, pp. 60–74, February 1997.

[37] P. Salembier and M. Pardàs, "Hierarchical Morphological Segmentation for Image Sequence Coding," *IEEE Transactions on Image Processing*, vol. 3, pp. 639–651, September 1994.

[38] H. Katata, N. Ito, and H. Kusao, "Temporal-Scalable Coding Based on Image Content," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 7, pp. 52–59, February 1997.

[39] B. Girod, K. B. Younes, R. Bernstein, P. Eisert, N. Färber, F. Hartung, U. Horn, E. Steinbach, K. Stuhlmüller, and T. Weigand, "Recent advances in video compression," in *Proceedings of IEEE ISCAS'96*, vol. 2, (Atlanta), pp. 580–583, 1996.

[40] S. Malassiotis and M. G. Strintzis, "Object-based coding of stereo image sequences using three-dimensional models," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 7, pp. 892–905, December 1997.

[41] L. Torres and M. Kunt, eds., *Video Coding: The Second Generation Approach.* Norwell, Massachusetts: Kluwer Academic Publishers, 1996.

[42] L. Hanzo, P. J. Cherriman, and J. Streit, *Wireless Video Communications: Second to Third Generation Systems and Beyond.* IEEE Press, January 2001.

[43] R. Sekuler and R. Blake, *Perception.* McGraw-Hill Inc., Singapore, 3 ed., 1994.

[44] T. N. Cornsweet, *Visual Perception.* Academic Press, London, 1970.

[45] A. Jain, *Fundamentals of Digital Image Processing.* Prentice-Hall Inc., 1989.

[46] R. Clarke, *Transform Coding of Images.* Academic Press, 1985.

[47] P. Tudor, "MPEG-2 video compression," *Electronic and Communication Engineering Journal*, vol. 7, pp. 257–264, December 1995.

[48] B. Rogowitz, "The human visual system: A guide for the display technologist," in *Proc. Society of Information Display (SID)*, vol. 24, pp. 235–252, 1983.

[49] J. R. Jain and A. K.Jain, "Displacement measurement and its application in interframe image coding," *IEEE Transactions on Communications*, vol. COM-29, pp. 1799–1808, December 1981.

[50] H. Gharavi and M. Mills, "Block matching motion estimation algorithms: New results," *IEEE Transactions on Circuits and Systems*, vol. 37, pp. 649–651, May 1990.

[51] Q. Wang and R.J.Clarke, "Motion estimation and compensation for image sequence coding," *Signal Processing: Image Communication*, vol. 4, pp. 161–174, April 1992.

[52] K. Chun and J. Ra, "An improved block matching algorithm based on successive refinement of motion vector candidates," *Signal Processing: Image Communication*, vol. 6, pp. 115–122, May 1994.

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

[54] B. Zeng, R. Li, and M. L. Liao, "Optimization of fast block motion estimation algorithms," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 7, pp. 833–844, December 1997.

[55] A. Puri, R. Aravind, and B. Haskell, "Adaptive frame/field motion compensated video coding," *Signal Processing: Image Communication*, vol. 5, pp. 39–58, February 1993.

[56] T. Huang and A. Netravali, "Motion and structure from feature correspondences: A review," *Proceedings of the IEEE*, vol. 82, pp. 252–267, February 1994.

[57] F. Dufaux and F. Moscheni, "Motion estimation techniques for digital TV: A review and a new contribution," *Proceedings of the IEEE*, vol. 83, pp. 858–875, June 1995.

[58] J. Aggarwal and N. Nandhakumar, "On the computation of motion from sequences of images: A review," *Proceedings of the IEEE*, vol. 76, pp. 917–935, August 1988.

[59] M. I. Sezan and R. L. Lagendijk, eds., *Motion Analysis and Image Sequence Processing*. Norwell, Massachusetts: Kluwer Academic Publishers, 1993.

[60] M. T. Orchard and G. J. Sullivan, "Overlapped block motion compensation: An estimation-theoretic approach," *IEEE Transactions on Image Processing*, vol. 3, pp. 693–699, September 1994.

[61] C. Stiller and J. Konrad, "Estimating motion in image sequences," *IEEE Signal Processing Magazine*, vol. 16, pp. 70–91, July 1999.

[62] K. Rao and P. Yip, *Discrete Cosine Transform*. Academic Press, New York, 1990.

[63] T. Masaki, Y. Morimoto, T. Onoye, and I. Shirakawa, "VLSI implementation of inverse discrete cosine transformer and motion compensator for MPEG-2 HDTV video decoding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 5, pp. 387–395, October 1995.

