

DISSERTATION

Optimization of Video Streaming over 3G Networks

ausgeführt zum Zwecke der Erlangung des akademischen Grades
eines Doktors der technischen Wissenschaften

eingereicht an der Technischen Universität Wien
Fakultät für Elektrotechnik und Informationstechnik

von

DI Luca Superiori

Grohgasse 6/3C

A-1050 Wien

Österreich

geboren am 07. Juni 1980 in Camerino, Italien

Matrikelnummer: 0527973

Wien, 25 März 2010

Begutachter:

Univ. Prof. Dr. Markus Rupp

Institut für Nachrichtentechnik und Hochfrequenztechnik

Technische Universität Wien

Österreich

Univ. Prof. Dr. Daniele Giusto

Consorzio Nazionale Interuniversitario per le Telecomunicazioni

Università di Cagliari

Italien

Abstract

VIDEO streaming over cellular networks has been made possible in the last years by better performing video codecs and wireless cellular networks oriented to data transmission. The interaction between two heterogeneous worlds, the telecommunication infrastructure and the coding video software, calls for advanced optimization mechanisms. The actors involved in the optimization process are the cellular system's access network, UMTS and HSDPA, the wireless transmission channel and the final user equipped with a mobile device capable of decoding video sequences. The knowledge and characterization of each of the building blocks allow the optimization of each element to the specific needs of the others.

This doctoral thesis discusses three main contributions. In the first part, the effects of transmission errors on video streams are analyzed. Incorrectly received video packets are usually discarded by the lower layers and not conveyed to the application. Part of the payload, however, still contains valid encoded information. Only the data stored after the error occurrence cannot be correctly interpreted, because of the variable length coding. Several error detection strategies at application layer are analyzed and compared. Finally, a proposal for a more efficient sorting of the encoded information is discussed.

The second part of the thesis deals with the optimization of a specific service: soccer video streaming. Soccer represents one of the most attractive contents, as nomadic users are interested on receiving this content live in their mobile device. The way human users evaluate the quality of this specific video service is the key issue driving the study. By means of unsupervised segmentation, three regions of the frame have been identified: the field, the players with the ball, and the audience. As the three regions have a different impact in terms of subjective quality, their rate-distortion behavior has been optimized by assigning more data and, therefore, increasing the quality of the subjectively most important items.

In the third part, two cross-layer mechanisms are presented. As first, the application of SP and SI frames, specially encoded frames able to limit the temporal error propagation, has been discussed as an alternative to packet retransmission. The proposed optimization has been performed considering the measured error characteristics of the UMTS DCH. The second investigation is dedicated to the optimization of video streaming over HSDPA networks. Within a single video data stream, packets with different importance are discriminated. Exploiting secondary PDP contexts, several logical connection with different quality settings can be established. By means of header filtering, each packet is transmitted utilizing the appropriate channel, privileging the most important payloads.

Kurzfassung

VIDEO Streaming über zellulare Netze wurde in den letzten Jahren durch bessere Video Codecs und Funknetze ermöglicht, die Datenübertragung unterstützen. Die Interaktion zweier heterogener Welten, die Telekommunikationsinfrastruktur und der Videokodierungssoftware, ruft nach optimierenden Mechanismen. Die Akteure in diesem Optimierungsprozess sind das zellulare Zugriffsnetz (UMTS und HSDPA), der Funkkanal und der Kunde am anderen Ende, der mit einem Videoendgerät ausgestattet ist. Die genaue Kenntniss und Charakterisierung der beteiligten Funktionsblöcke erlaubt die Optimierung einzelner Elemente bezüglich der Anforderungen der Anderen.

Diese Dissertationsarbeit diskutiert drei wesentliche Beiträge. Im ersten Teil werden die Auswirkungen von Übertragungsfehlern analysiert. Inkorrekt empfangene Videodatenpakete werden üblicherweise durch die unteren Schichten entsorgt und nicht an die Applikationsschicht weitergeführt. Ein Teil davon enthält jedoch korrekt kodierte Information. Nur die Daten, die nach einem Fehler auftreten, werden durch die variable Längenkodierung falsch interpretiert. Verschiedene Detektionsstrategien auf der Applikationsschicht werden analysiert und verglichen. Schliesslich wird ein Vorschlag zur effizienten Sortierung der kodierten Information diskutiert.

Der zweite Teil der Arbeit behandelt die Optimierung eines speziellen Dienstes: Fussballvideo-Streaming. Fussball stellt den meistnachgefragtesten Inhalt dar, da bewegte Nutzer diesen Inhalt live empfangen wollen. Die Art wie menschliche Betrachter die Qualität dieses spezifischen Dienstes beurteilen, ist der Schlüssel der diese Studie antreibt. Durch unbeaufsichtigte Segmentierung werden drei unterschiedliche Regionen identifiziert: das Spielfeld, die Spieler mit dem Ball und die Zuschauer. Da die drei Regionen unterschiedlichen Einfluss auf die subjektive Qualität haben, wurde ihr rate-distortion Verhalten so optimiert, dass nur dort mehr Daten zugewiesen wurden, wo die Qualität den höchsten Anstieg hat, als den wichtigsten Objekten.

Im dritten Teil werden zwei Cross-Layer Mechanismen präsentiert. Zunächst wird die Verwendung von SP und SI Rahmen zur Kodierung vorgeschlagen anstelle von Paketwiedervermittlung, da sie die Fehlerfortpflanzung begrenzen. Die vorgeschlagene Optimierung wurde basierend auf gemessener Fehlercharakteristik des UMTS DCH ausgeführt. Die zweite vorgeschlagene Methode behandelt speziell die Optimierung von Video-Streaming in HSDPA Netzen. Innerhalb eines einzelnen Video-Streams werden Pakete unterschiedlicher Wichtigkeit entschieden. Mithilfe eines secondary PDP contexts werden verschiedene logische Verbindungen mit unterschiedlicher Übertragungsqualität realisiert. Durch "header" Filterung wird jedes Paket durch den zugehörigen Kanal übertragen wodurch die wichtigen Anteile höhere Qualität erfahren.

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Outline and Contributions	2
2	Video Streaming over 3G Cellular Networks	7
2.1	Video Streaming Paradigms	7
2.1.1	Broadcast	8
2.1.2	Multicast	8
2.1.3	Unicast	9
2.2	Streaming Technologies	9
2.2.1	3GPP PSS	9
2.2.2	Broadcast Technologies for Mobile Devices	12
2.3	Comparison and criticism	13
3	Detection of Errors in Corrupted Video Sequences	15
3.1	Effects of Errors at Frame and Sequence Level	15
3.2	Effects of Errors at Packet Level	17
3.3	Syntax Analysis	18
3.3.1	Error Handling Mechanism	21
3.3.2	Simulation Setup	23
3.3.3	Objective Quality Comparison	23
3.3.4	Detection Performance	27
3.4	Visual Artefacts Detection	29
3.4.1	Inter Frames Impairments	30
3.4.2	Inter Frames Impairments	31
3.4.3	Results	32
3.5	Watermarking	33
3.6	Smart Sorting	37
3.7	Conclusions and Self Criticism	40
4	Optimization of Soccer Video Streaming	45
4.1	Ball Visibility Improvement	45
4.1.1	MSE Based Mechanisms	47
4.1.2	Clustering Based Detection Mechanisms	49
4.1.2.1	Clustering	49
4.1.2.2	Feature Extraction and Analysis	51
4.1.2.3	Tracking Mechanism	52
4.1.2.4	Results	52
4.2	Encoding Optimization	54
4.2.1	Frame Segmentation	54

4.2.2	Application of FMO	57
4.2.3	Scene Analysis	62
4.2.4	Optimization of Grandstand's Encoding	66
4.2.5	Results	67
4.3	Conclusions and Self Criticism	70
5	Cross-Layer Optimization of Video Streaming	73
5.1	Cross Layer Application of SI/SP frames	74
5.1.1	Channel Model	77
5.1.2	Simulation Setup	79
5.1.3	Results	81
5.2	Optimization of Video Streaming over HSDPA Networks	84
5.2.1	Traffic Classes in IP and 3G Networks	84
5.2.2	Implementation of Secondary PDP Context in HSDPA Networks	87
5.2.3	HSDPA System Level Simulator	90
5.2.4	Results	91
5.3	Conclusions and Self Criticism	94
6	Conclusions	97
6.1	Results of this Thesis	97
6.2	Open Points and future Work	98
A	H.264/AVC	101
A-1	Network Abstraction Layer	102
A-2	Video Coding Layer	103
A-2.1	Intra Frame Prediction	104
A-2.2	Inter Frame Prediction	105
A-2.3	Transform and Quantization	105
A-3	Entropy Coding in H.264/AVC	106
A-4	Elements encoded into a NALU	108
A-5	Decoding	109
A-6	JM Reference Software	110
B	UMTS and HSPA Overview	111
B-1	GSM	111
B-2	GPRS	113
B-3	UMTS	114
B-4	HSPA	115
C	Abbreviations and Symbols	117
D	List of files used	123
E	Acknowledgements	125
	List of Figures	127
	Bibliography	131

Chapter 1

Introduction

1.1 Motivation

AT the time this doctoral thesis has been commenced, February 2006, the H.264/Advanced Video Coding (AVC) [1–3] was the three-years old state of the art video codec currently in the market. Today, March 2010, H.264/AVC is still the state of the art video codec although seven years from the date of its standardization have passed. H.264/AVC does not provide any major breakthrough, but fulfills to achieve efficient coding rate and flexibility by further enhancing the features of multiple previous standards.

The enhanced coding performance of the H.264/AVC and the increased data rates offered by the current cellular networks make the delivering of video streams to nomadic users possible. Universal Mobile Telecommunication System (UMTS), also addressed as the third generation of cellular wireless networks, has been launched in 2002 (in Austria) and allows a packet switched connection with data rates up to 384 kbit/s. Since 2006 UMTS networks have been upgraded with High Speed Packet Access (HSPA) capabilities, further increasing the data rate up to 14 Mbit/s in Single Input Single Output (SISO) mode and promises up to 42 Mbit/s in Multiple Input Multiple Output (MIMO) mode.

The transmission of encoded video streams over the wireless link of cellular network offers challenging open issues. Some of them are the focus of this doctoral thesis. The proposed optimizations have been performed identifying the optimal working point of the *rate-distortion* system behavior.

The data rate is defined as the average net amount of data the transmitter is allocating each second for the video service transmitted to a specific mobile user. As several customers are sharing a limited amount of resources, from a network operator's point of view the user data rate has to be kept as small as possible. Besides protocol overheads and signalling, the transmitted data considered in this thesis is an encoded video sequence. The amount of data associated to a video sequence depends on how much encoded information is necessary for building its reconstructed version at the decoder side. The video data rate can be adapted to the channel capacity either by adjusting the amount of frames sent each second or, as the H.264/AVC is a *lossy* video codec, by tuning the amount of refining information associated to each frame. The more refining information is sent, the better is the quality of the reconstructed sequence.

The quality of the reconstructed video has to match the expectations of the video content. From

the network operator’s point of view, a customer is served with a specified Quality of Service (QoS) [4], a value that includes the data rate, the packet error rate and maximum delay time. The user satisfaction, however, is not measured by these variables, but is rather evaluated by distortion metrics indicating the user’s Quality of Experience (QoE) [5]. The distortion is measured either with *objective* metrics such as the Luminance Peak Signal to Noise Ratio (Y-PSNR), or subjective such as the Mean Opinion Score (MOS). The former is an easily implementable mathematic technique, widely accepted although it requires the knowledge of the original sequence and it is not well correlated with specific features of the Human Visual System (HVS). The MOS is obtained by averaging the evaluations of human users. Subjective measurements are time consuming, expensive (in particular in case the test persons are paid) and the results cannot easily be reproduced, as they depend on the size and composition of test set. In this work, both metrics are used depending on the scope of the performed investigation.

The goal of this thesis is to exploit the knowledge of the video codec as well as the wireless network to build *ad hoc* solutions designed for this specific environment. The balance of the requirements of the users and of the network operator is performed by means of a deep investigation of the three involved actors: the access network, the wireless link and the end user. The access network has to manage the limited resources among concurrent users with different needs. The wireless channel is prone to fading and interference, possibly impairing the transmitted data. At the end of the chain, the human user is unaware of the previous two steps and is only focusing on the video service. Optimizing video services over cellular networks requires the three contributors to be handled as single individuals but as cross cooperating entities.

1.2 Outline and Contributions

This doctoral thesis covers three main topics as reflected by the system subdivision depicted in Figure 1.1, discussed in the previous section. In the following, the organization of this thesis as well as the author’s contributions are highlighted.

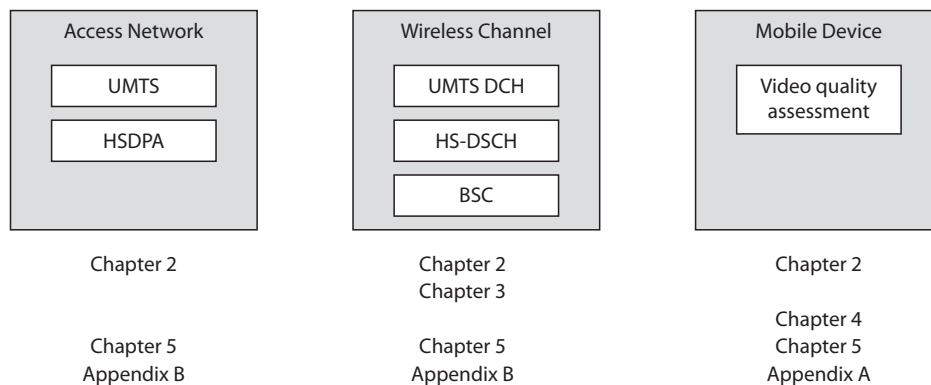


Figure 1.1: Organization of this thesis.

Chapter 2: Video Streaming over Third Generation (3G) Cellular Networks A brief overview of video streaming over wireless network is offered in this chapter. Three data delivery paradigms (broadcast, multicast and unicast) are presented. Afterward, the three current video streaming technologies for wireless network are introduced: Third Generation Partnership Project (3GPP) Packet Switched Services (PSS), 3GPP Multimedia Broadcast Multicast Service (MBMS) and Digital Video Broadcasting Handheld (DVB-H). This thesis mainly focuses on the PSS, although part of the obtained results can be also exploited by the other technologies.

Chapter 3: Detection of Errors in Corrupted Video Sequences This chapter presents an analysis of the effects of transmission errors as well as a collection of genuine mechanisms for reducing their impact on the reconstructed video.

Incorrectly received packets are usually not further conveyed to the application, as their correctness is evaluated at lower layer protocols, such as Universal Datagram Protocol (UDP). The missing information is concealed by the video decoder by means of temporal and spatial interpolation. As a single packet can contain an encoded region as big as the entire frame, discarding the whole packet [6] results in a significant degradation of the perceived quality. In order to limit the impact of damaged packets, the decoding of a corrupted packet is further analyzed. Even a single bit inversion makes the packet fail the checksum test, causing the whole payload to be marked as damaged. Our approach consists of delivering the corrupted payload to the application in order to exploit possible valid information.

A single bit inversion modifies the value of the decoded codewords. In H.264/AVC the length of the codewords depends on the structure of their content. The bit inversion, therefore, modifies the boundaries of the affected codewords, as well as those of the following codewords until the end of the packet. The information content stored before the error occurrence is not affected and still valid. In order to exploit this, the position of the error in the corrupted packet has to be identified. In [7, Superiori et al.]¹, [8, Superiori et al.],[9, Superiori et al.] and [10, Superiori et al.] a syntax analysis based error detection mechanism has been proposed. This approach is based on a work proposed by Barni [11] for H.263. As the corrupted codewords cause exceptions during the decoding, these have been caught and considered as the error position. The information before the error detection is decoded, the following data is concealed. As the position of the error occurrence and error detection does not coincide, the reconstructed video is examined in the pixel domain for detecting visual artefacts [12, Superiori et al.].

An application of invisible watermarking as an error detection mechanism has been discussed in [13, Superiori et al.]. By inserting a pattern known both at the encoder and at the decoder, possible transmission errors can be detected in case the watermarked information becomes corrupted. The insertion of the watermarking, however, causes the quality degradation of the video sequences even before transmission.

In [14, Superiori et al.] is proposed to differently arrange the content of the packet for better protecting the most important encoded information. The information stored at the beginning of the packet has a lower probability to be affected by error propagation. The most valuable information of each macroblock is, therefore, moved to the beginning of the packet. In case an error occurs, the

¹In this chapter the contributions of the author are highlighted by citing the name in references.

probability of losing the motion compensation information decreases.

Chapter 4: Optimization of Soccer Video Streaming This chapter discusses the optimization of a specific video server, streaming of soccer sequences, aiming at the enhancement of the subjective quality as experienced by human users.

Due to the high compression ratio needed, in low resolution video sequences the soccer ball can become invisible at the screen. As the visibility of the ball is necessary for the understanding of the game, the subjective quality is severely impaired. In [15, 16, Superiori et al.], the authors proposed a preprocessing mechanism to enhance the visibility of the ball at the user side. The detection of the ball in the original sequence represents the most challenging task. As an improvement of the original Mean Square Error (MSE) based method, in [17, Superiori et al.] a more robust clustering based mechanism has been introduced. It consists of the detection of the ball as an object with specific color and geometric features.

In literature, different works deal with the optimization of soccer video streaming from a subjective human perception point of view. In particular, in [18, 19] it is proposed to transmit and display only a cropped subregion of each frame centered around the identified ball position. This solution causes the loss of information and falls into copyright breaching. In order to further optimize soccer video streaming without removing part of the frame, the rate distortion behavior of three fundamental regions, the field, the players with the ball, and the audience has been analyzed. The investigation showed how most of the encoded data was employed for encoding the audience. For a human user, however, the audience represents the least important region for the understanding of the game. Exploiting Flexible Macroblock Ordering (FMO), a novel error resilience tool introduced in H.264/AVC, a method for optimizing the rate of each region has been proposed in [20, Superiori et al.]. A stronger compression ratio has been applied to the audience and more data has been reserved for the players and the ball.

Although the audience represents a static background, also the temporal encoding showed limited efficiency. Due to the reduced frame rate and decimated resolution, the high frequency components of consecutive pictures present significant difference. Although these cannot be appreciated by human users, they call for considerable additional amount of encoding coefficients. As the audience remains static, the difference between consecutive shots is solely originated by the camera movement. In order to further reduce the amount of bits associated to the grandstands, the possibility of applying a rigid translation of the audience region without additional coefficients in place of the standard motion compensation is discussed in [21, Superiori et al.]. The translation reflects the apparent audience movement caused by the camera movement. This has been estimated using a method [22, Superiori et al.] based on the statistics of the encoded frame.

Chapter 5: Cross Layer Optimization of Video Streaming The knowledge of the access network and wireless channel properties can be exploited for performing cross-layer optimizations.