[64] N. Ahmed, T. Natrajan, and K. R. Rao, "Discrete cosine transform," *IEEE Transactions on Computer*, vol. C-23, pp. 90–93, December 1984.

[65] A. Netravali and B. Haskell, *Digital Pictures: Representation, Compression and Standards*. Plenum Press, New York, 2 ed., 1995.

[66] M. Nelson and J. Gailly, *The Data Compression Book*. M & T Books, 1996.

[67] B. G. Haskell, A. Puri, and A. N. Netravali, *Digital Video: An Introduction To MPEG-2*. Digital Multimedia Standards Series, Chapman & Hall, 1997.

[68] A. Puri and R. Aravind, "Motion-compensated video coding with adaptive perceptual quantization," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 1, pp. 351–361, December 1991.

[69] J. L. Mitchell, W. B. Pennebaker, C. E. Fogg, and D. Le Gall, eds., *MPEG Video Compression Standard.* Digital Multimedia Standards Series, Chapman & Hall, 1997.

[70] A. Ortega and K. Ramchandran, "Rate-distortion methods for image and video compression," *IEEE Signal Processing Magazine*, vol. 15, pp. 23–50, November 1998.

[71] G. J. Sullivan and T. Wiegand, "Rate-distortion optimization for video compression," *IEEE Signal Processing Magazine*, vol. 15, pp. 74–90, November 1998.

[72] S. Park, Y. Lee, and H. Chang, "A new MPEG-2 rate control scheme using scene change detection," *Electronics and Telecommunications Research Institute (ETRI) Journal*, vol. 18, pp. 61–74, July 1996.

[73] W. Ding and B. Liu, "Rate control of MPEG video coding and recording by rate-quantization modeling," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, pp. 12–20, February 1996.

[74] T.-Y. Chung, K.-H. Jung, Y.-N. Oh, and D.-H. Shin, "Quantization control for improvement of image quality compatible with MPEG-2," *IEEE Transactions on Consumer Electronics*, vol. 40, pp. 821–825, November 1994.

[75] C.-F. Chang and J.-S. Wang, "A stable buffer control strategy for MPEG coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 7, pp. 920–924, December 1997.

[76] K. Chun, K. Lim, H. Cho, and J. Ra, "An adaptive perceptual quantization algorithm for video coding," *IEEE Transactions on Consumer Electronics*, vol. 39, pp. 555–558, August 1993.

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

[78] H. Sun, W. Kwok, M. Chien, and C. H. J. Ju, "MPEG coding performance improvement by jointly optimizing coding mode decisions and rate control," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 7, pp. 449–458, June 1997.

[79] G. Schusters and A. Katsaggelos, *Rate-distortion based video compression.* Kluwer Academic Publishers, March 1997.

[80] T. Chiang and Y. Q. Zhang, "A new rate control scheme using quadratic rate distortion model," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 7, pp. 246–250, February 1997.

[81] H. M. Hang and J. J. Chen, "Source model for transform video coder and its application – Part I: Fundamental theory," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 7, pp. 287–298, April 1997.

[82] J. J. Chen and H. M. Hang, "Source model for transform video coder and its application – Part II: Variable frame rate coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 7, pp. 299–311, April 1997.

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

[84] R. Koenen, "MPEG-4: Multimedia for our time," *IEEE Spectrum*, vol. 36, pp. 26–33, February 1999.

[85] R. Talluri, "Error-resilient video coding in the ISO MPEG-4 standard," *IEEE Communications Magazine*, pp. 112–119, June 1998.

[86] O. Avaro, P. A. Chou, A. Eleftheriadis, C. Herpel, C. Reader, and J. Signès, "The MPEG-4 Systems and Description Languages: A Way Ahead in Audio Visual Information Representation," *Signal Processing: Image Communication*, vol. 9, pp. 385–431, May 1997.

[87] T. Sikora, "The MPEG-4 Video Standard Verification Model," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 7, pp. 19–31, February 1997.

[88] L. Ducla-Soares and F. Pereira, "Error resilience and concealment performance for MPEG-4 frame-based video coding," *Signal Processing: Image Communication*, vol. 14, pp. 447–472, May 1999.

[89] F. Seytter, "An efficient multiplex architecture for mobile MPEG-4 systems," *Signal Processing: Image Communication*, vol. 14, pp. 599–606, May 1999.

[90] A. Puri, R. Kollarits, and B. Haskell, "Basics of stereoscopic video, new compression results with MPEG-2 and a proposal for MPEG-4," *Signal Processing: Image Communication*, vol. 10, pp. 201–234, July 1997.