The error characteristics of the transport blocks in the UMTS Dedicated CHannel (DCH) have been measured and modeled by Karner [23]. This information has been exploited for performing more accurate simulations, by mapping the error as Transport Block (TB) level to errors at Internet Protocol (IP) level. As the effect of a damaged frame persists in time because of the temporal error

propagation, the resynchronization is performed inserting spatially encoded pictures. As their encoding is much less efficient than their temporal encoding, the usage of spatially encoded frames has to be limited. H.264/*AVC* has defined two new frame types, the Switching P frame (*SP*) and Switching I frame (*SI*) frames [24]. The former recalls the temporal, the latter the spatial encoding mechanism. As they can be seamlessly substituted, the temporal error propagation can be stopped by sending an *SI* frame in place of an *SP* frame as soon as an error has been reported by the receiver side. The optimization of their application for the measured *UMTS DCH* is discussed in [25, Superiori et al.].

In 2006 an evolution of *UMTS*, the High Speed Download Packet Access (*HSDPA*), has been introduced. *HSDPA* allows even higher data rates and more scheduling flexibility. A video stream is transmitted dedicating a given amount of resources to the user and fixing some transmission parameters such as delay and priority. The packets of a video stream, however, do not all have the same importance. Because of their error resilience features, the spatial encoded pictures have to be protected better than the temporal encoded pictures. In order to optimize the available resources the network has to be made aware of the importance of the packet's payload.

QoS specifications have been both defined by International Engineering Task Force (*IETF*) for the *IP* world as well as by *3GPP* for *UMTS*. Signalizing the *QoS* setting of a packet at *IP* level is not sufficient as the *UMTS* Terrestrial Radio Access Network (*UTRAN*) elements do not understand the *IP* protocol. Approaches relying on the mapping of the *IP* *QoS* classes in those of the *UMTS* [26, 27] does not discriminate between packets belonging to the same application but with different importance. The solution proposed in this doctoral thesis consists of two logical channels with different quality settings are established between the access network and the user. As they are both associated to a single *IP* address, this mechanism is transparent to the application layer. The packets are then filtered depending on the type of frames they contain and associate with the appropriate logical channel, as described in [28, Superiori et al.] and [29, Superiori et al.]. The results have been obtained by means of an *HSDPA* system level simulator, able to adapt the transmission parameters to the importance of the packet.

Chapter 6: Conclusions A summary of the results of this thesis as well as some considerations about possible future work are offered in this chapter.

Appendixes Further introductory information is provided in the appendixes. Appendix A offers a detailed description of the H.264/*AVC* standards, focusing on the codec characteristics discussed in this thesis. Appendix B describes the technologies for delivering data over wireless cellular networks: General packet radio service (*GPRS*), *UMTS* and *HSPA*.

Chapter 2

Video Streaming over 3G Cellular Networks

THE notion of *multimedia streaming* includes the collection of techniques for delivering multimedia contents from a *streaming server* to an *end user*, the latter being able to play the multimedia content before it has been completely downloaded first.

The multimedia server may vary from professional equipments (TeleVision (TV) studios, video hosting/sharing websites) to small handset devices (mobile phones capable of video calls), passing through medium budget instruments (personal computers connected to the internet). The end user is assumed to be a human being, accessing the multimedia contents through different classes of equipments, from high quality and high definition (TV screens) through average quality and average resolution (personal computer) to low quality and low resolution equipments (mobile devices). An appropriate *transmission technology* is in charge of delivering the multimedia content from the streaming server to the end user. Different transmission technologies are utilized depending on the end user capabilities, on the necessity of a feedback channel and on the amount of users receiving the same multimedia content.

Almost all the possible combinations of streaming server, transmission technology and terminal classes are allowed, possibly including some interfaces scaling resolution and data rate to the user equipment requirements.

This chapter is structured as follows. Section 2.1 describes three basic video streaming paradigms: broadcast, multicast and unicast. Section 2.2 contains a brief introduction over three streaming technologies: PSS, the one considered in this doctoral thesis, MBMS and DVB-H. Finally, conclusive comparisons and criticism are summarized in Section 2.3.

2.1 Video Streaming Paradigms

In this section, three video streaming paradigms are presented. Their definition is not dependent on the streaming server or on the user class considered, but rather on the transmission technology and on the number of users receiving the same multimedia content as well as on the level of interactivity they allow for.

2.1.1 Broadcast

Broadcast is probably the most common streaming paradigm. The definition of broadcasting for multimedia is equivalent to its specification for computer networks: the same content is delivered simultaneously to all the users attached to the network. In this paradigm, the source is broadcasting a multimedia stream over the network (both wired and wireless), reaching a set of users that share a given amount of common capabilities.

The most known and advertised examples of multimedia broadcasting is the Digital Video Broadcasting (DVB) family, in particular the Digital Video Broadcasting Satellite (DVB-S) [30], Digital Video Broadcasting Terrestrial (DVB-T) [31], currently substituting the analog television broadcast, DVB-H [32] and Digital Video Broadcasting Satellite to Handheld (DVB-SH) [33]. The first two specifications address the delivery of high quality streams, in this context the term *quality* is strictly related to the transmission *data rate*, and high resolution video streams. The users are typically static (DVB-S and -T streams are usually displayed in a TV screen) and connected to the network by means of a satellite dish (DVB-S) or a stub antenna (DVB-T). DVB-H (see Section 2.2.2) and DVB-SH are designed to serve nomadic users displaying the multimedia content into a portable device (handheld). The considered resolutions are smaller, the quality is strongly dependent on the settings of the service provider. The network access is provided by an appropriate antenna in the mobile device. An equivalent delivery mechanism for mobile devices is the Multimedia Broadcast Multicast Service, standardized by the 3GPP (see Section 2.2.2).

In the Internet, broadcasting is mostly utilized for transmitting live events (sports, demonstrations, political events and more). When receiving broadcast transmissions, the end users do not have control on the time when a given program is aired, they cannot pause and playback (if the device is not implementing a buffer storage). The users can select the content among a finite number of programs depending on the chosen technology. Another common characteristic of the broadcast, is the limited capabilities of the feedback channels. Since all the users are receiving the same stream, it is not possible to adapt the transmission to the needs of a specific user. Any form of interactivity is not supported.

2.1.2 Multicast

Multicast, as in the wired networking, is defined as a transmission of the same content to a subset of the users attached to the network. These users have to join a specific multicast group. In other words, the network elements discriminate whether there are some multicast users in the subnet they are serving: if multicast receivers are present, they will forward the stream, otherwise they will drop the stream. This decreases the network load, as the resources are occupied only in case one or more users are listening.

The practical difference between broadcast and multicast in wireless cellular networks is basically the way the services are billed. Most of the time the broadcast streams are free for the customer of a given provider, whereas the multicast streams require an explicit subscription that is charged accordingly.

In terms of resource allocation, multicasting offers more benefits for wired networks (Internet) with intermediate nodes (switches) rather than for wireless networks. The network load in wireless

broadcast and multicast transmissions is, in fact, not dependent on the number of users attached to the network. In wired transmissions, instead, the stream has to be carried through the network to all the users. If some of them are not listening, network capacity is wasted.

Multicast makes use of the interaction of two Internet protocols: the Internet Group Management Protocol (IGMP) [34] and the Protocol Independent Multicast (PIM) [35]. By means of the former, the receiver signalizes to the network that it is interested on receiving the data sent to a specific group. The latter protocol is instead employed to build distribution trees by means of data replication. Each node of the network delivers the stream further only if it has received the appropriate IGMP message. This transmission paradigm is used mainly for Internet Protocol TeleVision (IPTV). As for broadcasting, multicasting does not foresee a mechanism.

2.1.3 Unicast

Unicast refers to an end to end transmission from the server to a unique user of the network. In the Internet world, the most common multimedia unicast services are the video sharing sites, such as Youtube. Since the user decides both the content of the multimedia library he wants to access as well as the timing, the connection is labeled as “on demand”. The 3GPP defined the specification for unicast transmissions in the 3GPP PSS (see Section 2.2.1) specifications.

From the server point of view, this is the most demanding services as, in the previous two configurations, a single stream is now serving a single user. Also from the network point of view, unicast is the most resource consuming type of transmission, particularly when considering a wireless network. Because of the scarcity of the available spectrum, unicast transmissions generate a high amount of traffic that, possibly, may exceed the available capacity. Therefore, for unicast connections an *admission control* is necessary.

Unicast is the transmission paradigm that allows for the highest grade of interactivity. Since each connection is dedicated to a single user, he has the full control over the stream. Fast forwarding, pause and replay are allowed without the necessity of buffering the whole stream.

2.2 Streaming Technologies

In this section, the three streaming technologies are described. A detailed description is reserved for the PSS in Section 2.2.1, as it is the streaming technology this doctoral thesis is addressing. This section does not claim to be an exhaustive explanation of PSS, but rather is intended to offer an overview of the features exploited in the following investigations. A brief overview of MBMS and DVB-H is provided in Section 2.2.2.

2.2.1 3GPP PSS

3GPP PSS [36] is the specification for unicast video transmission describing how the transparent end-to-end packet switched streaming services has to be implemented in third generation mobile networks. The PSS mostly deals with application level services, therefore the protocol stack, sketched in Figure 2.1, covers the layers between application and IP. This doctoral thesis is focused on the leftmost part

Video Audio Speech	Scene description Still images Text	Capability exchange	
Payload format	Synthetic Audio		
RTP	HTTP	RTSP	
UDP	TCP		UDP
IP			

Figure 2.1: PSS services and protocols.

of the table.

In the following the protocol stack for **PSS** applications, as drawn in Figure 2.2, is discussed. The

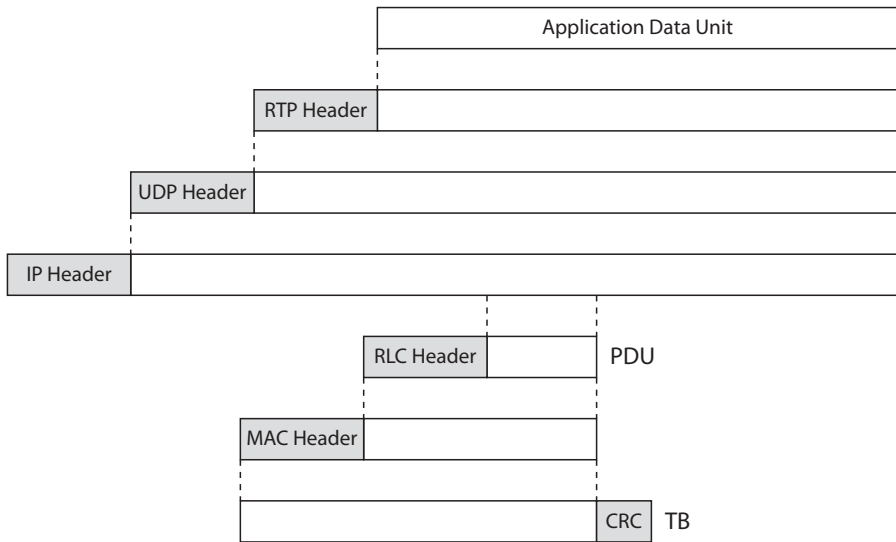


Figure 2.2: PSS protocol stack.

application layer payload is the packet produced by the application, a video and/or audio encoder for multimedia streaming. The application payload is further encapsulated into the Real Time Protocol (RTP) [6, 37–39]. The RTP standardizes a packet format for the end-to-end delivering of audio and video through the Internet. The RTP is labelled as a session oriented *unreliable* protocol, since it does not guarantee the delivery of the packet nor offer any mechanism for recovering packet loss. In the RTP headers (12 bytes), the sequence of packets is marked by means of a unique *sequence number* and a *timestamp* shared by all the packets referred to the same sample, also useful to synchronize different sources such as video and audio. This allows the application to sort out the packet sequence, remove duplicated packets and, possibly, reacting to missing packets or to packets arriving too late. To fight the jitter, that is the variation of the packet’s arrival time, a *de-jitter buffer* is implemented at the application layer. The size of the jitter buffer has to be selected carefully, since a small jitter buffer would increase the packet loss rate and a large jitter buffer would add unnecessary delay.

The UDP [40] and the Transport Control Protocol (TCP) [41] are the two transport layer protocols commonly used in the Internet. The TCP offers a guaranteed transport services based on retransmission for missing or damaged packets. As this causes unpredictable transport delays not allowed in real

time communications, TCP is not used. UDP, on the other hand, is again a simple and unreliable datagram transport protocol. In the UDP header, 8 bytes long, the source and destination port are specified, as well as the length of the packet. Moreover, a sixteen bit long checksum is utilized for verifying the correctness of the header and of the payload.

IP [42] is employed as network layer protocol. As a detailed discussion of the IP header is far besides the scope of this work, only the necessary features will be presented. The IP is a *connectionless protocol* and works under the assumption of an unreliable network dynamically changing its structure. The intelligence is only located at the ends of the transmissions, since the intermediate nodes just forward the information according to routing tables. This may cause packet loss, duplication of packets, data corruption as well as out of order data delivery. The IP header consists of 20 bytes, containing, besides other information, source and destination IP address, packet length and a checksum calculated over the header itself. In Chapter 5.2.2, a scheme exploiting the Type of Service (ToS), one of the IP header fields, is described.

For multimedia streams, the IP is in charge of splitting and recollecting the Service Data Unit (SDU) exceeding the network's Maximum Transfer Unit (MTU). In order to minimize the delay and avoid packet fragmentation, the size of the packet at the application layer is bounded by the MTU (for wireless networks 1500 bytes) minus the lower layers header size. However, the size of the IP packet for multimedia streaming has to be chosen carefully [43]. The usage of payload sizes close to the MTU increases the end-to-end latency and, in case of packet loss, makes the concealment of the missing information more challenging. The usage of small payload sizes, on the other hand, causes the increase of the packet header overhead, as shown in Figure 2.3, and, by increasing the packet rate,

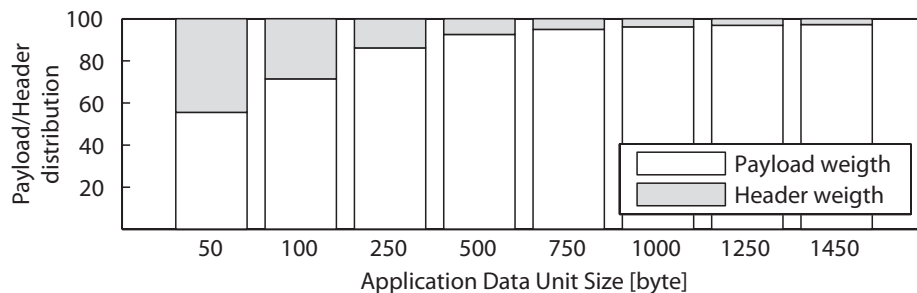


Figure 2.3: Overhead of the IP/UDP/RTP headers.

loads unnecessarily the server and the network elements.

The IP packets are further transmitted over the UTRAN [44]. In UMTS the IP packets are further segmented at the Radio Link Control (RLC) [45] into RLC Protocol Data Unit (PDU). The size of the RLC PDU payload depends on the data rate associated to the bearer assigned to the user. In the reasonable case of a data rate smaller than 384 kbit/s, the size of the payload is 320 bits. The RLC of the UTRAN defines an Acknowledgment Mode (AM), allowing the receiver to detect errors and feedback to the server the status of the received packet, and an Unacknowledgement Mode (UM), enabling only the detection but no feedback. The size of the RLC header is 16 bits for AM and 8 bits for UM, respectively.

The Medium Access Control (MAC) [46] layer is responsible of mapping the packet onto the

transport channel. The **MAC PDU** consists of the **RLC** packet and a **MAC** header and it is usually referred to as **TB**. One byte Cyclic Redundancy Check (**CRC**) is added to each transport block for verifying at the receiver side the correctness of the **MAC PDU**.

The **3GPP PSS** specifications define how the end-to-end connection has to be managed. In order to establish a connection, the client accesses a Real Time Stream Protocol (**RTSP**) address and receives back a Real Time Stream Protocol (**SDP**) file containing the multimedia stream addresses and the timing information [47].

Next, the client and the server go through the *capability exchange*, a process where the hardware and software features of the client are compared with those the server offers. The **PSS hardware vocabulary** comprises the vendor and the model name, the screen resolution, the pixel aspect ratio and more. The software vocabulary consists of, among others, the supported decoding bit rate, the decoder buffer size and the supported *multimedia codecs*. The specification [48] describes two groups of codecs: the mandatory and the optional codecs. Since this doctoral thesis is dedicated to the multimedia video stream, in the following only the video codecs are discussed. A **PSS** compliant client has to support the International Telecommunication Unit standardization sector (**ITU-T**) Video Coding Expert Group (**VCEG**) H.263 [49] profile 0 level 45. Optionally, the client has to support the ITU H.263 profile 3 level 45, the International Standard Organisation (**ISO**) Moving Picture Expert Group (**MPEG**) MPEG-4 visual simple profile level 3 (with a set of constraints) [50] as well as the state of the art (at the time of writing of this doctoral thesis) the **ITU-T/ISO H.264/AVC** (see Appendix A) [51, 52].

The **3GPP PSS** specifications describe also the **QoS** parameters for the end-to-end connection. This aspect is further investigated in Chapter 5.

2.2.2 Broadcast Technologies for Mobile Devices

MBMS [53] is a streaming service standardized by **3GPP** for transferring audio and video to several users. Due to the nature of the **3GPP** connection, an uplink (feedback) channel is available, although, being the service shared with other users, the interactivity is limited. In this extent, the difference between broadcast and multicast relies on the fact that broadcast is a *push-type* service, whereas multicast needs an uplink channel in order to let the users subscribe the stream.

Digital Video Broadcasting for Handheld is a mobile **TV** format developed by the European Telecommunication Standard Institute (**ETSI**) and adopted as standard EN 302 304 in November 2004. **DVB-H**, as the **TV** formats for non nomadic users **DVB-T** and **DVB-S**, is a broadcast system that allows the subscribers of the service to access to a limited number of channels. The **DVB-H** format is based on the **DVB-T**, with improved features for ensuring enhanced mobile **TV** experience. In particular, a time slice technique has been implemented in order to save battery life by keeping the receiver alive only in the time slice where data is transmitted. Moreover, an enhanced error correction mechanism, MultiProtocol Encapsulation (**MPE**) Forward Error Correction (**FEC**), fights transmission errors.

The difference between the **3GPP** broadcasting and **DVB-H** lies in the technological implementation, since the service offered to the customer is basically the same. The most relevant difference is the infrastructure. **DVB-H** needs its own infrastructures whereas **MBMS** utilizes the already available

cellular network. **DVB-H** does not need a cellular system since all the customers are sharing the same content. The number of necessary **DVB-H** transmit stations depends only on the feasible transmission coverage and not on the number of users that have to be served. Considering the achievable bit rate, in **DVB-H** the throughput per carrier is limited by the battery saving target whereas in **MBMS** the limitation of the base station output power consumption is driving the performance of the system.

2.3 Comparison and criticism

The acceptance of mobile **TV** is, at the time of writing of this thesis, far below the expectations. Although the European Commission is pushing and encouraging the usage of mobile **TV** by electing **DVB-H** as the mobile **TV** standard, all the forecasts made from the introduction of the standards have been corrected for being too optimistic, as the market is not accepting enthusiastically the mobile **TV** services currently offered. In the history of telecommunication, there is plenty of services that unachieved the expectations: in order of appearance, Wireless Application Protocol (**WAP**), Multimedia Messaging Service (**MMS**), video call and mobile **TV** were supposed to be the *killing applications* for the introduction of the respective technological breakthrough.