[91] I. Moccagatta, S. Soudagar, J. Liang, and H. Chen, "Error-resilient coding in JPEG-2000 and MPEG-4," *IEEE Journal on Selected Areas in Communications*, vol. 18, pp. 899–914, June 2000.

[92] Y. Lee, F. Kossentini, R. Ward, and M. Smith, "Toward MPEG-4: An improved H.263-based video coder," *Signal Processing: Image Communication*, vol. 10, pp. 143–158, 1997.

[93] K. Ngan, S. Panchanathan, T. Sikora, and M.-T. Sun, "Special issue on segmentation, description and retrieval of video content," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 8, September 1998.

[94] K. Ngan, S. Panchanathan, T. Sikora, and M.-T. Sun, "Special issue on representation and coding of images and video II," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, February 1999.

[95] H. H. Chen, T. Ebrahimi, G. Rajan, C. Horne, P. K. Doenges, and L. Chiariglione, "Special issue on synthetic/natural hybrid video coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, March 1999.

[96] F. Pereira, S.-F. Chang, R. Koenen, A. Puri, and O. Avaro, "Special issue on object-based video coding and description," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, December 1999.

[97] T. Ebrahimi and C. Horne, "MPEG-4 natural video coding – An overview," *Signal Processing: Image Communication*, vol. 15, pp. 365–385, 2000.

[98] O. Avaro, A. Eleftheriadis, C. Herpel, G. Rajan, and L. Ward, "MPEG-4 systems: Overview," *Signal Processing: Image Communication*, vol. 15, pp. 281–298, 2000.

[99] C. Herpel and A. Eleftheriadis, "MPEG-4 systems: Elementary stream management," *Signal Processing: Image Communication*, vol. 15, pp. 299–320, 2000.

[100] L. Hanzo, W. Webb, and T. Keller, *Single- and Multi-carrier Quadrature Amplitude Modulation: Principles and Applications for Personal Communications, WLANs and Broadcasting.* John Wiley & Sons, 2000.

[101] J. Griffiths, *Radio Wave Propagation and Antennas - An Introduction.* Prentice Hall, 1987.

[102] ISO/IEC JTC1/SC29/WG11, *Report of the formal verification tests on MPEG-4 video error resilience*, December 1998. N2604.

[103] R. Talluri, I. Moccagatta, Y. Nag, and G. Cheung, "Error concealment by data partitioning," *Signal Processing: Image Communication*, vol. 14, pp. 505–518, May 1999.

[104] Takashima, Wada, and Murakami, "Reversible variable length codes," *IEEE Transactions on Communication*, vol. 43, pp. 158–162, February–April 1995.

[105] G. Wen and J. Villasenor, "A class of reversible variable length codes for robust image and video coding," in *Proceedings of the International Conference on Image Processing*, vol. 2, (Santa Barbara, California), pp. 65–68, October 1997.

[106] M. Budagavi and J. D. Gibson, "Random lag selection in multiframe motion compensation," in *IEEE International Symposium on Information Theory*, August 1998. Submitted for publication.

[107] Y. Wang, S. Wenger, J. Wen, and A. K. Katsaggelos, "Error resilient video coding techniques: Real time video communications over unreliable networks," *IEEE Signal Processing Magazine*, vol. 17, pp. 61–82, July 2000.

[108] S. Wenger, G. Côté, M. Gallant, and F. Kossentini, *H.263 Test Model number 11 revision 1*, August 1999. Q15-G-16R1. Available at ftp://standard.pictel.com/videosite/.

[109] S. Wenger, G. 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.

[110] W. H. J. Chen and J. N. Hwang, "The CBERC: A content-based error-resilient coding technique for packet video communications," *IEEE Transactions on Circuits and Systems for Video Technology*, 1999. Submitted for possible publication.

[111] 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.

[112] G. Côté, S. Shirani, and F. Kossentini, "Optimal mode selection and synchronization for robust video communications over error-prone networks," *IEEE Journal on Selected Areas in Communications*, vol. 18, pp. 952–965, June 2000.

[113] D. Wu, Y. T. Hou, B. Li, W. Zhu, Y. Q. Zhang, and H. J. Chao, "An end-to-end approach for optimal mode selection in internet video communication: Theory and application," *IEEE Journal on Selected Areas in Communications*, vol. 18, pp. 977–995, June 2000.