Since the technological limitations (bandwidth, delay and more) have been overcome by the current transmission systems, the success formula for the mobile **TV** has two input variables: the price and the content. Both strictly linked by each other. The main complaint regards the choice of the channels offered that reflects the same of the standard **TV**. The users are not willing to pay additional money for the same content they have available with their **DVB-T** subscription.

Moreover, the standard **TV** channels programming is not suited for the nomadic usage of the mobile **TV** consumers. According to a survey [54], mobile **TV** users access video contents mostly when waiting or when using public transportation. The expectation of the customers is, therefore, to watch a content starting in the moment they turn on their mobile device and which duration is compatible with the time they have at disposal. Wireless video broadcast systems do not offer, by definition, any on demand service. The user has no influence on the air time of the content and there is not the possibility of continue playing the content from the point it has been stopped last time.

On the other hand, unicast transmission (such as Youtube), allowing users to fill in waiting time watching mobile **TV**, are extremely resource consuming for the mobile operator that would be forced to charge it accordingly making the cost unattractive. Another key question is the availability of **DVB-H** capable terminals that are not common in the market and, usually, require an external antenna to be used to receive the signal.

The reception of the service is depending on the world location. Despite a discrete success in Asia, in Europe several mobile operators turned off (Virgin Mobile) or are considering of interrupting the offered mobile **TV** services. According to a survey performed in Austria in August 2008 [55], three fourth of the mobile users reported to be *uninterested* in mobile **TV**. This figure varies from country to country, but still remains far behind when compared to the mass market diffusion of mobile **TV**.

Chapter 3

Detection of Errors in Corrupted Video Sequences

THE transmissions of data over wireless channels are affected by errors due to multiple causes, such as poor channel condition, interference and noise at the receiver. For non delay sensitive services, such as Internet browsing or File Transport Protocol (FTP) traffic, the application requires for retransmission of the incorrectly received packets.

For video streaming applications missing and damaged packets have to be handled differently. For broadcast and multicast transmissions, the retransmission of the damaged or not received packets is not possible. A single source is serving several users and, even assuming the presence of a feedback channel, cannot react to the needs of a single user. Retransmission can be problematic also for unicast transmissions. In case of network congestion, the overloaded cell may not be able of retransmit the damaged packet before the time constraint placed by the playout buffer is reached.

In a wireless scenario the application has to cope with missing or damaged packets as well as with packets arriving too late [56]. The application reacts by *concealing* the missing information. To conceal, that is *to hide*, the error basically means replace the missing or invalid data with other information extrapolated from the already available video content. The video concealment can be of spatial, temporal or hybrid nature. In the spatial case, the missing information is concealed interpolating the available surrounding macroblocks. In the temporal concealment, a motion compensation is estimated on basis of the already decoded macroblocks or on the previous picture's motion characteristics. The hybrid concealment methods consist of a mixture of the two approaches, depending on the frame type as well as on the picture properties.

In this chapter different error detection mechanisms for detecting errors in *damaged packets* are presented. After briefly presenting the effects of errors at frame level and at packet level, in Section 3.1 and 3.2, respectively, different error detection mechanisms are presented in the following sections.

3.1 Effects of Errors at Frame and Sequence Level

In the different layers of the 3GPP PSS protocol stack, shown in Figure 2.2, there are already

strategies for deciding whether an IP packet has been incorrectly received or not.

A *checksum* is included for evaluating the correctness of a bit sequence. Given a generic sequence of bits, being this the packet payload and/or header, the checksum is a fixed block of data whose value depends on a specific characteristic of the bit sequence. The simplest algorithm is the calculation of the parity words. Once decided the length of the checksum word, say n , the bit sequence is segmented into blocks of n bits. The checksum is then obtained as the exclusive or of all the n -bits long blocks. If a transmission error corrupts part of the payload, the checksum calculated at the receiver side will not match the encoded one.

In UDP, a two bytes checksum is calculated using a slightly more complex procedure. The UDP header, the UDP payload and, if necessary, padding bits are considered for computing the checksum. The one's complement sum of a two byte long data blocks is calculated and its 16-bit one's complement is stored as checksum. Also the transport blocks possess a CRC that allows the detection of damaged TBs, however the application has no direct access to this information and specific cross layer information has to be conveyed to make it available (see Chapter 5).

In case a packet is detected as damaged at UDP level, it can be discarded and not conveyed further to the RTP. The effects are the same as in the case of IP packet loss. An RTP header field, the *sequence number*, is used for sorting the received RTP packets in the playout buffer. In case the application notices discontinuity in the packet sequence and cannot allow for retransmissions, the missing information is concealed.

An estimation of the missing data is exploited for displaying the video content on the screen and impair as little as possible the user experience. However, even if the reconstruction at the instant t of the missing part of the picture may not be annoying for the users, the concealed picture is further used as a source of prediction by the following Inter predicted frames. Since the prediction is computed at the encoder side (considering the error free sequence as reference), if a concealed frame is used at the decoder side, a drift between encoder and decoder references is introduced.

This effect is shown in Figure 3.1. As a packet of F_2 has been incorrectly received, the corresponding

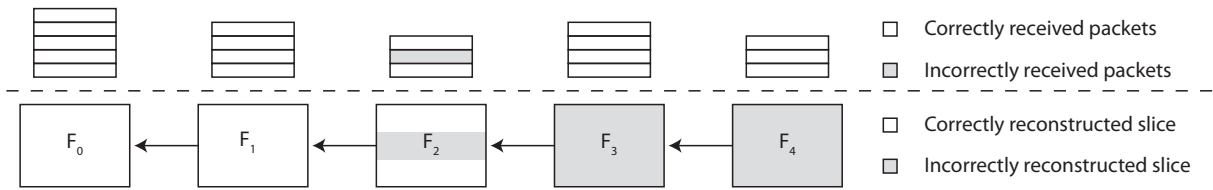


Figure 3.1: Temporal error propagation.

slice of F_2 is incorrectly reconstructed or concealed. The frame F_3 , although all the packets have been correctly received, is employing a corrupted reference for performing temporal prediction. This effect, defined in the following as *temporal error propagation*, involves all the following Inter encoded pictures until the end of the Group Of Picture (GOP). Only the next Intra predicted frame does not contain reference to the previous pictures. As Inter-GOP prediction is allowed, the insertion of I frames is not sufficient to stop the temporal error propagation. If the reference buffer size is larger than one, the first inter predicted frame of one GOP may use as reference a frame belonging to the previous GOP. The temporal error propagation is interrupted only in case the reference buffer contains only

one picture or if the I frame empties the content of the buffer. In this last case the I frame is called Instantaneous Decoding Refresh (IDR) frame.

3.2 Effects of Errors at Packet Level

Even though the UDP packet fails the checksum test, it can be still conveyed further to the RTP layer and the Network Abstraction Layer Unit (NALU) can be handed as input to the video decoder. In this case, the Forbidden (F) bit of the NALU header is set to one. This informs the decoder that it has possibly to cope with a damaged payload. In the following, the results of the analysis of the effects of errors in damaged NALUs are presented. The analysis has been performed studying deeply the structure of the code as explained in the standard and the standard development code Joint Model (JM) [57].

In order to fully understand the content of this section, it is suggested to refer to the description of the different entropic encoding strategies defined in H.264/AVC (see Section A-3) and to the overview of the encoded elements stored in the NALU (see Section A-4).

In the following, the effect of errors in the bitstream is discussed. Depending on the entropic coding style and on the affected syntax element, different effects arise. The case of codewords with variable length is handled considering exp-Golomb encoded codewords and subsequently generalized. The structure of an exp-Golomb encoded codeword has the following form:

$$\underbrace{0_1 \dots 0_M}_M 1 \underbrace{b_1 \dots b_M}_M.$$

The codeword consists of M zeroes, the *leading zeroes*, one “1” followed by M bits, the *info field*. The number of bits in the info field, M , depends on the number of zeroes preceding the first one. The number of leading zeroes forces, therefore, the length of the word itself to be equal to $2M + 1$. The effect of even a single bit inversion in the first M zeroes is depicted in Figure 3.2. The original

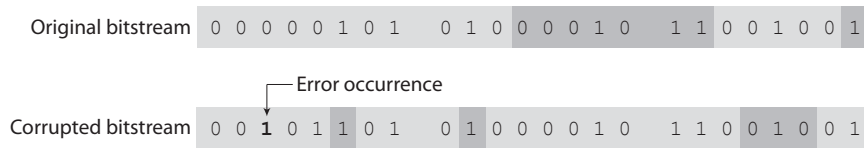


Figure 3.2: Decoding desynchronization at bitstream level.

bitstream consists of four words of length 11 ($M = 5$), 7 ($M = 3$), 5 ($M = 2$), 1 ($M = 0$), respectively. Assuming now an error inverting the third bit, in the corrupted bitstream a “1” is found in the third position. The first decoded codeword has, therefore, $M = 2$ and length five. As the boundaries of the first codeword have been misinterpreted, the following codewords are incorrectly interpreted as well. As no resynchronization words are allowed in H.264/AVC, the *decoding desynchronization* propagates until the end of the NALU.

For exp-Golomb encoded codewords, one error occurring in the leading zeroes or in the first “1” causes decoding desynchronization, as the parameter M is incorrectly evaluated. In case the info field

is corrupted, as the parameter M is not changed, there is no direct desynchronization caused by the misunderstanding of the codeword boundaries. However, as the encoded value is incorrectly reconstructed, the decoder can misinterpret the significance of the following decoded codewords. Assume that an error is affecting the info field of the codeword, indicating the number of macroblock subblocks. Although no direct desynchronization occurs, the decoder will read the necessary parameters (such as motion vectors) for a wrong number of partitions, leading again to decoding desynchronization.

A similar behavior is observed for tabled codewords or Variable Length Coding (VLC) levels affected by errors. The structure of the Video Coding Layer (VCL) codewords is similar to that of the exp-Golomb. The first $M + 1$ bits of the Context Adaptive Variable Length Coding (CAVLC) words are highly sensitive to errors, since, as for the exp-Golomb codewords, a single bit inversion leads to the misinterpretation of the length of the prefix, causing decoding desynchronization. Errors in the info field cause the incorrect decoding of the stored value. This will for sure affect the decoded picture in the pixel domain and, possibly, cause the choice of the wrong VLC- N routine, Table A.1, for the following levels. An error in the sign has only effects in the pixel domain. For tabled codewords, the decoder may choose a word of different length (causing direct desynchronization) or just reconstruct the wrong value, possibly inducing the misunderstanding of the following parameters.

Summarizing, the entropic coding in H.264/AVC makes the codewords extremely sensitive to bit inversions. As no resynchronization words are applied, if the error causes decoding desynchronization, this effect lasts until the end of the NALU.

3.3 Syntax Analysis

After introducing the effects of errors both at sequence and packet level, in this section the proposed error handling mechanism will be discussed. It has been described how the errors only affect the decoding of the current and, possibly, following codewords. The information elements preceding the error occurrence can still be correctly decoded. Since the position of the error is not known, the definition of efficient error detection mechanisms is a task of major importance. In the following, an error detection mechanism based on the code syntax analysis will be presented. A similar method for the H.263 codec was proposed in [11]. However, H.263 does not make an extensive use of variable length coding as H.264/AVC does and, moreover, allows for resynchronization words between Groups Of Blocks (GOB).

The detection mechanism described in this section has been implemented in the standard reference software JM version 10.2 [57]. The analyzed decoder functionalities have been subdivided into the following two elements:

1. Read: Evaluate the value obtained when reading a codeword.
2. Decode: Use the obtained value to reconstruct the macroblock.

This conceptual distinction is consistent with the two different logical functions defined in the JM: Read and Decode one macroblock, as indicated in Figure 3.3.

In the first function, Read, each codeword is interpreted, obtaining the encoded value. In the second function, Decode, such value is then employed to reconstruct the picture in the pixel domain, applying, for example, motion compensation or correction by means of residuals.

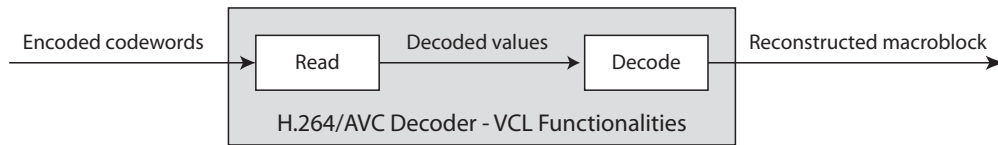


Figure 3.3: H.264/AVC decoder blocks.

It has been noticed that the original **JM** software is not able to handle errors in the bitstream, as they make the decoder crash. Such crashes are due to unexpected codewords or values causing exception in the reading or decoding phase. Observing the behavior of the decoder in case of error, different typologies of exception has been defined:

- **Out of Range codewords (OR)**. **OR** occurs when the value associated to the decoded codeword lies outside an admitted range. This may happen, for example, when reading the number of skipped macroblocks. In case of error, the number of skipped macroblock may exceed the number of macroblocks not yet decoded.
- **Illegal Codewords (IC)**. **IC** occurs when the codeword does not find any correspondence in the appropriate look up table. This may happen when using incomplete look-up tables that contain only a subset of the possible codewords with a given length.
- **Contextual Error (CE)**. This kind of errors arises when the value obtained in the read phase leads the decoder to perform forbidden actions. For example, one bit inversion may change the codeword associated to the intra prediction mode of the first macroblock of a slice. At bitstream level this error cannot be detected. However when applying the required prediction mode in the decode phase, no prediction mode different from Direct Current (**DC**) (the average value of the block is predicted without referencing any neighbor) is allowed.

These kinds of errors can be tracked during the decoding, and the crashes of the decoder can be avoided setting appropriate exception catching strategies. However, these errors may not be caused directly by bit inversions affecting the currently read codeword, but rather be caused by decoding desynchronization due to errors occurred before. The distance between the error occurrence and the error detection will be defined as *detection distance*. This distance can be expressed in bits, but it is more meaningful to express it in terms of macroblocks. The information encoded between the error occurrence and the error detection will be incorrectly decoded, resulting in impairment in the reconstructed picture. The detection distance, therefore, has to be kept as small as possible. In order to fulfill this requirement, all the information elements of the encoded stream has been analyzed and, for each, the typology of error that can arise has been identified. For sake of completeness, also the encoding style (Exp-Golomb codewords (**EG**), Tabled Encoding (**TE**), Fixed Length (**FL**) and **VLC**) has been mentioned. The name of the parameters are those of the standard development software, they match, besides nomenclature, those defined in the standard.

I Frames

Parameter Name	Encoding	Error
<code>mb_type</code>	<i>EG</i>	<i>OR</i>
<code>intra4x4_pred_mode</code>	<i>TE</i>	<i>CE</i>

Since the spatial prediction uses reference to the surrounding macroblocks, if they are not available, not yet decoded or belonging to another slice, a contextual error is produced.

<code>intra_chroma_pred_mode</code>	<i>EG</i>	<i>OR</i>
<code>coded_block_pattern</code>	<i>EG</i>	<i>OR</i>
<code>mb_qp_delta</code>	<i>EG</i>	<i>OR</i>
<code>Luma(Chroma) # c & tr.1s</code>	<i>TE</i>	<i>IC</i>

The look-up table used to decode this value is not complete. The decoded codeword cannot find reference to any legal value.

<code>Luma(Chroma) trailing ones sign</code>	<i>FL</i>
--	-----------

The signs of the trailing ones are fixed length encoded and do not influence any of the following parameters. By means of syntax check it is not possible to detect such errors.

<code>Luma(Chroma) lev</code>	<i>VLC</i>	<i>OR/CE</i>
-------------------------------	------------	--------------

Decoded macroblock pixels can only take values lying in the range $[0,255]$. During Read phase, values outside the bounds are associated to errors. During the Decode phase, the residuals are added to the predicted values and the contextual check is performed. An extended range $[-\lambda, 255 + \lambda]$ is considered due to possible quantization offset.

<code>Luma(Chroma) totalrun</code>	<i>TE</i>	<i>IC</i>
------------------------------------	-----------	-----------

<code>Luma(Chroma) run</code>	<i>TE</i>	<i>IC/OR</i>
-------------------------------	-----------	--------------

Depending on the number of remaining zeros, a *VLC* look-up table is chosen. For more than six remaining zeros, a single table covering the zero run range $[0,14]$ is used. Therefore, the decoder is exposed to out of range errors.

P Frame

Many of the parameters encoded in P frames are equivalent to those utilized to describe an I frame. In the following only the parameters specific for Intra encoding are discussed.

<code>mb_skip_run</code>	<i>EG</i>	<i>OR/CE</i>
--------------------------	-----------	--------------

The number of skipped macroblocks cannot be greater than the number of not yet decoded MacroBlock (*MB*)s belonging to the current frame

<code>sub_mb_type</code>	<i>EG</i>	<i>OR</i>
<code>ref_idx_l0</code>	<i>EG</i>	<i>OR/CE</i>

The index of the reference frame cannot be greater than the actual reference buffer size

<code>mvd_l0</code>	<i>EG</i>	<i>CE</i>
---------------------	-----------	-----------

Slice Header

<code>first_mb_in_slice</code>	<i>EG</i>	<i>OR</i>
<code>pic_parameter_set_id</code>	<i>EG</i>	<i>OR/CE</i>

The VCL-NALU cannot reference a Picture Parameter Set (PPS) index greater than the number of available PPSs.

<code>slice_type</code>	<i>EG</i>	<i>OR</i>
<code>frame_num</code>	<i>EG</i>	<i>OR</i>
Depending on the GOP structure, out of range errors can be detected		
<code>pic_order_cnt_lsb</code>	<i>EG</i>	<i>OR</i>
<code>slice_qp_delta</code>	<i>EG</i>	<i>OR</i>

3.3.1 Error Handling Mechanism

Once the rules of the syntax analysis have been discussed, the three compared handling mechanisms are described. The macroblocks marked as corrupted will be concealed by means of a simple copy-paste mechanism. Assuming that the macroblock in row i and column j of the frame f , $MB_f(i, j)$ is labeled as corrupted, it will be replaced by the macroblock $MB_{f-1}(i, j)$ occupying the same position in the previous picture.

1. **Straight Decoding (SD)**. This approach consists of the decoding of the corrupted bitstream by means of a modified H.264/AVC decoder. Without any detection mechanism, in case one of the syntax errors is detected, the decoder replaces the invalid value with the its closest valid value. Since no error detection is performed, the macroblocks preceding the error occurrence are correctly decoded, whereas the following ones are incorrectly decoded.
2. **Slice Level Concealment (SLC)**. This approach represents the standard error handling mechanism currently considered in the literature. In case a packet fails the UDP checksum test, it is discarded. All the macroblocks contained in the NALU are marked as corrupted and concealed. Also the macroblocks preceding the error are concealed, even if this is not necessary.
3. **MacroBlock Level Concealment (MBLC)**. This handling mechanism exploits the syntax analysis described before. Even though the packet fails the UDP checksum test, it is decoded. If a syntax error is detected, the current macroblock as well as all the following ones until the end of the slice are marked as erroneous. As discussed before, the macroblock where the error is detected is not necessarily the one where the error occurred. This causes the macroblocks between the error occurrence and the error detection to be, possibly, incorrectly decoded. However, the information preceding the error detection is correctly exploited whereas the data after the error detection is marked as corrupted and concealed. The proposed method represents a compromise between the before mentioned two considered approaches.

In Figure 3.4, the decoding of a picture using the three proposed approaches is discussed. The considered picture is an intra predicted frame, consisting of four slices. An error has been introduced in the second slice, that contains the macroblocks with index from 30 to 59. The error will therefore propagate spatially until the end of the second slice, namely up to the macroblock 59.

In Figure 3.4(a) the decoding of the corrupted packet considering the straight decoding approach is shown. It can be noticed that, beginning from the MB 35, the impairments propagate till the end of the slice. The macroblocks before the error occurrence are correctly decoded.

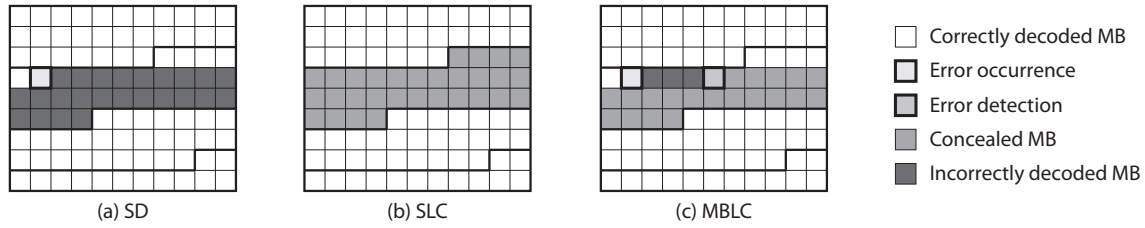


Figure 3.4: Error handling mechanisms.

Figure 3.4(b) shows the result of the decoding when implementing the slice level concealment approach. Since the packet failed the **UDP** checksum test, the whole slice is concealed. The macroblocks from 30 to 34, even if correctly decodable, have been concealed.

The result of the proposed method (MBLC) is shown in Figure 3.4(c). The macroblocks from 30 to 34 are correctly decoded. The error is detected in the macroblock 39, therefore the macroblock from 39 to 59 are concealed. In this example, a detection distance of four macroblocks has been measured.

Figure 3.5 shows a qualitative comparison between the three proposed methods. The cumulative

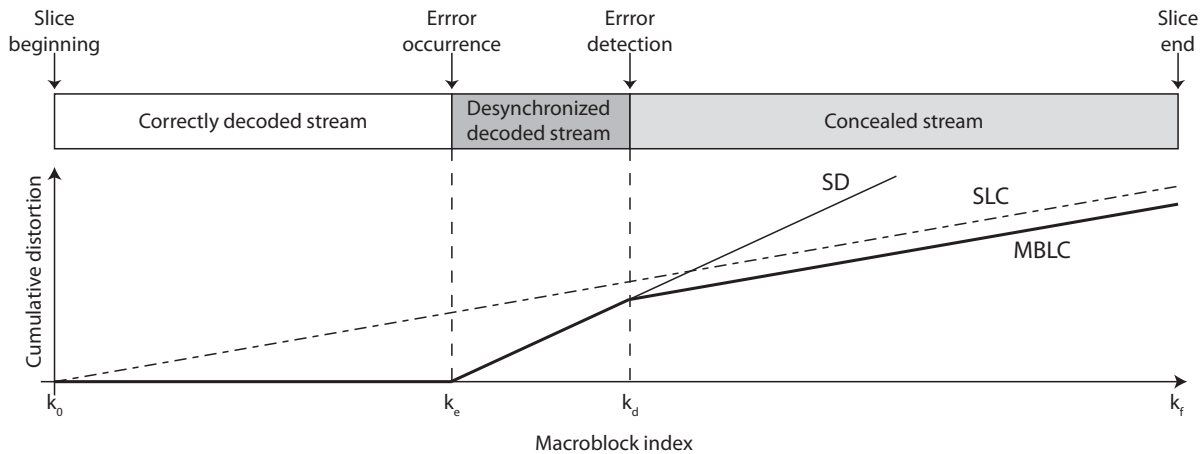


Figure 3.5: Cumulative distortion using the three considered approaches.

distortion between the slice decoded without errors and corrupted slice handled by the three considered mechanisms is displayed. It can be noticed that, using the **SD**, the distortion starts increasing as soon as the error occurs (k_0). The **SLC**, as the concealment is applied to the entire slice from k_0 to k_f , introduces a lighter grade of distortion all over the slice. The proposed approach, the **MBLC**, limits the distortion caused by desynchronized decoding to the macroblocks between k_e and k_d . The distortion introduced by the concealment is spread from the error detection k_d until the end of the slice k_f . The performance of the proposed method depends strongly on the detection distance and on the position in which the error occurs. For errors detected near the beginning of the slice, the **MBLC** acts as the **SLC**. Increasing detection distances affect the performance of the proposed method.

3.3.2 Simulation Setup

In order to evaluate the performance of the three considered methods, the decoding of differently corrupted sequences has been simulated. The sequences were encoded by the standard development encoder without any modification. Depending on the considered handling method, different features in the JM decoder were enabled. In case of SLC, once an error has been detected within the packet, all the macroblocks stored in the NALU are marked as erroneous. When considering SD, all the incorrect codewords or values are turned to a valid value. For MBLC, during the decoding different flags are raised in case invalid codewords or values have been detected. Once a flag has been raised, the decoder stops reading and decoding the macroblock and conceals it. The flags are turned back to zero at the beginning of the following slice.

The sequence used for the simulation is the "Foreman". It consists of 400 frames played at 30 frame per second with a resolution of 176×144 pixels (Quarter Common Intermediate Format (QCIF)). The GOP size was set to 10 and different Quantization Parameter (QP) have been investigated.

3.3.3 Objective Quality Comparison

The first analysis performed regards the resulting quality in terms of Y-PSNR. For a set of sequences obtained by encoding the Foreman video with different quantization parameters, the performance of the three methods have been compared for error patterns characterized a by different Bit Error Ratio (BER). Lower BERs have lower probability of errors, therefore the number of simulations increases with diminishing BERs.

The graphs in Figure 3.6 show the resulting objective quality. The four lines drawn represent,

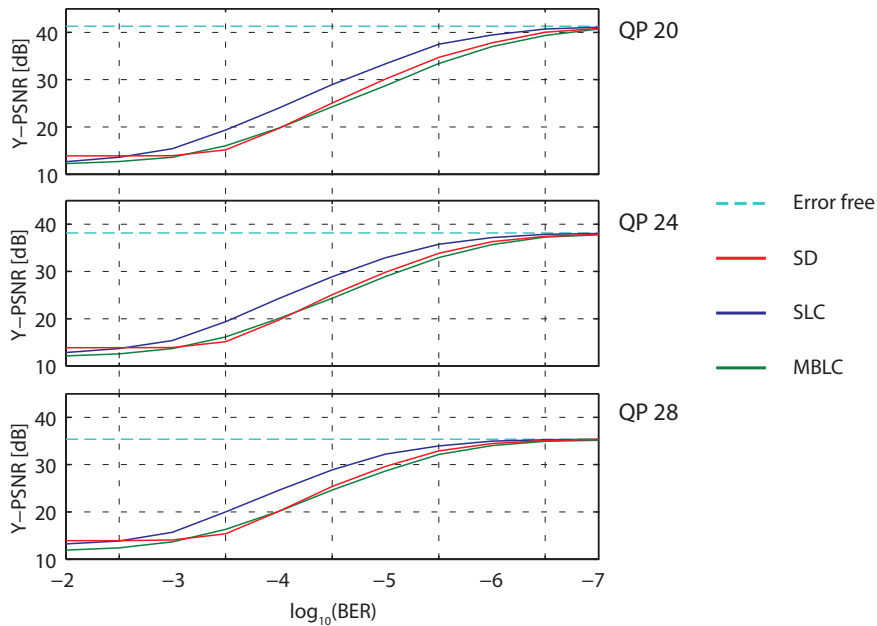


Figure 3.6: Y-PSNR comparison using different handling mechanisms and QPs.

respectively, the quality when considering the error free decoding, therefore not depending on the BER and considered as the reference quality, the macroblock level concealment, the slice level concealment,

and the straight decoding.

The proposed method performs better than the standard handling mechanism, particularly in the range of BERs from 10^{-4} to 10^{-6} . For lower BERs, the three mechanisms are close to the error free case, since the number of errors introduced is not relevant. For higher BERs, such as 10^{-2} , the errors are introduced so frequently that the overall quality of the video is unsatisfactory independently from the chosen handling mechanism.

The curve indicating the performance of SLC describes the behavior of the standard error handling approach. In the following, an analytical model is derived for approximating the empirical results. The simulations have been performed selecting a fixed BER. By means of SLC, the entire slice encoded into a packet is concealed in case the packet's UDP checksum test fails.

The bit error probability has been converted into the frame error probabilities P_I and P_P for the I and P frames, respectively. These probabilities are defined as follows:

$$P_F = 1 - (1 - p_b)^{n_b} \approx \min(p_b \cdot n_b, 1), \quad (3.1)$$

where P_F indicates the generic frame error probability, p_b the bit error probability and n_b the size of the encoded frame. A comparison between the packet error probability calculated by Equation (3.1) and those measured by simulation is shown Figure 3.7.

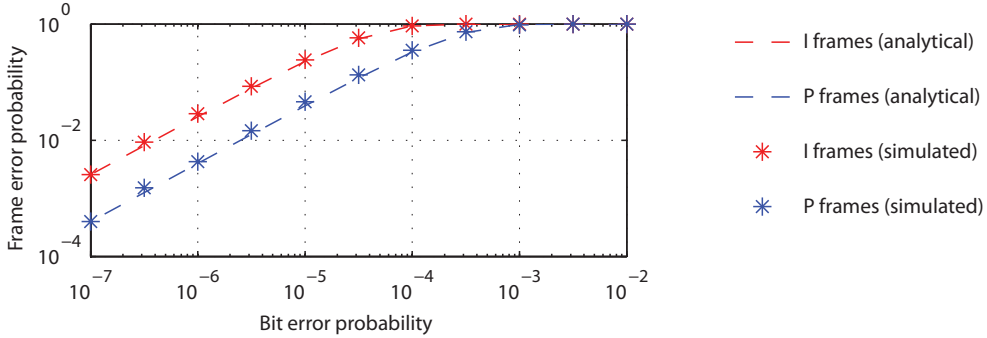


Figure 3.7: Bit error probability mapped to packet error probability.

Considering the chosen encoding settings, the size of an encoded P frame is smaller than the maximum packet size, that is 750 bytes. For the P frames, therefore, the frame error probability coincides with the packet error probability. In the following discussion, all the quality measurements will be performed employing the MSE rather than the Y-PSNR, as the former considers linear values whereas the second a logarithmic scale. Y-PSNR and MSE are linked by the equation

$$\text{Y-PSNR} = 10 \cdot \log \left(\frac{255^2}{\text{MSE}} \right). \quad (3.2)$$

The average MSE of the P frames after transmission, $\text{MSE}_{P,\text{TX}}$ has been expressed as a function of the before mentioned probabilities:

$$\text{MSE}_{P,\text{TX}} = \underbrace{P_I \cdot \text{MSE}_{I,\text{ERR}}}_I + \underbrace{\frac{1 - P_I}{\text{GOP} - 1} \sum_{i=1}^{\text{GOP}-1} ((1 - P_P)^i \cdot \text{MSE}_{P,\text{EF}} + (1 - (1 - P_P)^i) \cdot \text{MSE}_{P,\text{ERR}})}_{II}. \quad (3.3)$$

The quality of the P frames strongly relies on the correctness of the previously decoded frames. Considering a single GOP, and then generalizing, the first underbraced term refers to the quality of the reference I frames. The $MSE_{I,ERR}$ defines the average MSE of an erroneous I frame. In case the I frame is correct (second underbraced term) the average quality of the P frame depends on the previously decoded P frames. The quality of the i -th P frame can be expressed as the probability of the i -th frame and the previous P frames belonging to the same GOP to be correct, $(1 - P_P)^i$, multiplied by the average MSE of the correctly decoded P frames, $MSE_{P,EF}$ ¹, summed to the probability that an error affects the current or the previous P frames multiplied by the average MSE of the damaged P frames, $MSE_{P,ERR}$. For averaging, all the possible GOP positions are considered and averaged.

Once one error has occurred, the following P frames belonging to the same GOP are affected by temporal error propagation. For obtaining the result in Equation (3.3) and those following, two assumptions have been made:

1. A second error occurring in a GOP that has already been affected by temporal error propagation, does not cause additional distortion.
2. The quality degradation after the first error occurrence remains constant. This second argument is true only in average: in practice it may increase, decrease or remain constant depending on the sequence characteristics.

For the I frames, an equivalent formulation describing the average I frames quality, $MSE_{I,TX}$, has been defined:

$$MSE_{I,TX} = \underbrace{(1 - P_I) \cdot MSE_{I,EF}}_I + P_I \cdot N(P_I) \cdot \underbrace{\left((1 - P_{TP}) \cdot MSE_{SI,ERR} + P_{TP} \cdot \frac{MSE_{P,ERR} \cdot \delta_{ERR}}{M} \right)}_{II}. \quad (3.4)$$

The first term of the equation considers the case of error free reconstruction of the I frame, the second refers to the case of reconstruction by means of error concealment. The first term describes the contribution of correctly reconstructed I frames, that occur with probability $(1 - P_I)$.

As mentioned before, an I frame consists of more than a single packet. The size of the packet is fixed in byte, the number of packets necessary for encoding the entire frame depends on the characteristics of each frame. In the considered scenario an I frame consists of 5,33 packets, in average. The factor $N(p_b)$ considers the average amount of corrupted slices with the bit error probability p_b . As described in Appendix A, each I packet is self-contained. It is reconstructed using only the information contained into the packet, without exploiting any data belonging neither to the same nor to another frame. This effect is described in Figure 3.8.

As the distortion of a single slice is not depending on the quality of the other slices belonging to the same frame, the distortion increases linearly with the number of damaged slices. In case one slice is concealed, the measured distortion (the parenthesis in the second term of Equation 3.3) depends on two terms. As the implemented concealment mechanism is the simple copy-paste, in case the I frame is not correctly received the performance depends on the quality of the previous frame, this is the last

¹ $MSE_{P,EF}$ is measured between the reconstructed and the original picture. The distortion is due to the degradation caused by the lossy compression

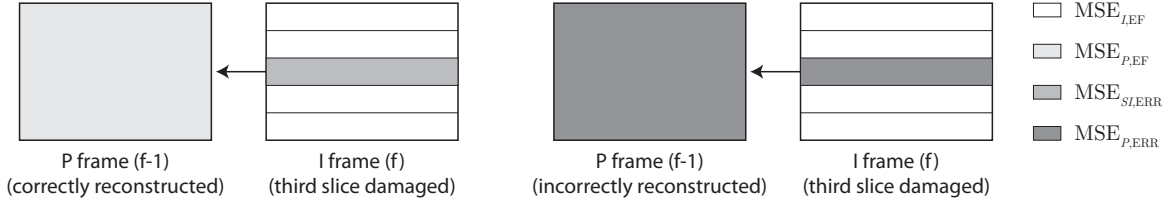


Figure 3.8: MSE of corrupted I slices depending on the quality of the previous P frame.

P frame of the previous GOP. The probability of the last frame not to be correctly reconstructed has been labeled

$$P_{TP} = P_I + (1 - (1 - P_P)^{(\text{GOP}-1)}), \quad (3.5)$$

that is the probability of having a damaged frame in the GOP. In case the previous P frame is correct, $(1 - P_{TP})$, the distortion limited to a single slice is equal to $MSE_{SI,ERR}$. If the previous frame is damaged, P_{TP} , the distortion in the damaged slice is equal to the one measured in the corrupted P frames, but relatively to the concealed region. The term $MSE_{P,ERR}$ is, therefore, divided by the average amount of slices in an I frame, M . The term δ_{ERR} takes into account a systematic error of the model, due to the previously introduced assumptions. All the frames are, in average, damaged for BERs larger than 10^{-4} . This means that each frame is using an already corrupted picture for performing error concealment. When considering these conditions, the assumptions used for building the model are not in force anymore. However, as such a scenario does not provide any acceptable quality it has been preferred to concentrate the reliability range of the model to a reasonable BER interval.

In order to validate the proposed models, the empirical values obtained by simulation have been compared with the analytical estimations. Note that the term I in Equation (3.3) is equivalent to the term II in Equation (3.4). Equation (3.3) and Equation (3.4) are used for calculating the average MSE of P and I frames, respectively, after transmission. The results are sketched in Figure 3.9. .

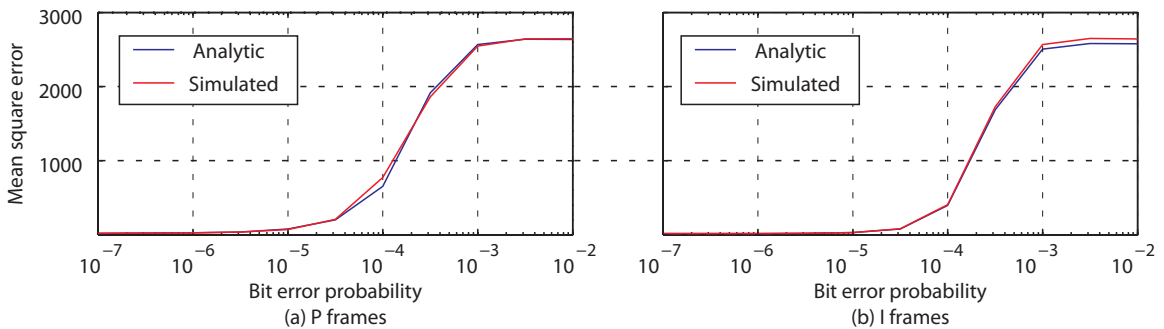


Figure 3.9: MSE after transmission of P and I frames (QP=28).

In order to obtain the average sequence mean square error, MSE_{TX} , the last two results have been averaged:

$$MSE_{TX} = \frac{1}{\text{GOP}} \cdot MSE_{I,TX} + \frac{\text{GOP} - 1}{\text{GOP}} \cdot MSE_{I,TX}. \quad (3.6)$$

The results are plotted in Figure 3.10, both in terms of MSE (left) and Y-PSNR (right).

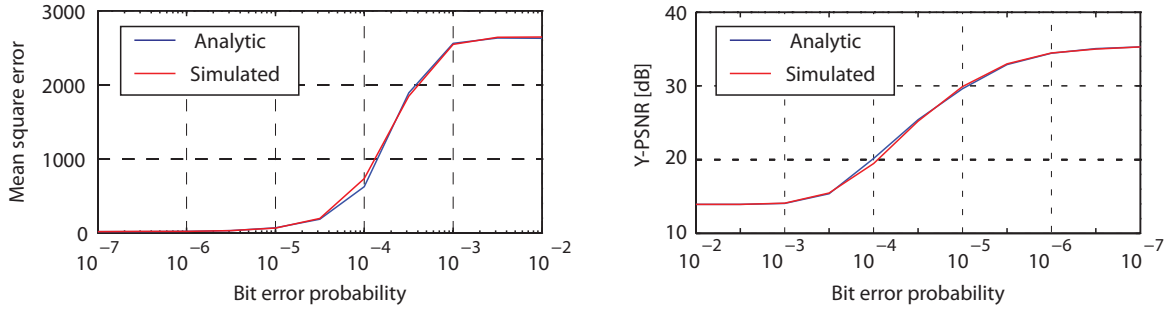


Figure 3.10: Average MSE after transmission (QP=28).