[114] G. de los Reyes, A. R. Reibman, S. F. Chang, and J. C. I. Chuang, "Error-resilient transcoding for video over wireless channels," *IEEE Journal on Selected Areas in Communications*, vol. 18, pp. 1063–1073, June 2000.

[115] J. Y. Liao and J. Villasenor, "Adaptive intra block update for robust transmission of H.263," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 10, pp. 30–35, February 2000.

[116] I. Rhee and S. R. Joshi, "Error recovery for interactive video transmission over the internet," *IEEE Journal on Selected Areas in Communications*, vol. 18, pp. 1033–1049, June 2000.

[117] R. Swann and N. Kingsbury, "Techniques for improving the error resilience of MPEG-2 codecs," in *Proceedings of the International Conference on Image Processing*, pp. 450–455, September 1996.

[118] R. Swann and N. Kingsbury, "Transcoding of MPEG-2 for enhanced resilience to transmission errors," in *Proceedings of the International Conference on Image Processing*, pp. 813–816, September 1996.

[119] R. Swann, *MPEG-II Video Coding For Noisy Channels.* PhD thesis, University of Cambridge, March 1998.

[120] D. W. Redmill and N. G. Kingsbury, "The EREC: an error resilient technique for coding variable length blocks of data," *IEEE Transactions on Image Processing*, vol. 5, pp. 565–574, April 1994.

[121] P. Czerepinski, M. F. Tariq, D. Bull, N. Canagarajah, and A. Nix, "Robust matching pursuits video transmission using the Hiperlan/1 air interface standard," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 6, (Istanbul, Turkey), pp. 1939–1942, June 2000.

[122] M. F. Tariq, P. Czerepinski, A. Nix, D. Bull, and N. Canagarajah, "Robust and scalable matching pursuits video transmission using the Bluetooth air interface standard," *IEEE Transactions on Consumer Electronics*, vol. 46, pp. 673–681, August 2000.

[123] T. Chiang and D. Anastassiou, "Hierarchical coding of digital television," *IEEE Communications Magazine*, pp. 38–45, May 1996.

[124] A. Aravind, M. R. Civanlar, and A. R. Reibman, "Packet loss resilience of MPEG-2 scalable video coding algorithms," *IEEE Transaction on Circuits and Systems for Video Technology*, vol. 6, pp. 426–435, October 1996.

[125] H. Gharavi, "Multilevel video coding and distribution architectures for emerging broadband digital networks," *IEEE Transaction on Circuits and Systems for Video Technology*, vol. 6, pp. 459–469, October 1996.

[126] H. Gharavi, "Efficient multilevel coding scheme for video transmission," *Electronics Letters*, vol. 28, pp. 949–951, May 1992.

[127] H. Gharavi and W. Ng, "H.263 compatible video coding and transmission," in *First International Workshop on Wireless Image/Video Communications*, (Loughborough University, Loughborough, U.K.), pp. 115–120, September 1996.

[128] U. Benzler, "Scalable multiresolution video coding using subband decomposition," in *First International Workshop on Wireless Image/Video Communications*, (Loughborough University, Loughborough, U.K.), pp. 109–114, September 1996.

[129] U. Horn, B. Girod, and B. Belzer, "Scalable video coding for multimedia applications and robust transmission over wireless channels," in *7th International Workshop on Packet Video*, (Brisbane, Australia), pp. 43–48, March 1996.

[130] E. Steinbach, N. Färber, and B. Girod, "Standard compatible extension of H.263 for robust video transmission in mobile environments," *IEEE Transaction on Circuits and Systems for Video Technology*, vol. 7, pp. 872–881, December 1997.

[131] P. Cherriman, "Mobile multimedia networking," Master's thesis, Department of Electronic and Computer Science, University of Southampton, UK, November 1996.

[132] P. Cherriman and L. Hanzo, "ARQ-assisted H261 and H263-based programmable video transceivers," in *Proceedings Of ICCS'96/ISPAC'96*, (Singapore), pp. 1369–1373, November 1996.

[133] M. Khansari, A. Jalali, E. Dubois, and P. Mermelstein, "Low bit rate video transmission over fading channels for wireless microcellular systems," *IEEE Transaction on Circuits and Systems for Video Technology*, vol. 6, pp. 1–11, February 1996.

[134] S. Aign, "Error concealment, early resynchronization and iterative decoding for MPEG-2," in *IEEE International Conference on Communication (ICC)*, vol. 3, (Montreal, Canada), pp. 1654–1658, 1997.

[135] S. Aign, "Error concealment improvements for MPEG-2 using enhanced error detection and early re-synchronization," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 4, (Munich, Germany), pp. 2625–2628, April 1997.

[136] R. E. V. Dyck, "MPEG-4 image transmission using MAP source-controlled channel decoding," *IEEE Journal on Selected Areas in Communications*, vol. 18, pp. 1087–1098, June 2000.

[137] J. Hagenauer, "Source-controlled channel decoding," *IEEE Transactions on Communications*, vol. 43, pp. 2449–2457, September 1995.

[138] Z. Peng, Y. Huang, and D. J. Costello, "Turbo codes for image tranmission – A joint channel and source decoding approach," *IEEE Journal on Selected Areas in Communications*, vol. 18, pp. 868–879, June 2000.

[139] W. Xu, J. Hagenauer, and J. Hollmann, "Joint source-channel decoding using the residual redundancy in compressed images," in *IEEE International Conference on Communication (ICC)*, vol. 1, pp. 142–148, June 1996.

[140] W. Xu, J. Hagenauer, and J. Hollmann, "Joint cource and channel decoding of compressed still images," *Signal Processing: Image Communication*, vol. 15, pp. 217–229, 1999.

[141] J. Wen and J. D. Villasenor, "Utilizing soft information in decoding of variable length codes," in *Data Compression Conference*, pp. 131–139, 1999.

[142] A. Murad and T. Fuja, "Variable length source codes, channel codes, and list decoding," in *IEEE Information Theory and Communications Workshop*, (Kruger National Park, South Africa), p. 114, June 1999.

[143] A. Murad and T. Fuja, "Robust transmission of variable-length encoded sources," in *IEEE Wireless Communications and Networking Conference*, vol. 2, pp. 968–972, 1999.

[144] A. Murad and T. Fuja, "Joint source-channel decoding of variable-length encoded sources," in *IEEE Information Theory and Communications Workshop*, (Killarney, Ireland), pp. 94–95, June 1998.

[145] L. Pérez and K. Sayood, "Joint source/channel coding for variable length codes using a precoder," in *IEEE Wireless Communications and Networking Conference*, vol. 2, pp. 983–987, 1999.

[146] N. Demir and K. Sayood, "Joint source/channel coding for variable length codes," in *Data Compression Conference (DCC)*, pp. 139–148, 1998.

[147] V. Balakirsky, "Joint source-channel coding with variable length codes," in *IEEE International Symposium on Information Theory (ISIT)*, (Ulm, Germany), pp. 1014–1021, May 1996.

[148] M. Park and D. Miller, "Joint source-channel decoding for variable-length encoded data by exact and approximate MAP sequence estimation," *IEEE Transactions on Communications*, vol. 48, pp. 1–6, January 2000.

[149] I.-W. Tsai and C.-L. Huang, "Hybrid cell loss concealment methods for MPEG-2-based packet video," *Signal Processing: Image Communication*, vol. 9, pp. 99–124, January 1997.

[150] W. Kwok and H. Sun, "Multi-directional interpolation for spatial error concealment," *IEEE Transactions on Consumer Electronics*, vol. 39, pp. 455–460, August 1993.

[151] 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.

[152] M.-C. Hong, H. Schwab, L. P. Kondi, and A. K. Katsaggelos, "Error concealment algorithms for compressed video," *Signal Processing: Image Communication*, vol. 14, pp. 473–492, May 1999.

[153] S. Kaiser and K. Fazel, "Comparison of error concealment techniques for an MPEG-2 video decoder in terrestrial TV-broadcasting," *Signal Processing: Image Communication*, vol. 14, pp. 655–676, May 1999.

[154] Y. Wang and Q.-F. Zhu, "Error control and concealment for video communication: A review," *Proceedings of the IEEE*, vol. 86, pp. 974–997, May 1998.

[155] V. DeBrunner, L. DeBrunner, L. Wang, and S. Radhakrishnan, "Error control and concealment for image transmission," *IEEE Communications Surveys and Tutorials*, vol. 3, pp. 2–9, First Quarter 2000. Available at http://www.comsoc.org/pubs/surveys.

[156] C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: turbo codes," *IEEE Transactions on Communications*, vol. 44, pp. 1261–1271, October 1996.

[157] C. S. Lee, T. Keller, and L. Hanzo, "OFDM-based turbo-coded hierarchical and non-hierarchical terrestrial mobile digital video broadcasting," *IEEE Transaction on Broadcasting*, vol. 46, pp. 1–22, March 2000.