3.3.4 Detection Performance

As discussed before, the detection probability as well as the detection distance represent key issues for the performance of the proposed method. Since the desynchronized decoding remains limited to the single `NALU`, new simulations were performed introducing a single error each slice. It should be therefore underlined that the focus of these simulations is given on the detection performances and not on the quality, that would also be dependent on the temporal propagation of the resulting errors.

The simulations were performed on a sequence encoded with quantization parameter $QP = 28$ and `NALU` size limited to 700 bytes. These encoding parameters lead the I frames to be segmented in more than four slices, whereas the P frames usually consist of a single `NALU`.

The probability of detecting an error in the I frames is around 60%, whereas for the P frames it is around 47%. This difference can be explained considering the information elements encoded in the two different encoded slices. The I frames are self contained: the encoded information is sufficient to reconstruct the picture without referencing any other previous decoded picture. This results in a less effective prediction and in a set of coefficients with higher information content. Such coefficients are much more sensitive to bit inversions and to desynchronized decoding. The packets containing encoded P frames, on the contrary, contain the information for reconstructing the picture applying motion compensation to the previously decoded pictures. Most of the information is contained in the motion vectors, describing the position of the best prediction of the considered block in the reference picture. Errors affecting the motion vectors are hardly detectable and would cause the selection of a wrong prediction block.

Figure 3.11 shows the detection distance, expressed in number of macroblocks, for the two types of predicted frames. Both normalized histograms have an exponential trend. For the I frames, 90% of the detected errors in the I frames are detected within 2 MBs. The range varies between 0 and 20 MBs, as for the considered encoding settings, an I slice contains up to 25 MBs. The exponential trend has the form

$$P_{I,SC}(k_d - k_e) = 1.552 \cdot \exp(-1.508 \cdot (k_d - k_e)). \quad (3.7)$$

The fit has been obtained by means of non-linear least square fit.

For the P frames, this detection increases and, in average, the detection occurs within 7 MBs. The detection distance for the P frames is higher because of a specific element encoded in the P frames, namely the `mb_skip_run`. It signalizes how many macroblocks have to be skipped, that means how

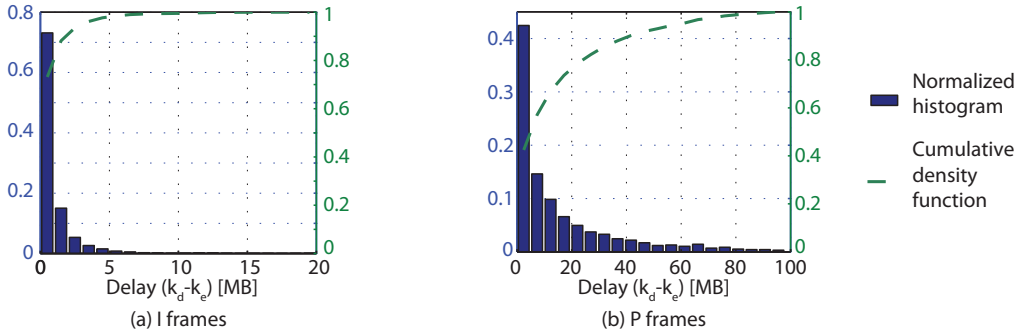


Figure 3.11: Detected errors: distance between error occurrence and error detection, $k_d - k_e$.

many macroblocks can be reconstructed without further encoded information. Errors affecting this parameter will modify the number of skipped macroblocks. Since, in most of the cases, the encoded value is zero, an error will increase the number of the skipped macroblocks and, therefore, the detection distance. The model that has been found to best approximate the normalized histogram has the form

$$P_{P,SC}(k_d - k_e) = 0.757 \cdot \exp(-0.386 \cdot (k_d - k_e)) + 0.146 \cdot \exp(-0.046 \cdot (k_d - k_e)). \quad (3.8)$$

For the errors that have not been detected by the syntax analysis, a similar investigation has been performed. In this case, the distance between the error occurrence and the end of the slice has been measured. The histogram of the distribution is plotted in Figure 3.12. For the I frames, 50% of the

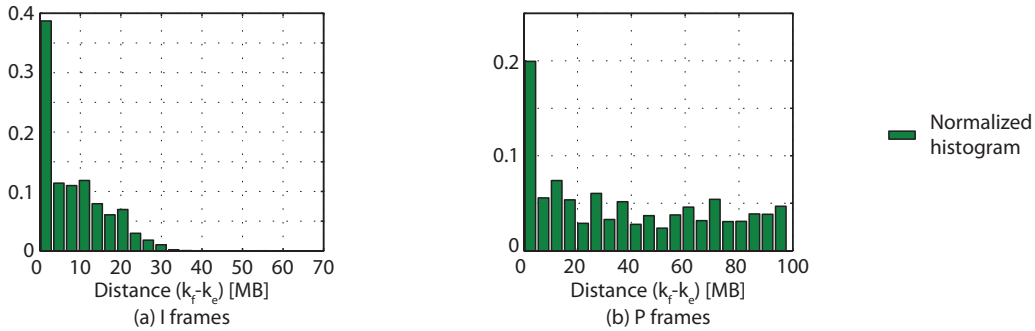


Figure 3.12: Undetected errors: distance between error appearance and end of slice, $k_f - k_e$.

undetected errors are located in the last two macroblocks of the slice. Also for the P frames a peak is recognized for short distances between the error occurrence and the end of the slice. This shows how a considerable number of errors cannot be detected because the number of macroblocks to be decoded after the error occurrence is smaller than the average detection distance.

In order to evaluate the influence of the undetected errors, their impact in terms of objective distortion has been measured. In Figure 3.13, the average distortion introduced by the missed detection is drawn as a function of the distance between the error occurrence and the end of the slice.

The distortion is measured as the **MSE** between the macroblocks affected by decoding desynchronization and the same macroblocks reconstructed in an error free environment. Small **MSE** values signalize that, even if the error has not been detected, the effects did not impair the decoded picture. Since the **MSE** is calculated over the whole area possibly affected by decoding desynchronization, One

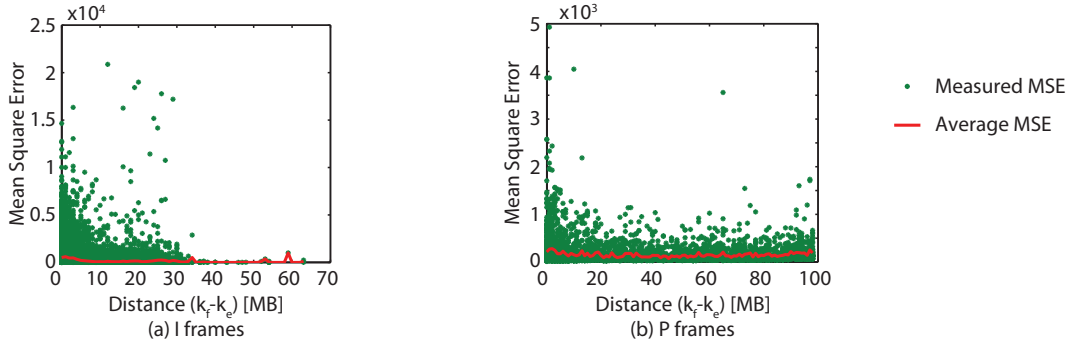


Figure 3.13: Undetected errors: MSE calculated in the range $k_f - k_e$.



Figure 3.14: Visual impairments caused by undetected errors.

might expect increasing MSE values for increasing detection distances. However, the performed simulations demonstrate that the resulting MSE does not depend on the number of affected macroblocks. This leads to the conclusion that, in average, the undetected errors do not affect significantly the decoded picture.

The previous reasoning is not valid for small values of $k_f - k_e$ that, in average, is the highest MSE value. As the MBLC suffers from a detection distance, these errors were not detected for being too close to the end of the slice. In other words, these undetected errors cause harmful artifacts, resulting in increasing MSE, and would be detected by MBLC if enough macroblocks were following.

3.4 Visual Artefacts Detection

As shown in Section 3.3.4, the syntax analysis still suffers of a detection distance. In the region between error occurrence and error detection, the encoded information is incorrectly decoded. This results in visual impairments in the pixel domain. The detection of such visual artefacts can further improve the performance of the syntax analysis, reducing the detection distance. However, the detection of impairments in the video domain calls for refined preprocessing techniques at pixel level.

An analysis over video sequences obtained by decoding corrupted bitstreams has been performed and is described in the following. It is worth noticing that the effect of desynchronization was significantly different depending on the type of frame, Inter or Intra.

3.4.1 Inter Frames Impairments

The Inter encoded frames exploit the temporal correlation between consecutive frames. The encoded picture is reconstructed applying motion compensation to its macroblocks, or its sub-macroblocks. The differences between the current image and the available reference pictures are compensated by means of transformed quantized residuals. However, there is only a small dependency between the encoded information belonging to consecutive macroblocks of the same picture slice. This means that, in case of error, there is not a significant spatial propagation of artefacts due to the wrong reconstruction of a macroblock. However, the decoding desynchronization still remains a major drawback. As introduced in Section 3.3.4, the desynchronization of the decoding in the inter frames occurs more rarely than the intra frames. Errors in the motion vectors, moreover, can rarely be detected by syntax check. The field `mb_skip_run` signalizes the length of the run of skipped macroblocks, and it is encoded by means of exp-Golomb coding, as described in Section A-3. Since, usually, the macroblock is not skipped, the associated codeword is 1, indicating a zero run length. In case of desynchronization, the `mb_skip_run` can turn to 0, this would be the first bit of the prefix of a value bigger than zero. In case of decoding desynchronization this causes the overestimation of the number of macroblocks to be skipped. As a result, the impairments result to be isolated and spatially interleaved between skipped macroblocks.

When detecting errors in the frame n , we assume the frame $n - 1$ to be correct. To detect errors in P frames, we analyze the pixel-wise difference map $D_n(i, j)$ between frame n and frame $n - 1$:

$$D_n(i, j) = |F_n(i, j) - F_{n-1}(i, j)|. \quad (3.9)$$

Since we aim to detect of artefacts with the resolution of a macroblock, the difference map D_n is then reshaped considering the average difference in the 16×16 pixels. The difference map is not only dependent on possible visual artefacts, but also on the movement between the two consecutive pictures. The artefacts represent isolated, out of context, square regions of pixels. We therefore propose to implement a simple edge detector to highlight edginess in the picture. It has been noticed that observing the edge characteristics of 8×8 pixels sub-macroblocks represents the best compromise between false and missed detections.

The final decision whether one block k is detected as erroneous or not, is then taken considering the information about the difference map $D_n(k)$, defined as

$$D_n(k) = \sum_{(i,j) \in k} |D(i, j)|, \quad (3.10)$$

as well as the edginess map $E_n(k)$, defined as

$$E_n(k) = \frac{1}{8} \cdot \sum_{l=0}^7 |F_n(i_1, j_1 + l) - F_n(i_1 - 1, j_1 + l)| + \quad (3.11)$$

$$\frac{1}{8} \cdot \sum_{l=0}^7 |F_n(i_1 + l, j_1) - F_n(i_1 + l, j_1 - 1)| + \quad (3.12)$$

$$\frac{1}{8} \cdot \sum_{l=0}^7 |F_n(i_1 + 7, j_1 + l) - F_n(i_1 + 8, j_1 + l)| + \quad (3.13)$$

$$\frac{1}{8} \cdot \sum_{l=0}^7 |F_n(i_1 + l, j_1 + 7) - F_n(i_1 + l, j_1 + 8)|. \quad (3.14)$$

$$(3.15)$$

For each block, $E_n(k)$ represents the average difference between the rows (resp. column) at the border of the macroblock and its neighbors belonging to a surrounding macroblock. They are both compared with an adaptive threshold considering the movement characteristic of the whole picture.

3.4.2 Inter Frames Impairments

In the Intra predicted frames the correlation between neighboring macroblocks belonging to the same picture is exploited. Each block is predicted considering the luminance and chrominance component of the confining already encoded blocks (see Section A-2.1). At the decoder side, the quality of a single block strongly depends on the correctness of the neighbors. The spatial propagation of the errors in the I frames may occur also in case no desynchronization in decoding takes place. As an example, assume the codeword of a VLC residual to be affected by a bit inversion in the info field; therefore not causing decoding desynchronization. The considered block will be reconstructed differently from the block available at the encoder. The following blocks exploiting that macroblock as a reference will suffer of a spatial error propagation, since the obtained prediction will not be consistent with that considered at the encoder side.

Therefore, as shown in Figure 3.14(a), an error in the I frames usually corrupts the affected macroblocks as well as its successors until the end of the slice. The different behaviors observed in the two kinds of frame call for the design of two different detection mechanisms.

To ensure robustness to the detection mechanism, the decision is taken considering a voting system. Similarly to the detection performed in the inter frames, the input to the voting system are the block difference map $D_n(k)$ and the edge map $E_n(k)$. Since the detection performance of the syntax analysis in the intra frames was significantly better, the error position as detected by the syntax analysis is considered as well.

A scoring procedure is initialized each time the difference and edginess value of a block surpass a second threshold. This block is considered as the root of the artefacts sequence. The following blocks are further investigated: in case their difference and edginess characteristics are compatible with the artefact's features, the score is increased, otherwise decreased. A sequence of possible artefacts is terminated in the following cases:

1. The score of the sequence surpasses a given threshold. In that case, the root of the sequence is assumed to be the macroblock where the error occurred. The following macroblocks until the end of the slice are marked as corrupted.
2. The score of the sequence remains below a given threshold. In that case, the sequence is handled as a false positive. The detection is restarted and a new possible root is searched for in the following macroblocks.
3. The syntax analysis signalizes that an error was detected in the current macroblock. The root of the sequence is considered as the macroblock where the error occurred. The following macroblocks until the end of the slice are marked as corrupted.

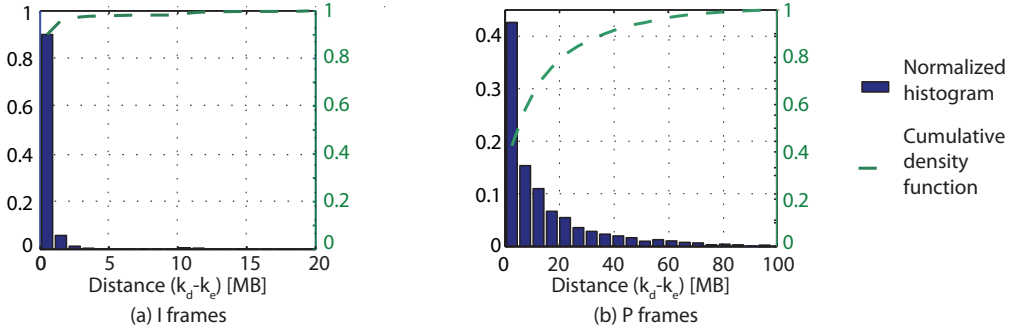


Figure 3.15: Errors detected by visual impairments detection: $k_d - k_e$.

4. The end of the slice has been reached, and none of the previous condition was fulfilled. In this case the current score is compared with a second threshold. In case the current score exceeds it, the root is considered as the macroblock affected by the error. Otherwise, the whole sequence is treated as a false positive.

All the considered thresholds, as discussed for the intra predicted frames, are adaptive and depend on the movement characteristic of the sequence.

3.4.3 Results

In order to measure the detection performance, the same simulation setup as described in Section 3.3.4 was considered. The results in terms of detection distance are shown in Figure 3.15.

For Inter encoded frames, the syntax analysis suffered of a significant detection distance. The detected errors were spotted, in average, after more than seven macroblocks. Performing the detection of visual impairments, such distance is slightly reduced to 6.8 MBs. The average motion compensation measured in the neighboring macroblocks is applied to those that are skipped, therefore they do not result as out of context macroblocks. The probability of detecting an error in a P frame by means of visual detection, $P_{P,VD}$, can be described as

$$P_{P,VD}(k_d - k_e) = 0.788 \cdot \exp(-0.456 \cdot (k_d - k_e)) + 0.197 \cdot \exp(-0.057 \cdot (k_d - k_e)). \quad (3.16)$$

Even though the performance of the syntax analysis for Intra encoded frames was satisfactory, by means of visual artefact detection it has been further improved. The average detection distance, in particular, has been reduced from 1.39 MBs to 0.92 MBs. Also the normalized histogram of the detection distance for I frames follows an exponential trend:

$$P_{I,VD}(k_d - k_e) = 3.499 \cdot \exp(-2.714 \cdot (k_d - k_e)). \quad (3.17)$$

Also the average detection probability for I frames has been increased from 54.35%, with the MBLC, to 59.99% by means of visual error detection. Although the overall detection probability does not exceed 60%, it has to be noted that, usually, the errors that do not cause desynchronization, do not produce any visible artefacts on the decoded picture. Also in the pixel domain, an error in

the trailing ones can be barely spotted, since it would influence only high frequency components. Moreover, it remains questionable whether the detection of such errors would influence positively the resulting quality. The following macroblocks, in fact, can be correctly decoded and possible drifts in the spatial prediction would result in negligible distortion. Marking these macroblocks as corrupted would cause the concealment not to exploit the available valid information.

3.5 Watermarking

In order to detect errors in the corrupted stream, additional data can be associated to the stream. In this section, the application of watermarking as an error detection mechanism is presented. Digital watermarking consists on embedding information into the multimedia content. In general the watermarking mechanisms are subdivided into two classes: visible and invisible watermarking.

In case watermarking is appreciable in the video or picture by the users, it is defined visible. Examples of visible watermarking are the logo of the TV broadcaster as well as the text or image signalizing the owner of the multimedia content.

Watermarking can also be invisible, if the embedded information is invisible to the users and can be only interpreted by an appropriate data decoder. Invisible watermarking is mostly employed by copyright protection systems for avoiding forbidden copying or modification of the content. For these applications, the invisible watermarking is a data pattern known both at the transmitter and at the receiver (secure parts), but unknown to the nodes in the middle (insecure part), as shown in Figure 3.16. If the content is modified in the insecure part, the retrieval function detects the corruption of the pattern and informs the decoder, that can react accordingly.



Figure 3.16: Transmission scheme with watermarking.

An application of digital watermarking as an error detection mechanism has been presented in [58]. There, Chen et al. presented a watermarking scheme for detecting errors in the encoded stream, by forcing a condition on the H.263 coefficients. Chen et al. claim to have improved the results of the syntax check analysis mechanism published in [11] by means of an Discrete Cosine Transformation (DCT) coefficients watermarking scheme, depicted in Figure 3.17. H.263 makes use of the same

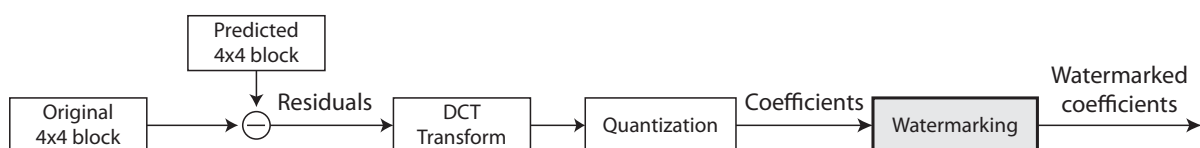


Figure 3.17: Watermarking of DCT coefficients.

conceptual encoding mechanisms as those described in Section A-2 for H.264/*AVC*. On this extent, the most noteworthy difference is the usage of residual blocks of 8×8 pixels. The following operations, *DCT* transformation and quantization are also applied on 8×8 blocks. The coefficients are watermarked by forcing the coefficients to be even, the method thus being called Force Even Watermarking (*FEW*). As described for H.264/*AVC*, the coefficients are zig-zag scanned and, out of the 64 coefficients, only the ones belonging to the set $[pos, 63]$ are watermarked, where $0 \leq pos \leq 63$. The inserted watermarking is *fragile* as any uncorrected channel error corrupts the embedded information. At the decoder side, the coefficients in the position $[pos, 63]$ are checked and, if any odd coefficient is present, the block is marked as erroneous.

The number of watermarked coefficients strongly influences the performance of the scheme. Watermarking introduces distortion before the transmission takes place, since watermarking modifies the original coefficients, that guaranteed the better quality. At the receiver side, small *pos* values increase the detection probability, therefore increase the quality of the decoded video after transmission. The choice of the best *pos* value depends on the trade off between distortion at the encoder and detection capability at the receiver side.

The presented results showed major improvements in terms of detection probability as well as quality at the decoder side if compared with the syntax analysis alone. However, Chen et al. considered only the case of errors inserted in the coefficients:

In order to focus on the detection capability for erroneous quantized DCT coefficient, we only cast bit errors on those coded bits that represent quantized DCT coefficients.

In [59], Nemethova et al. proposed an extension of the work for an H.264/*AVC* codec. In H.264/*AVC* the *DCT* is performed over 4×4 pixels blocks, the value *pos* ranges between 0 and 15. Besides the implementation of *FEW* for the H.264/*AVC*, Nemethova et al. introduced Relation Based Watermarking (*RBW*). *RBW* consists on forcing the value of some specific coefficients in order to fulfill a specific condition, such as the modulo function. The promising results obtained by the authors were still obtained under the unrealistic condition of errors affecting only the coefficients. This assumption is particularly not applicable for high *QP* values where the coefficients represent only a small fraction of the code.

The watermarking mechanisms presented in this doctoral thesis have been tested in a more realistic scenario, considering errors affecting the whole stream. For this reason, these methods have been implemented as an enhanced feature of the modified *JM* codec (see Section 3.3) capable of handling errors in the bitstream. It has been investigated whether the results obtained in [59] are still valid in case the assumption of errors affecting *only* the coefficients is not in force anymore.

When distributing the errors in the whole stream, the performance of the *FEW* decreased considerably. This effect is shown in Figure 3.18(a). The additional detection capabilities of watermarking are not able to compensate the distortion introduced at the encoder side. Slight improvements can be observed for one specific case, that is *BER* equal to 10^{-5} , low quantization parameter (24) and *p* equal to 14. The decrease in performance can be justified as follows:

- With increasing *QP*, more than 90 % of the coefficients are trailing ones. When applying *FEW*, all of them are turned to zero.

- Exp-Golomb encoded codewords are sensitive to bit inversions (see Section 3.2). For the performance of watermarking, two parameters suffer particularly this effect: `mb_skip_run` and `coded_block_pattern`.

1. `mb_skip_run` signals how many consecutive macroblocks are encoded using the *skip mode*. The default value (codeword 0), is assigned to non skipped macroblocks. If this value is misinterpreted, a succession of macroblocks is skipped increasing the detection delay.
2. `coded_block_pattern` signals whether one encoded macroblock contains coefficients (codeword 001XX) or it does not (1). In the considered QP ranges, most of the macroblocks are encoded with coefficients. If the word is misinterpreted and the first bit turns to a '1', no coefficients are associated to the macroblock. As the watermarking pattern has been forced in the coefficients, in this case no detection can be performed.

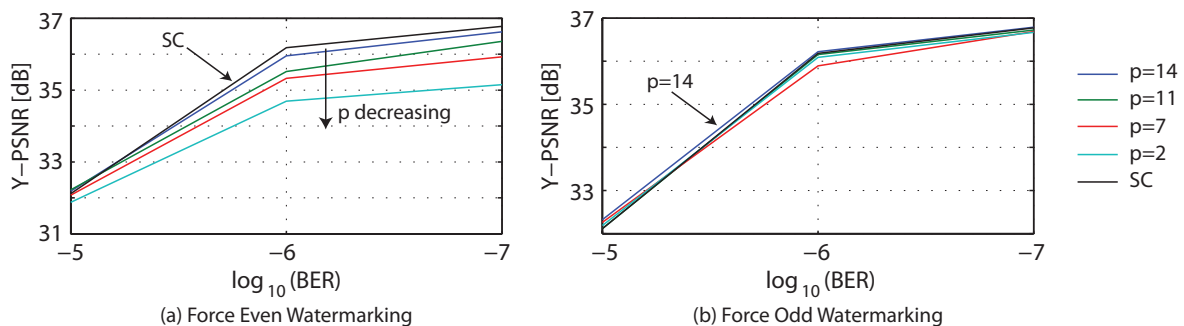


Figure 3.18: Comparison between FEW and FOW.

One of the **FEW** drawbacks is the modification of the trailing ones to zero. This reduces the quality at the encoder side, as the motion compensation is not improved by means of coefficients correction. For this reason, an equivalent approach named Force Odd Watermarking (**FOW**) has been implemented. As the name suggests, the encoded sequence is watermarked by turning all the even coefficients to odd ones. As a result, the trailing ones have not been modified. At the same time, however, as less coefficients have been watermarked, the detection capabilities have been reduced. The results obtained with **FOW** have been depicted in Figure 3.18(b). As the decrease in detection capabilities is partially compensated by a smaller introduced distortion, no particular relationship has been recognized between the value of P and the distortion measured at the decoder.

In the approach proposed in [59], the sixteen 4×4 subblocks belonging to a single macroblock have been independently watermarked. As the concealment method is performed on a macroblock basis, in this thesis a single watermarking condition has been set in force for the whole macroblock². The sixteen coefficient sets associated to each macroblock have been analyzed and a condition has been forced on three trailing ones for each macroblock. As the trailing ones in high frequency have smaller impact on the objective quality than those in low frequency, the three watermarked coefficients are those in the three highest frequency positions.

The improvements in terms of error detection probability and detection distance have been measured and are shown in Figure 3.19. The three curves are related to the performance of one of the two

²Rama Rao Ganji has to be thanked for the implementation of the method

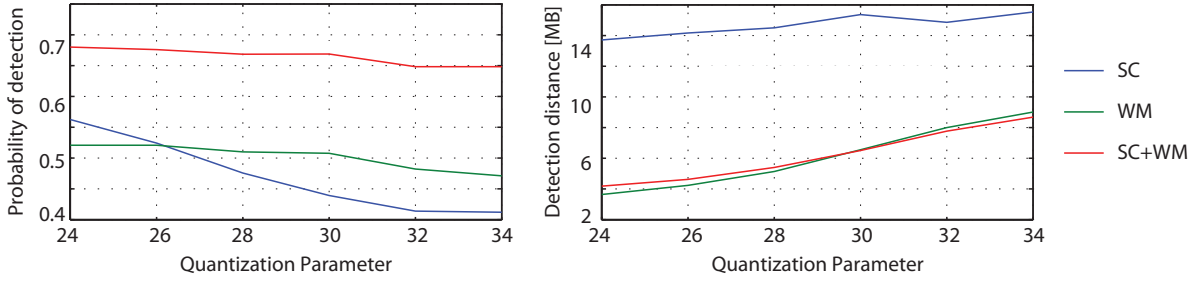


Figure 3.19: Error detection probability and detection distance of watermarking with three coefficients for each MB.

methods (SC and WM) enabled and the other disabled as well as the combination of both methods enabled. The detection probability of both strategies separately depends on the selected QP and lies around 50%. The detection probability of the two methods combined has been increased to 75%. In terms of detection distance, this has been significantly reduced by means of watermarking. Even though depending on the QP, the detection distance measured when enabling WM only is almost 3 MB shorter than the distance measured when only SC is enabled. When both methods are enabled, the measured detection distance stays close to the one obtained with WM enabled.

As a last result, the quality at the receiver side has been measured, see Figure 3.20. Surprisingly,

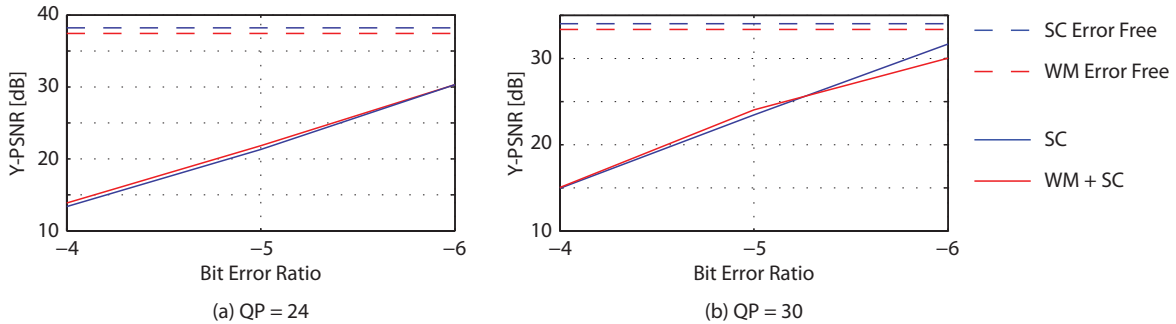


Figure 3.20: Quality of watermarking with three coefficients for each MB.

the considerably improved detection distance and detection distance does not reflect in significant quality improvement. With increasing quantization parameter, the advantages of using watermarking decrease. In Figure 3.20(a) is shown the case of a small QP equal to 24. The watermarking is able to compensate the degradation introduced at the encoder side and provide quality improvements if compared to the syntax check alone. When considering an higher quantization parameter like 30, Figure 3.20(b), the quality measured when using the watermarking remains below the quality obtained by applying syntax check to the original stream.

A more detailed observation of these results showed how most of the errors detected by watermarking but not detected by syntax check were not harming the quality at the receiver side (see Figure 3.13). As soon as one error is detected, all the following macroblocks are not decoded but rather concealed. If watermarking detects an error that was not causing quality degradation, the concealment of the following correctly decodable macroblocks is actually decreasing the overall quality. As shown

in Figure 3.19, for smaller QPs the error detection performed by SC remains dominant over WM. In this case, WM is often reducing the detection distance of the same errors also detected by SC, improving the overall performance. When, for larger QPs, WM becomes dominant over SC, several errors detected by WM would have not been detected by SC.

Specifically, errors in the sign of the trailing ones are not harmful for the reconstructed picture nor cause decoding desynchronization. Such errors are not detected by syntax check but make the watermarking conceal the following unharmed macroblocks. Summarizing, watermarking offers advantages in a combined scheme when it is used for reducing the detection distance of syntax check (small QPs). However, it has to be used carefully as a tool for detecting errors, as it may cause unnecessary concealment.

3.6 Smart Sorting

An error resilience features introduced in H.264/AVC but not supported by the baseline profile is Data Partitioning (DP) [60]. By means of DP the encoded stream is subdivided into different partitions and each syntax elements is associated to the appropriate partition. Each I frame is subdivided into two partitions (A and B), whereas each P frame is subdivided into three partitions (A, B and C). In this approach, we focus on the partitions of the P frames. Partition A contains control information such as the macroblock subdivision and the motion information (motion vectors). Partition B contains the intra related encoded coefficients whereas partition C contains the inter related ones.

By means of data partitioning, Unequal Error Protection (UEP) mechanisms are allowed. By means of UEP packets containing the most important information can be protected better. In case of DP, the data contained in the partition A allows a first reconstruction of the picture by means of motion compensation, therefore it has to be better protected. The remaining partitions are used for refining the prediction by means of the encoded residuals.

However, this functionality is not permitted in the baseline profile of the H.264/AVC. As a workaround, in this thesis it has been proposed to exploit the idea behind data partitioning for sorting differently the information encoded in the stream and better protect the most important elements. In Section 3.2 it has been shown how even a single bit inversion can corrupt the encoded information stored after the error occurrence. For this reason it has been proposed to store the most important information, equivalent to the partition A, at the beginning of the NALU followed by the remaining information, the partition B.

Because of the decoding desynchronization, the probability of a given bit b_n to be incorrectly interpreted due to previous occurred errors, $P_b(b_n)$, is equal to the cumulative density function of the bit error probability $p_b(b)$ calculated from the first bit of the NALU b_0 until the bit b_{n-1} :

$$P_b(b_n) = 1 - (1 - p_b(b_i))^n, \quad (3.18)$$

where $p_b(b_i)$ is the probability that an error occurs in bit b_i . The most important elements are, therefore, stored at the beginning of the NALU, where the probability of the data to be corrupted is smaller. This is shown in Figure 3.21. As it will be discussed in the conclusions (Section 3.7) the assumption of uncorrelated error probability is, for wireless transmission, not as realistic as for other

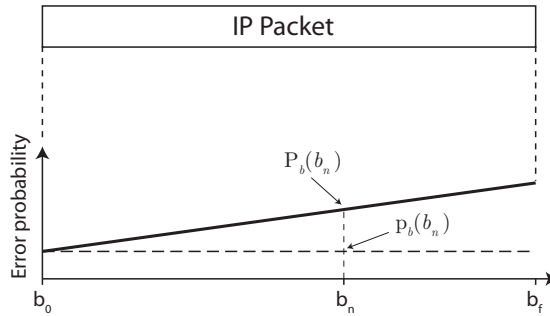


Figure 3.21: Error probability at bit level.

transmission schemes. An IP packet consists of the recollection of several transport blocks. Also assuming that a transport block can be either correct or damaged, therefore considering a TB error probability rather than a bit error probability, the data stored at the beginning of the NALU is still better protected. The method is implemented as shown in Figure 3.22. The transmitter side consists

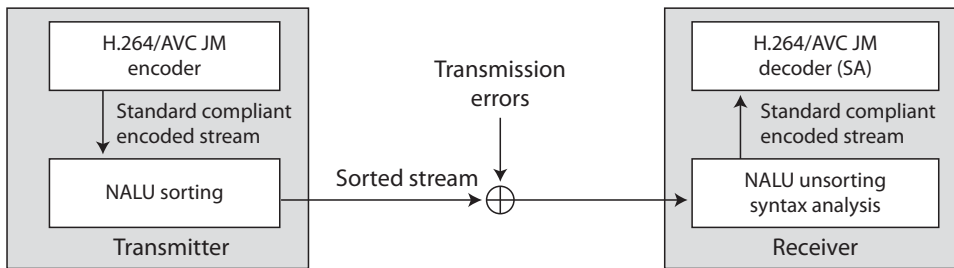


Figure 3.22: Implementation of smart NALU sorting.

of a standard H.264/AVC encoder based on the JM followed by a MATLAB script responsible of the slice sorting. According to the standard, a NALU consists of the Slice Header (SH) followed by the encoded information belonging to each macroblock. The motion information associated to the last macroblock belonging to the NALU are, therefore, stored close to the end of the packet, where the probability of the data to be corrupted is higher.

The encoded elements have *intra macroblock* dependency as well as *inter macroblock* dependency³. Almost all the encoded elements of both I and P possess *intra macroblock* dependencies, the understanding of a codeword depends on the values of the already decoded codewords belonging to the same macroblock. For example, the number of the decoded motion vectors depends on the codeword containing the number of macroblock subdivisions. The encoded elements, however, are interpreted also depending on the value of the decoded codewords belonging to previously decoded macroblocks. For instance, the codeword indicating the number of non-zero coefficients and trailing ones is interpreted depending on the number of non-zero coefficients in the neighboring macroblocks. The inter macroblock dependency is much stronger in the I frame than in the P frames. In the P frame, the only elements (besides the VLC levels) that have inter macroblock dependency are the motion vectors. The codeword associated to the motion vector, in fact, does not represent the absolute value of the

³The intra and inter macroblock dependencies have not to be confused with the intra and inter prediction types

motion vectors, but rather their relative amplitude if compared with the prediction built with respect of the neighboring macroblocks. This method has been implemented and tested for the Inter encoded macroblocks.

For sake of simplicity, the MATLAB script is supported by the trace file output by the JM reference software. In the trace file, each codeword is associated to its meaning and value in natural language. This helps the separation of the encoded stream into the different parameter classes. The sorting process is shown in Figure 3.23. The original stream, consisting of more than 15 different parameters,

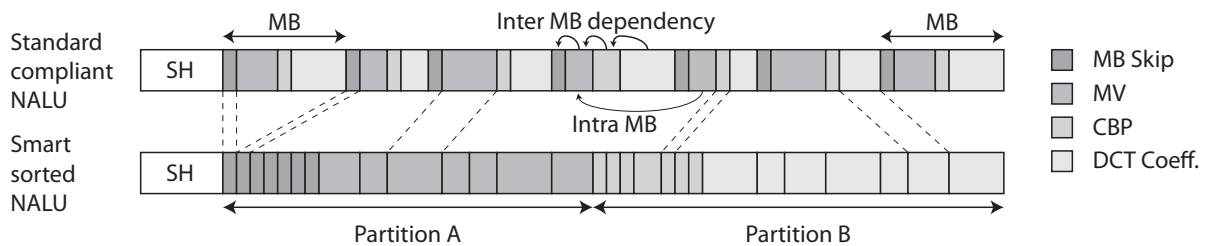


Figure 3.23: Sorting mechanism.

is classified into four groups. The first group contains the macroblock skip parameter, indicating whether a macroblock has been skipped or not. The second parameter block contains the macroblock subdivision as well as the motion vector information. These two first blocks represent the pseudo partition A. The third block contains the macroblocks Coded Block Pattern (CBP), the fourth the luma and chroma DCT coefficients.

At the receiver side, a MATLAB script is in charge of recollecting the parameters sorted at the transmitter side and reconstruct a standard compliant stream decodable by the H.264/AVC decoder. The script at the decoder side is much more complex than the one at the encoder side. As no trace file can support the stream recollecting, the script has to interpret the encoded codewords and offer basic syntax analysis functionalities. In detail, the script implements all the decoder functionalities except for the pixel level macroblock reconstruction. The error classes detected at this step are OR and IC.

As soon as an error is decoded, the script reconstruct the code modifying the corrupted part of the stream. If an error is detected when decoding the *pseudo* partition A, the current macroblock and its successors are set to the skip mode. If an error is detected during the decoding of the *pseudo* partition B, the coded block pattern of the current and of the following macroblocks is set to zero. In this way the corrupted coefficient information is excluded from the reconstructed stream.