[158] C. S. Lee, S. Vlahoyiannatos, and L. Hanzo, "Satellite based turbo-coded, blind-equalized 4-qam and 16-qam digital video broadcasting," *IEEE Transaction on Broadcasting*, vol. 46, pp. 23–33, March 2000.

[159] C. S. Lee, S. X. Ng, L. Piazzo, and L. Hanzo, "TCM, TTCM, BICM and iterative BICM assisted OFDM-based digital video broadcasting to mobile receivers," in *Proceedings of Vehicular Technology Conference (VTC)*, (Tel Aviv, Israel), Spring 2001. Submitted for possible publication.

[160] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression.* Kluwer Academic Publishers, 1992.

[161] R. M. Gray, "Vector quantization," *IEEE Acoustic, Speech and Signal Processing*, vol. 1, pp. 4–29, April 1984.

[162] W. Li and Y. Q. Zhang, "Vector-based signal processing and quantization for image and video compression," *Proceedings of the IEEE*, vol. 83, pp. 317–335, February 1995.

[163] P. J. Burt and E. H. Adelson, "The Laplacian pyramid as a compact image code," *IEEE Transactions on Communications*, vol. 31, pp. 532–540, April 1983.

[164] J. M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Transactions on Signal Processing*, vol. 41, pp. 3445–3462, December 1993.

[165] S. A. Martucci, I. Sodagar, T. Chiang, and Y. Q. Zhang, "A zerotree wavelet video coder," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 7, pp. 109–118, February 1997.

[166] Z. Xiong, K. Ramchandran, M. T. Orchard, and Y. Q. Zhang, "A comparative study of DCT- and wavelet-based image coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, pp. 692–695, August 1999.

[167] M. Hong, F. Kossentini, and M. J. T. Smith, "A family of efficient and channel error resilient wavelet/subband image coders," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, pp. 95–108, February 1999.

[168] P. Strobach, "Tree-structured scene adaptive coder," *IEEE Transactions on Communications*, vol. 38, pp. 477–486, April 1990.

[169] G. J. Sullivan and R. L. Baker, "Efficient quadtree coding of images and video," *IEEE Transactions on Image Processing*, vol. 3, pp. 327–331, May 1994.

[170] A. M. Michelson and A. H. Levesque, *Error Control Techniques for Digital Communication.* Wiley-Interscience, 1985.

[171] S. O'Leary and D. Priestly, "Mobile broadcasting of DVB-T signals," *IEEE Transactions on Broadcasting*, vol. 44, pp. 346–352, September 1998.

[172] W.-C. Lee, H.-M. Park, K.-J. Kang, and K.-B. Kim, "Performance analysis of viterbi decoder using channel state information in COFDM system," *IEEE Transactions on Broadcasting*, vol. 44, pp. 488–496, December 1998.

[173] S. O'Leary, "Hierarchical transmission and COFDM systems," *IEEE Transactions on Broadcasting*, vol. 43, pp. 166–174, June 1997.

[174] L. Thibault and M. T. Le, "Performance evaluation of COFDM for digital audio broadcasting Part I: parametric study," *IEEE Transactions on Broadcasting*, vol. 43, pp. 64–75, March 1997.

[175] L. Hanzo and J. Woodard, "An intelligent cordless voice terminal for indoors communications," *IEEE Transactions on Vehicular Technology*, vol. 44, pp. 735–749, November 1995.

[176] L. Hanzo and J. Streit, "Adaptive low rate wireless videophone schemes," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 5, pp. 305–318, August 1995.

[177] P. Shelswell, "The COFDM modulation system: the heart of digital audio broadcasting," *Electronics & Communication Engineering Journal*, vol. 7, pp. 127–136, June 1995.

[178] 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*, pp. 1064–1070, May 1993.

[179] S. B. Wicker, *Error Control Systems for Digital Communication and Storage.* Prentice Hall, 1994.

[180] A. Barbulescu and S. Pietrobou, "Interleaver design for turbo codes," *IEE Electronics Letters*, pp. 2107–2108, December 1994.

[181] P. Robertson, E. Villebrun, and P. Hoeher, "A comparison of optimal and suboptimal MAP decoding algorithms operating in the log domain," in *IEEE Proceedings of the International Conference on Communications*, pp. 1009–1013, June 1995.

[182] M. Failli, "Digital land mobile radio communications COST 207," tech. rep., European Commission, 1989.

[183] H. Gharavi and S. Alamouti, "Multipriority video transmission for third-generation wireless communication system," in Gharavi and Hanzo [185], pp. 1751–1763. pp. 1703–1846.

[184] 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.