The transmission is then simulated by means of a Binary Symmetric Channel (BSC) model. The bit inversion has been set in all possible positions of the stream and the resulting quality at the decoder side has been measured and compared to the results obtained with the syntax analysis proposed in Section 3.3. For this experiment, the size of each slice has been set equal to 650 bytes (5 200 bits). One slice contains, in average, 73.72 MB. The quantization parameter has been set to 26. The quality of the decoded picture when detecting errors in different positions has been measured and drawn in Figure 3.24.

The syntax analysis results are consistent with the intuitive observations. The quality is initially increasing linearly with the position of the error detection. However, as the correctly decoded mac-

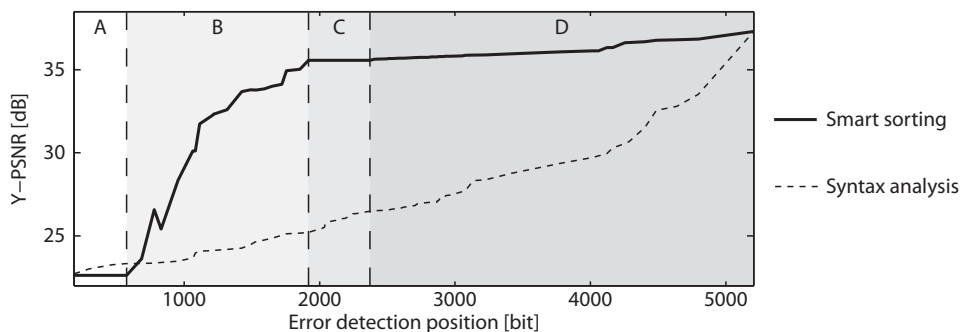


Figure 3.24: Smart sorting results.

roblocks serve as refinement for the concealment, the more macroblocks have been correctly decoded, the higher is the quality of the concealed region.

For discussing the results of the smart sorting algorithm, the results have been subdivided into four regions. The first comprises the macroblock skip, type and possible macroblock subdivisions of all the macroblocks stored into the packet. The data stored in this region does not contribute to improve the quality of the reconstructed frame. As all the other parameters are missing, knowing whether the macroblock has been skipped or how it is segmented does not provide any useful information for enhancing the concealment performance. In average, the region A corresponds to 6.17 MBs stored with the standard mechanism. If correctly decoded, they would positively contribute to increase the quality. For this reason, in region A the syntax analysis is working better than the smart sorting.

In region B the motion vectors applied to the regions defined in region A are stored. The motion vectors contribute consistently to the reconstruction of the picture. As soon as an error is detected, the already decoded motion vectors are exploited for improving the performance of the concealment. If all the motion vectors have been correctly decoded, the quality is already 86.6% of the error free case. The third region, C, contains the CBP and the macroblock's QP delta. Following the same observations made for the region A, these information elements do not help improving the picture reconstruction. The last region, D, contains the luminance and chrominance coefficients. Even though they represent almost 65% of the code size, their impact on the quality at the receiver side is not as significant as the motion vectors.

3.7 Conclusions and Self Criticism

Different error detection mechanisms have been presented in this chapter. The basic assumption behind these investigations is the necessity of exploiting the fraction of the payload that still contain valid information. Mobile operators keep the IP packet size as close as possible to the network's MTU. In case the entire damaged packet is discarded, the decoder has to cope with a significant amount of missing information. As the effectiveness of the concealment strongly depends on the size of the missing region, it is of major importance to keep this region as small as possible.

Error detection mechanisms allow the decoder to recognize the damaged payload's region. Subsequently the data stored before the error occurrence can be exploited. The effectiveness of the methods

has been measured considering two metrics: (i) error detection probability and (ii) detection distance. The first index measures the amount of detected errors over the total error occurrences. The second index expresses, in amount of macroblocks, the distance between the error occurrence and the error detection.

The error detection probability is the most immediate metric. However, it does not discriminate between errors that harm the perceived quality and those that does not cause any visible artefact. The error detection probability has, therefore, to be considered together with the distortion measured when no error is detected. In case the undetected errors do not lead to visible artefacts, the detection probability *per se* does not offer a proper effectiveness metric. The error detection distance takes into account how many macroblocks, following the error occurrence, are decoded before the error is detected. Following the previous discussion, only the detected errors are taken into account. As the region decoded between the error occurrence and error detection is incorrectly reconstructed, this distance has to be kept as small as possible.

Three error detection mechanisms have been compared. The first method, the syntax check, is a low complexity method directly implemented in the H.264/**AVC** decoder as exception handling application. As soon as some context-dependent forbidden codewords arise, the currently decoded macroblock as well as the following ones are assumed to be damaged. This method has a moderate error detection probability (below 50 %) and detection distance of 1.39 and 15.09 MBs for I and P frames, respectively. On top of that, two methods have been implemented: the visual artefacts detection and the invisible watermarking.

The former consists of a postprocessing tool analyzing the reconstructed picture. Specific patterns, such as blockiness or inconsistency, are searched for and, by means of a voting system, possible errors are detected. The improved performance is paid in terms of increased complexity, as the picture has to be analyzed in the pixel domain. The amount of the additional complexity is hardly quantifiable, as the method has been implemented in American National Standards Institute (**ANSI**) C as an optional feature of the standard decoder.

While the syntax check and the visual artefact detection are solely implemented at the decoder, watermarking calls for modifications at the encoder side as well. The encoder forces the value or the sign of specific coefficients to be consistent with a specific pattern. As the pattern is also known at the decoder, transmission errors are recognized by looking for broken patterns. Even though the performance of the method overcomes both the syntax check alone and its combination with the artefact's visual detection, the embedding of the pattern causes a quality degradation before transmission at the encoder side. In terms of quality comparison, therefore, the watermarking has to overcome the native quality degradation that is not present in the other methods, as they have been implemented for standard encoded sequences.

A final remark is reserved to the channel model that has been employed for obtaining the results. As in most of the works in literature, the **BSC** channel model has been considered. Although widely accepted for different scenarios, the **BSC** does not represent a reliable channel model for mobile communications. The **BSC** works under the assumption of randomly distributed errors, the error occurrences in **UMTS** are highly correlated and bursty, as explained in Section 5.1.1. Karner [23] showed how a model assuming uncorrelated transport error distribution cannot properly describe the

wireless link of the UMTS DCH.

If the correctness of the transport blocks can be evaluated observing the CRC information, this information can be exploited for reducing the area where the error has to be searched for. However, as it will be explained in Chapter 5, the information stored into each transport block is scrambled twice by means of interleavers before being transmitted through the physical channel. This means that, even though the a transmission error affects just a part of the physical block, the error is spread over the whole transport block by the interleavers. Moreover, it is more reasonable to assume that a transmission error affects several bits rather than one single bit. It has been showed before that almost half of the occurring errors, the detected ones, are harmful and cause artifacts. Assuming several errors within a single TB, it is almost sure that the data will be incorrectly reconstructed, as one or more errors are harmful. The most reasonable decision is to start the concealment as soon as one transport block fails the CRC checksum test.

The presented method finds application in more reliable transmission technology, such as Digital Subscriber Line (DSL). For wireless communications, therefore, error detection mechanisms considering information from the lower layers [61] are more meaningful. In [62], the visual error detection performance have been improved considering the correctness of the transport block as additional information. This limits both the computational complexity and increases the robustness of the method against false positive. Joint source-channel optimization methods based on the correctness of the transport blocks are described in Chapter 5.

The last presented method describes an alternative sorting of the payload’s data. The discussion regarding the error detection mechanism demonstrated how the data following the error occurrence cannot be correctly decoded because of decoding desynchronization. The data stored at the end of the packet has the highest probability of being incorrectly interpreted. By means of a different arrangement, the most important payload has been moved to the beginning of the packet before transmission. At the receiver side, the standard compliant sorting is reconstructed. As soon as an error is detected, the remaining data is ignored and the reconstruction is interrupted. Only the valid data is considered. This last approach offers advantages also when considering the more realistic transport block error model mentioned before. Figure 3.25 shows the case of a fixed transport block error

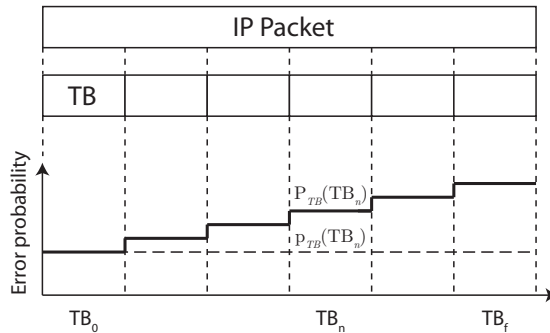


Figure 3.25: Error probability at transport block level.

probability $p_{TB}(TB)$. The data contained in the transport block TB_n cannot be correctly decoded if one of the previous transport blocks $TB_i, i \in [0, n)$ was incorrectly received. The probability of one transport block not to be correctly decoded, $P_{TB}(TB)$, linearly increases with the resolution of

one transport block. The data stored near the beginning of the packet remain better protected from decoding desynchronization also considering an error model based on transport blocks.

Chapter 4

Optimization of Soccer Video Streaming

SOCCER video is one of the most appreciated content streamed over 3G wireless networks. During some specific events, such as the World or European soccer championship, there is a *hype*, as many people on the move are interested on getting the content even without having a TV set available.

However, the quality as perceived by the customers is, in most of the cases, not satisfactory. Even though most of the user satisfaction is a matter of their expectations, when following a soccer match the visibility of some specific entities drives the user's appreciation.

This chapter describes a preprocessing algorithm and an encoding optimization scheme aiming at the enhancement of the quality as appreciated by the users. In both approaches, emphasis is given on the subjective response to the improved video sequences, with the quality being measured by means of subjective tests, such as MOS, rather than the objective Y-PSNR.

The first method to be presented is based on an idea from Nemethova et al. [63]. As the ball tends to disappear because of the encoding, the authors proposed a preprocessing algorithm for making the ball resistant to the smoothing introduced by the video encoder. The encoding optimization scheme is a genuine contribution, exploiting features of the H.264/AVC for improving the quality of the most important items of the soccer scene.

4.1 Ball Visibility Improvement

The key feature for delivering a satisfactory soccer content is the understanding of the soccer action. Understanding a soccer action means recognizing which team a player belongs to and locate the ball in the display. The former task is not hard to fulfill, since the two teams differ by the color of their uniforms and other contextual information, such as the position of the players in the field, helps recognizing each team. Due to the reduced screen resolution, a specific player cannot be recognizable in wide angle shots. However, this is totally consistent with the user's expectation when watching soccer videos in small handheld devices. As shown in [63], the visibility of the ball is not guaranteed. This problem is quite annoying, since, if the ball is not recognizable, the movement of the players cannot be understood.

The disappearing of the ball is caused by the spatial downsampling and low-pass filtering, due to the video encoding, performed in the transmission chain, as shown in Figure 4.1. The first smoothing step

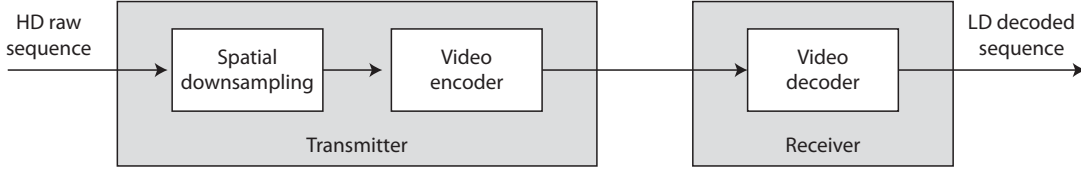


Figure 4.1: Transmission chain for soccer videos.

is performed by the resolution downsampling. The content provider delivers to the mobile operator sequences in higher resolution, say Video Graphic Array (VGA), that have to be downsampled to Common Intermediate Format (CIF) or even QCIF. Each pixel of the target resolution is calculated as the average of the appropriate pixels of the native resolution. The borders of the ball become smoothed, as they represent the averaged transition from the field pixels to the ball pixels.

A second low pass filtering is performed by the video encoder. As discussed in Section A, the picture is segmented into macroblocks. In the best case, the whole ball is contained in a single macroblock, in the worst case it is distributed among more macroblocks. In Intra encoded pictures, the block prediction is built considering the neighboring blocks that, in most of the cases, are green as the field. The difference between the prediction, a green block, and the original block, containing the ball, is then DCT transformed and quantized. Because of the quantization, other details are lost, and the ball is again smoothed towards green. In Inter encoded pictures, the prediction is built considering the previously reconstructed pictures. As the latter are already corrupted by low pass filtering and as the prediction residuals are still transformed and quantized, the smoothing effect is also present in Intra encoded pictures.

As a rule of thumb, it is beneficial to encode the Intra encoded pictures with a lower quantization parameter as, if the ball is not visible in the Intra reconstructed picture, it will vanish in the whole GOP. In order to overcome this problem, in [63] Nemethova et al. proposed a preprocessing mechanism schematically depicted in Figure 4.2.

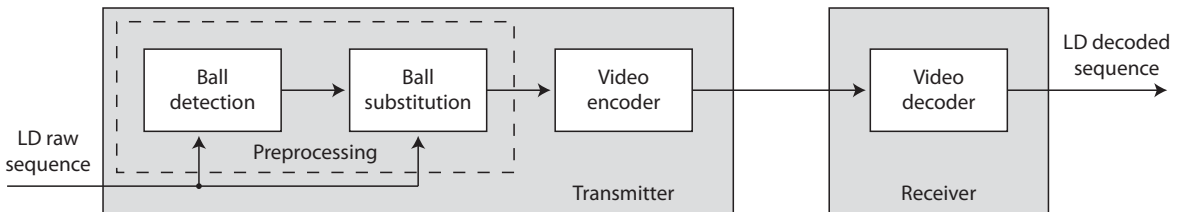


Figure 4.2: Preprocessing of low definition soccer video sequences.

As any required modification of the receiver side would have been expensive and almost unfeasible, it has been thought to implement the mechanism at the transmitter side before encoding. In a few words, the method consists of (i) the detection of the ball and (ii) of the replacement of the ball with a bigger and brighter template that is resistant to the following encoding steps.

4.1.1 MSE Based Mechanisms

In order to detect the ball, the original method as published in [15, 16] proposes the minimization of the **MSE** between the actual ball and a ball template. It considers the case where the available sequence has already been downsampled in resolution, making a common ball detection algorithm [64–66], based on shape detection or principal component analysis, not suitable. Methods considering the future position of the ball [67, 68] are also not considered because of stringent delay constraints.

A set of ball templates has been defined *a priori* considering different shape and color that the ball assumes in the common soccer video sequences. The dominant color component of the field is identified and, considering a tolerance threshold, the whole field is replaced by this color. In the first five frames after a scene change, the whole frame is compared to the templates and the points minimizing the **MSE** are chosen as candidates. After the candidates of the first five frames have been defined, the best set of positions is chosen. The best set of ball positions minimizes the distance between the position trajectory and its second order linear approximation, since it is assumed that the ball moves straight or, at most, as a parabola.

Once the best trajectory has been chosen, the position of the ball is estimated using a second order Minimum Mean Square Error (**MMSE**) estimator. The ball estimation represents the center of a square region where the search is confined, reducing the computational complexity of the algorithm. All the pixels of the search region are compared with the template by means of **MSE**. The position that minimizes the **MSE** is assumed to be the center of the ball. However, if the smallest **MSE** is higher than a threshold, no ball is found in the region. In this case, the size of the region is increased adaptively, until it reaches the borders of the frame. The absence of the ball is caused by occlusions, such as a player hiding the ball. An alternative cause can be the shape of the ball suddenly changing or an undetected scene change.

The ball detection mechanism suffers of two main flaws: missed and false detection. A missed detection occurs in case the ball is visible in the search region but is not found by the detector. A false detection occurs in case an object different from the ball is detected as ball.

In the following, the contributions of this doctoral thesis to the described method are discussed. As a first contribution, it has been tried to enhance the performance of the original method by improving its functionalities, introducing the following modifications:

- The templates consisted of the ball surrounded by a small region of field. The region of field should contribute as small as possible to the **MSE** calculation. For this reason the new templates consist of the ball surrounded by a blue area. The blue area is then substituted by the dominant color, minimizing the impact of the field pixels, as shown in Figure 4.3.

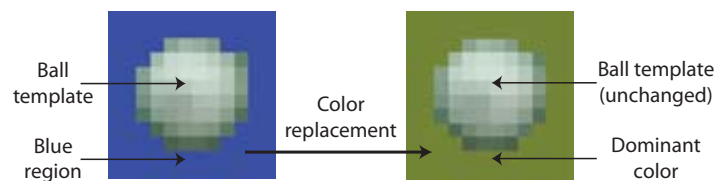


Figure 4.3: Dominant color replacement.

- The region of the field, surrounding the ball template, depends on the size of the template itself. Additionally, four rectangular regions are built around the template (Figure 4.4(a)). The MSE of the template and of the four regions is evaluated separately. If one or two neighboring regions fail the MSE test, it is interpreted as a partial occlusion, for instance the ball approaching a player. If more than two regions fail the MSE test, the sample is discarded.



Figure 4.4: Partial occlusions handling.

- The template is adaptively updated considering the original shape of the template as well as the shape of the ball as detected on the video. In order not to make sudden changes, a forgetting factor is introduced. The template \mathbf{T}_f used in the frame f can then be expressed as:

$$\mathbf{T}_f = \sum_{i=1}^5 \delta_i \cdot \mathbf{T}_{f-i}, \quad (4.1)$$

with $\sum_{i=1}^5 \delta_i = 1$ and $\delta_1 > \delta_2 > \dots > \delta_5$.

- Different estimation mechanisms (MMSE, Least Weighted Sum Error (LWSE)) of different order have been compared. Moreover, the performance of the estimator when predicting the absolute value of the ball as well as the relative movement within consecutive frames has been considered. It turned out that a third order MMSE estimator of the relative ball position was the best solution.

The estimated ball position in the frame f , $\hat{x}(f)$, can be therefore expressed as the vectorial sum of the previous position, $x(f-1)$, and the estimated movement $\hat{m}(f)$:

$$\hat{x}(f) = x(f-1) + \hat{m}(f). \quad (4.2)$$

The estimated movement is defined as

$$\hat{m}(f) = \sum_{i=1}^3 a_i \cdot m(f-i), \quad (4.3)$$

where the coefficients a_i are the solution of the MMSE estimator.

Even though these improvements enhance the performance of the method, some intrinsic limitations still bound the detection capabilities. If missed detections are not critical, causing the increase of the size of the Region Of Interest (ROI), even a single false detection can impair the ball detection until the next scene change. In case of false detection, the algorithm wrongly identifies the position of the ball. The consequences are:

1. The ROI position is incorrectly estimated,
2. The size of the ROI is set to the minimum value (in case the false detection follows missed detections),
3. The template is wrongly updated.

The three effects together make the algorithm insist on the item incorrectly estimated as ball and minimize the probability of finding the true ball.

Moreover, the method has another disadvantage, namely the extensive usage of thresholds. Even though adaptive thresholding mechanisms have been considered for adapting the characteristics of the video to the templates, the decision remains pretty empirical and is not automatically suitable for all kinds of video sequences. The method, indeed, is not able to react to some specific events, such as sudden light conditions change and fast zooming.