[185] H. Gharavi and L. Hanzo, eds., *Proceedings of the IEEE*, vol. 87, October 1999. pp. 1703–1846.

[186] G. Reali, G. Baruffa, S. Cacopardi, and F. Frescura, "Enhancing satellite broadcasting services using multiresolution modulations," *IEEE Transactions on Broadcasting*, vol. 44, pp. 497–506, December 1998.

[187] Y. Hsu, Y. Chen, C. Huang, and M. Sun, "MPEG-2 spatial scalable coding and transport stream error concealment for satellite TV broadcasting using Ka-band," *IEEE Transactions on Broadcasting*, vol. 44, pp. 77–86, March 1998.

[188] L. Atzori, F. D. Natale, M. D. Gregario, and D. Giusto, "Multimedia information broadcasting using digital TV channels," *IEEE Transactions on Broadcasting*, vol. 43, pp. 383–392, December 1997.

[189] W. Sohn, O. Kwon, and J. Chae, "Digital DBS system design and implementation for TV and data broadcasting using Koreasat," *IEEE Transactions on Broadcasting*, vol. 44, pp. 316–323, September 1998.

[190] M. Karaliopoulos and F.-N. Pavlidou, "Modelling the land mobile satellite channel: a review," *Electronics and Communication Engineering Journal*, vol. 11, pp. 235–248, October 1999.

[191] J. Goldhirsh and W. J. Vogel, "Mobile satellite system fade statistics for shadowing and multipath from roadside trees at UHF and L-band," *IEEE Transactions on Antennas and Propagation*, vol. 37, pp. 489–498, April 1989.

[192] W. Vogel and J. Goldhirsh, "Multipath fading at L band for low elevation angle, land mobile satellite scenarios," *IEEE Journal on Selected Areas in Communications*, vol. 13, pp. 197–204, February 1995.

[193] W. Vogel and G. Torrence, "Propagation measurements for satellite radio reception inside buildings," *IEEE Transactions on Antennas and Propagation*, vol. 41, pp. 954–961, July 1993.

[194] W. Vogel and U. Hong, "Measurement and modelling of land mobile satellite propagation at UHF and L-band," *IEEE Transactions on Antennas and Propagation*, vol. 36, pp. 707–719, May 1988.

[195] S. Saunders, C. Tzaras, and B. Evans, "Physical statistical propagation model for mobile satellite channel," tech. rep., European Commission, 1998.

[196] S. Saunders, *Antennas and Propagation for Wireless Communication Systems Concept and Design.* John Wiley & Sons, 1999.

[197] K. Wesolowsky, "Analysis and properties of the modified constant modulus algorithm for blind equalization," *European Transactions on Telecommunication*, vol. 3, pp. 225–230, May–June 1992.

[198] M. Goursat and A. Benveniste, "Blind equalizers," *IEEE Transactions on Communications*, vol. COM–28, pp. 871–883, August 1984.

[199] G. Picchi and G. Prati, "Blind equalization and carrier recovery using a "stop–and–go" decision–directed algorithm," *IEEE Transactions on Communications*, vol. COM–35, pp. 877–887, September 1987.

[200] A. Polydoros, R. Raheli, and C. Tzou, "Per–survivor processing: a general approach to MLSE in uncertain environments," *IEEE Transactions on Communications*, vol. COM–43, pp. 354–364, February–April 1995.

[201] D. N. Godard, "Self–recovering equalization and carrier tracking in two–dimensional data communication systems," *IEEE Transactions on Communications*, vol. COM–28, pp. 1867–1875, November 1980.

[202] Y. Sato, "A method of self–recovering equalization for multilevel amplitude–modulation systems," *IEEE Transactions on Communications*, vol. COM–23, pp. 679–682, June 1975.

[203] Z. Ding, R. A. Kennedy, B. D. O. Anderson, and R. C. Johnson, "Ill-convergence of Godard blind equalizers in data communications systems," *IEEE Transactions on Communications*, vol. COM-39, pp. 1313–1327, September 1991.

[204] G. Ungerboeck and I. Csajka, "On improving data link performance by increasing the channel alphabet and introducing sequence coding," in *International Symposium on Information Theory*, (Ronneby, Sweden), June 1976.

[205] G. Ungerboeck, "Channel coding with multilevel/phase signals," *IEEE Transactions on Information Theory*, vol. IT-28, pp. 55–67, January 1982.

[206] D. Divsalar and M. K. Simon, "The design of trellis coded MPSK for fading channel: Performance criteria," *IEEE Transactions on Communications*, vol. 36, pp. 1004–1012, September 1988.

[207] D. Divsalar and M. K. Simon, "The design of trellis coded MPSK for fading channel: Set partitioning for optimum code design," *IEEE Transactions on Communications*, vol. 36, pp. 1013–1021, September 1988.

[208] P. Robertson and 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.

[209] E. Zehavi, "8-PSK trellis codes for a Rayleigh channel," *IEEE Transactions on Communications*, vol. 40, pp. 873–884, May 1992.

[210] X. Li and J. A. Ritcey, "Bit-interleaved coded modulation with iterative decoding," *IEEE Communications Letters*, vol. 1, pp. 169–171, November 1997.

[211] X. Li and J. A. Ritcey, "Bit-interleaved coded modulation with iterative decoding using soft feedback," *IEE Electronics Letters*, vol. 34, pp. 942–943, May 1998.

[212] X. Li and J. A. Ritcey, "Trellis coded modulation with bit interleaving and iterative decoding," *IEEE Journal on Selected Areas in Communications*, vol. 17, pp. 715–724, April 1999.

[213] S. Lin and D. J. Costello, *Error Control Coding: Fundamentals and Applications.* Prentice Hall, 1983.

[214] 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.

[215] L. Chiariglione, "MPEG and Multimedia Communications," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 7, pp. 5–18, February 1997.

[216] *ISO/IEC 14496-3: Information Technology - Coding of Audio-Visual Objects: Audio*, November 1998. Final Draft International Standard, ISO/IEC JTC1/SC29/WG11 N2503.

[217] *ISO/IEC 14496-1: Information Technology - Coding of Audio-Visual Objects: Systems*, November 1998. Final Draft International Standard, ISO/IEC JTC1/SC29/WG11 N2501.

[218] S. Tsekeridou and I. Pitas, "MPEG-2 error concealment based on block-matching principles," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 10, pp. 646–658, June 2000.

[219] W.-M. Lam and A. R. Reibman, "An error concealment algorithm for images subject to channel errors," *IEEE Transactions on Image Processing*, vol. 4, p. 1995, May 1995.

[220] *ISO/IEC 14496-6: Information Technology - Coding of Audio-Visual Objects: Delivery Multimedia Integration Framework*, November 1998. Final Draft International Standard, ISO/IEC JTC1/SC29/WG11 N2506.

[221] ECMA-262, *ECMAScript Language Specification*, December 1999. 3rd Edition.

[222] D. Flanagan, *Javascript.* O'Reilly & Associates, 3 ed., 1998.

[223] *ISO/IEC 14772-1: The Virtual Reality Modelling Language*, 1997. Available at http://www.vrml.org/.

[224] J. Signés, "Binary format for scenes (BIFS): Combining MPEG-4 media to build rich multimedia services." Available at http://www.cselt.it/mpeg/documents/koenen/-signes.zip.

[225] J. Signés, Y. Fisher, and A. Eleftheriadis, "MPEG-4's binary format for scene description." Available at http://www.cselt.it/leonardo/icjfiles/mpeg-4_si/5-BIFS-paper/5-BIFS-paper.htm.

[226] A. M. Tekalp and J. Ostermann, "Face and 2-D mesh animation in MPEG-4," *Signal Processing: Image Communication*, vol. 15, pp. 387–421, 2000.

[227] R. Koenen, "Profiles and levels in MPEG-4: Approach and overview," *Signal Processing: Image Communication*, vol. 15, pp. 463–478, 2000.

[228] *ISO/IEC 13818-1: Information Technology - Generic Coding of Moving Pictures and Associated Audio Information - Part 1: Systems*, November 1993.

[229] J. Ribas-Corbera and S. Lei, "Rate control in DCT video coding for low delay communications," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, pp. 172–185, February 1999.

[230] A. Vetro, H. Sun, and Y. Wang, "MPEG-4 rate control for multiple video objects," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, pp. 186–199, February 1999.

[231] J. L Ronda, M. Eckert, F. Jaureguizar, and N. García, "Rate control and bit allocation for MPEG-4," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, pp. 1243–1258, December 1999.

[232] J. D. Foley, A. V. Dam, S. K. Feiner, and J. F. Hughes, *Computer Graphics: Principles and Practice.* Addision Wesley Publishing Company, 2 ed., 1995.

[233] F. Bossen and T. Ebrahimi, "A simple and efficient binary shape coding technique based on bitmap representation," in *IEEE International Conference on Acoustics,*

Speech, and Signal Processing, vol. 4, (Munich, Germany), pp. 3129–3132, April 1997.