4.1.2 Clustering Based Detection Mechanisms

In order to overcome the drawbacks of the MSE based mechanisms, a genuine new contribution has been implemented. A similar approach for a simplified environment has been proposed in [69], in the case of RoboSoccer applications.

The basic requirement is to avoid the MSE based comparison for matching the template with the search region. The MSE performs a pixel-wise comparison of the template with a subsection of the search area, whose size is consistent with the template. As the size of the ball is really small, it can be contained into a square region of 4×4 pixels, and its shape is constantly changing over time, a pixel-wise approach is not appropriate.

As we are looking for an *item*, the ball, rather than for a *pattern*, a more suitable mechanism based on object recognition has been implemented. In a few words, rather than looking for a collection of pixels, representing the ball, the algorithm looks for an object with some specific characteristics in terms of color, shape and size. The method, schematically depicted in Figure 4.5, consists of

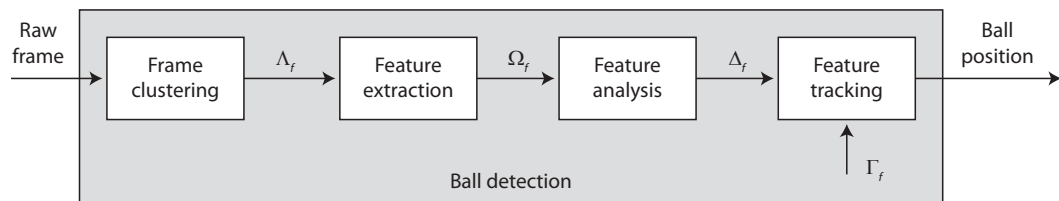


Figure 4.5: Clustering based detection mechanism.

a clustering algorithm followed by features extraction and analysis blocks. In the following, the functionalities of the blocks are briefly described.

4.1.2.1 Clustering

The clustering algorithm analyzes each frame f , or a ROI, and returns a set of candidate balls Λ_f . The clustering mechanism is described in Figure 4.6. It is well known [70] that the Red Green Blue (RGB) color space suffers of high inter-component correlation, as high values of Red usually correspond to

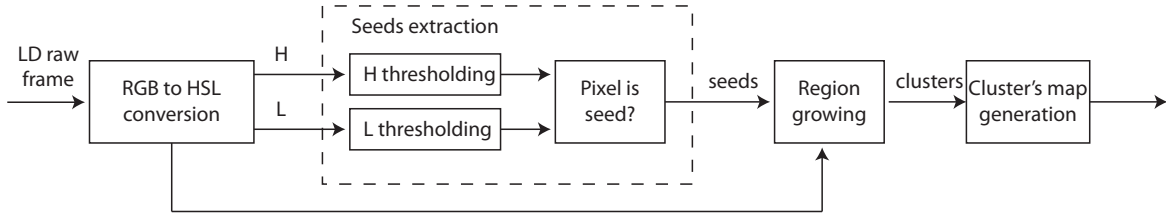


Figure 4.6: Clustering mechanism.

high values of Green and Blue. For this reason, instead of the **RGB** color space, the Hue Saturation Lightness (**HSL**) color space has been employed. The **HSL** color space consists of the following three components:

H represents the Hue, the tone of the color. The hue describes the pure color, as defined in [71]:

"the degree to which a stimulus can be described as similar to or different from stimuli that are described as red, green, blue, and yellow."

The Hue is expressed as an angle in the color wheel, with 0, 120 and 240 being associated to pure red, green and blue, respectively.

S describes the color's Saturation. The saturation, also known as colorfulness, indicates how much the color differs from gray. High saturation values are associated to vivid (pure) colors, low saturation values to gray tones.

L is the Luminance. The Luminance, as well as the slightly differently defined Value, is an index of the brightness of the color as perceived by the human eyes. A low value of luminance is typical for dark colors, nearly black, the highest values of luminance are used for describing the white color.

The clustering algorithm is based on a hybrid thresholding and region growing [72, 73] algorithm. The scope of the clustering is to identify objects with features similar to those of the ball candidates. The first property taken into consideration is the brightness of the pixels. By means of thresholding the brightest pixels are highlighted, as the ball is assumed to be almost white. These pixels represent the *seeds* of the ball candidates.

The seeds are ranked for increasing brightness and are considered the starting item of a ball candidate. By means of a *region growing* algorithm, the pixels neighboring a candidate (a seed as a first step) are analyzed. In case their Hue and Luminance components match the average values of the candidate, they are added to the region and their surrounding pixel are, iteratively, analyzed. Saturation has not been taken into account, since it did not improve the performance of the algorithm despite a slight increase of the computational complexity.

The process ends when all the pixels surrounding the candidates have been analyzed. The clustering algorithm outputs for each frame a set of candidates $\Lambda_f = \{\lambda_{f,i}\}$, where each candidate $\lambda_{f,i}$ is represented by a rectangular map, as shown in Figure 4.7. The map is built upon the rectangle circumscribing the shape of the candidate. Each pixel of the map can be either an *item*, if it belongs to the candidate, or a *hole*, if it does not.

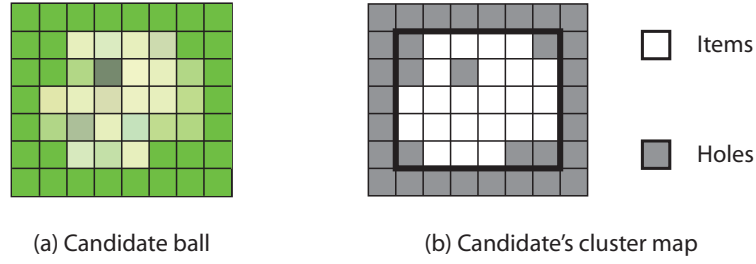


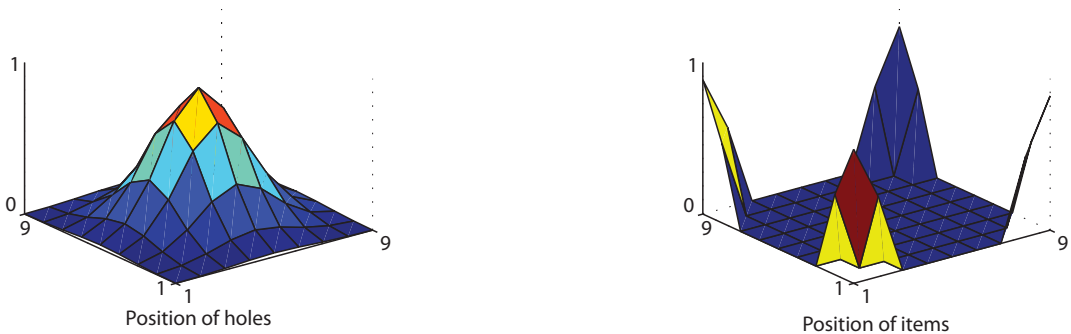
Figure 4.7: Cluster map.

4.1.2.2 Feature Extraction and Analysis

For each frame, the feature of each candidate $\lambda_{f,i}$ is analyzed in order to extract a set of features $\Omega_f = \{\omega_{f,i}\}$. The features are then analyzed and compared to those of a typical ball.

Two features are directly obtained from the clustering process, namely the average cluster Hue and Luminance components. Due to spatial downsampling, the resulting ball color is not white as expected, but rather light green. For lower values of luminance, a range of Hue components around the color green is considered. The size of the range increases with increasing values of lightness, as the color green is considered. The size of the range increases with increasing values of lightness, as the color becomes indistinguishable from white. Candidates having color characteristics outside the permitted range are discarded.

Three shape properties are extracted from the binary map associated to the $\lambda_{f,i}$. The *Axis Ratio* [74] indicates the proportion between the two dimensions of the rectangular map. As the ball is assumed to be almost round, it should approximate the value one. The cluster *Density* [74] is defined as the ratio between the number of items in the map and the area of the rectangular map. For a ball, the density should be similar to the ratio of the circle area and the area of the square the circle is inscribed into, that is $\pi/4$. The cluster *Roundness* measures the similarity of the shape to a circle. As direct shape comparisons are not effective, the candidate has to be characterized by *items* in the center and *holes* in the corners. Deviations from this requirement are penalized by means of the two-dimensional functions depicted in Figure 4.8.

Figure 4.8: Penalizing functions for 9×9 pixel cluster.

A hole in the center of the cluster is penalized much more than a hole in the surrounding area, whereas holes in the corners are not penalized at all. An equivalent rule is defined for the items, that are penalized more when close to the corners.

The shape properties are evaluated with respect to the cluster size. Big clusters have to approximate better the real shape of the ball, small clusters, possibly characterizing the ball in wide angle shots, reflect only roughly the true ball's features. Again, the candidates whose features are not compatible with the reference, are discarded. As a result of the features extraction and analysis, a set of refined candidates with their features $\Delta_f = \{\delta_{f,i}\} \subset \Omega_f$ are defined. The similarity between each feature $\delta_{f,i}$ and the desired value is measured by means of an index named *ball score*.

4.1.2.3 Tracking Mechanism

A drawback of the MSE based method was the limited tracking possibility. The tracking was bounded to the position of the center of the template. When using clusters, a set of features is described. The new approach allows for an enhanced tracking mechanism, comprising also the characteristics $\delta_{f,i}$ of the cluster.

Once a sequence of M frames has been processed (with M depending on the frame rate, for example $M = 5$ for frame rate equal to 30 frames per second) the algorithm builds a temporal succession of the candidates $\delta_{f,i}$ with $f \in [1, M]$ called *paths*. Each candidate $\delta_{1,i}$ (belonging to the first frame) is defined as root of a path. Each path is initialized by inheriting the spatial and color properties of the cluster it is starting with.

Once the first set of the paths $\Gamma = \{\gamma_j\} = \delta_{1,j}$ has been created, for each of the following frames we evaluate the similarity between the properties of each candidate $\delta_{f,i}$ and each existing path γ_j . We first process the physical distance, that has to be below a certain threshold depending on the frame rate. If a path has a cardinality higher than one, a presumed position is predicted considering the average x and y speed.

Subsequently, the color, size, and shape characteristics of the candidates are evaluated with respect to the corresponding properties of each path. In order to suit the variability caused by zooming and changing light condition, the path characteristics are averaged considering a window of N frames, where N depends on the frame rate (for example, $N = 7$ for frame rate equal to 30 frames per second). Each candidate is then attached to the path that results to be the most compatible with its characteristics. If more than one candidate $\delta_{f,i}$ is associated to the same path γ_j , the path γ_j is duplicated and each replica is connected to a unique candidate. If no path appears to be appropriate for a cluster, a new path is initialized.

Each of the previous comparisons returns a certain quality index defined as *path score*. The best path is then chosen considering the maximum sum between the *path scores* and the average *ball score* of the path items.

4.1.2.4 Results

The simulations were performed over a set of five soccer video clips (labeled "fussball n ", with $n \in [1, 5]$) in CIF (352×288 pixels) resolution. The sequences were recorded at 15 frames per second, the sequence length was chosen to simulate the true camera shot time. The results are shown in Table 4.1.

For each of the investigated sequences we manually counted the number of frames where the ball was visible and neither occluded nor out of the shot (third column in Table 4.1, labeled as "#balls").

Seq. Name	#frames	#balls	<i>det</i>	<i>false</i>	<i>miss</i>	<i>acc</i>
fussball 1	141	130	124	1	5	95.4 %
fussball 2	141	137	132	1	4	96.4 %
fussball 3	141	135	128	1	6	94.8 %
fussball 4	141	136	131	2	3	96.3 %
fussball 5	191	126	117	11	7	92.8 %

Table 4.1: Detection and tracking performances.

The following columns show the number of correct detections (*det*), the number of false detections (*false*), and the number of missed detections (*miss*). The accuracy (*acc*), indicating the number of correct detections compared to the number of frames where the ball was present, has been chosen as an overall quality index.

The first four rows of Table 4.1 show the results for common wide angle shots sequences, both with artificial or natural light, as shown in Figure 4.9(a) and (b). The average ball size lies over $4 \times$



Figure 4.9: Characteristics of different soccer shots.

4 pixels and the shirts of the two teams are easily distinguishable from the ball. The number of false detection remains in all cases extremely low and the average accuracy is over 95%.

We tested also a worst case (fifth row of the Table 4.1, labeled as "fussball 5"), where the shots were characterized by an extreme wide angle as shown in Figure 4.9(c) and (d).

The wide angle results in a ball of 2×2 pixels. Therefore, due to downsampling, the ball color is green, slightly lighter than the field. This effect can be observed in Figure 4.9(c) where the position of the ball is indicated by an arrow. Moreover, during fast movements, the ball tends to fade to the background green and, in steady frames, it is not easily detectable by a human viewer either.

Another problem arises if a team has uniform colors that confuse the algorithm. Green shirts with white insert (arms, writing on the front or on the back), white shorts as well as white socks surrounded by green fields can erroneously be detected as a candidate ball. In case of occlusion, the thresholds have been reduced and objects similar to the ball can cause false detection. Figure 4.9(d) shows an advertisement in the player's shirt detected as ball during an occlusion.

The impact of these effects is strongly limited by the path analysis. Objects looking similar to the ball vary their size, shape and color in time depending on their exposure to the camera. Additionally, the algorithm works in a very conservative way. Objects found after an occlusion (or, generically, after a certain number of frames where no ball was found) are not immediately added to the path and, therefore, are not considered for updating the object properties until the resulting path shows

compatibility with its predecessors.

In the considered sequence, the ball was occluded or not visible in over 34% of the frames. Despite a higher number of false detections, the algorithm was able to follow the true ball as soon as it was visible again.

4.2 Encoding Optimization

One of the most limiting drawbacks of the method presented in Section 4.1 is the necessity of altering the content of the sequence. Although the brand of the ball is nearly impossible to recognize in wide angle shots, the proposed algorithm modifies a sequence introducing an external item: a brighter and bigger template. This is often not allowed due to legal issues (copyright).

In order to overcome this problem, an encoding optimization mechanism has been developed, aiming at maximizing the rate distortion behavior of the encoder without the necessity of artificially modifying the content of the sequence. To this extent, optimizing the encoding mechanism means associate more bits to the most important and less bits to the least important regions of the frame. The performed investigation addresses the maximization of the user's experience rather than the improvement of objective distortion metrics, therefore, the word *important* refers to subjective importance.

Which objects are then *subjectively* most important in the context of low resolution soccer streaming? The most important objects are the items that capture the attention of the viewer as well as the items that are necessary for the correct understanding of the game and, if not properly displayed, degrade the user experience. As discussed in Section 4.1, the ball visibility is necessary for a satisfactory quality of experience. The quality of the players has an influence as well, as this visual information is necessary for a better game understanding. The field is characterized by an almost homogeneous pattern and a specific color, green. Moreover, the grandstands and other objects, although not directly involved in the soccer action, are displayed in the frame. Approaches already available in the literature, propose the recognition of the area where the action is taking place and, in case of wide angle shots, the automatic cropping of the subregion of interest [18, 19]. However, no cropping is considered in this work and the full resolution is kept.

In the following sections, it will be discussed how the specific characteristics of the frame elements are exploited for optimizing the encoding of the picture. More specifically, Section 4.2.1 describes the segmentation mechanism utilized for recognizing the frame regions. Section 4.2.2 explains how the FMO has been used for optimizing the sequence encoding. In Section 4.2.3 it is explained how some information included in the encoded sequence can be exploited for describing the characteristics of the sequence and, in Section 4.2.4, an application of this information is proposed.

4.2.1 Frame Segmentation

As discussed in the Appendix A, the encoding mechanism of H.264/AVC is block based. The granularity of the segmentation is, therefore, as small as one macroblock. The segmentation mechanism outputs a map indicating which region each macroblock belongs to. For sake of simplicity, the three regions have been labeled as follows:

- **R0** The field,

- **R1** The ball, the players and the field lines,
- **R2** The audience.

The input images are in **RGB** format. The problem of segmenting the soccer field has already been discussed in several works in the literature, such as [75–79]. These methods, however, either perform a segmentation by means of complex color spaces that, in some cases, is not fully unsupervised or aim at recognizing and track objects such as specific players or the ball. In this work, we implemented a standard color space clustering mechanism able to perform the picture segmentation with the resolution of one macroblock.

For the same reasons discussed in Section 4.1.2.1, the color space is converted to the **HSL**. The first step consists of identifying the dominant color of the field. For this reason, the histogram of the **HSL** components is analyzed, as shown in Figure 4.10. In Figure 4.10.a, the histogram of the Hue

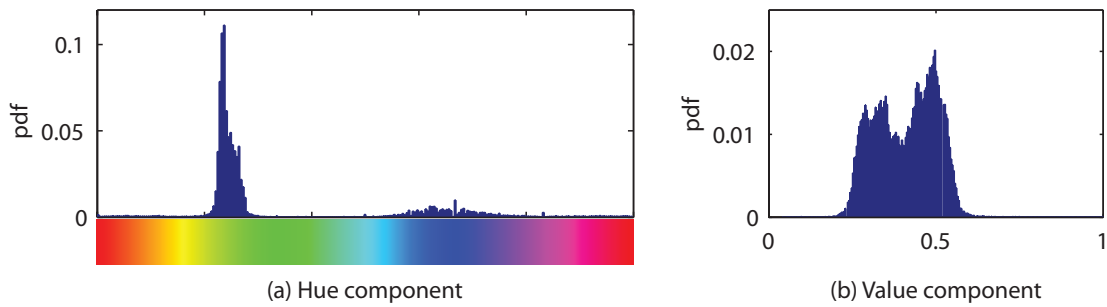


Figure 4.10: Normalized histogram of the HSV components.

component is drawn. A dominant component corresponding to the field in the green Hue range can be easily recognized. The distribution of the Luminance component of the pixels whose Hue component was recognized as field, is drawn in Figure 4.10.b. The different components reflect varying light and shadowing conditions.

The main idea behind the segmentation algorithm is to consider the information about the color of the pixels representing the field. We can therefore bound the tolerated values of Hue, Saturation and Value. Once the dominant component of the field has been detected, the field can be easily highlighted, as shown in Figure 4.11.

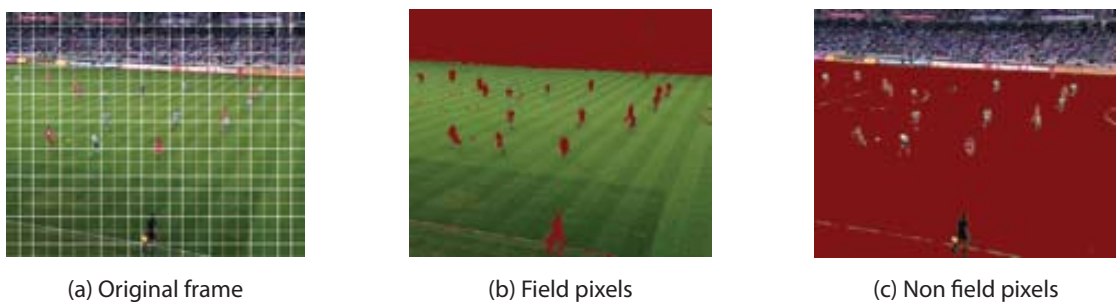


Figure 4.11: Field identification.

The segmentation mechanisms is schematically described in Figure 4.12. It starts with the identi-

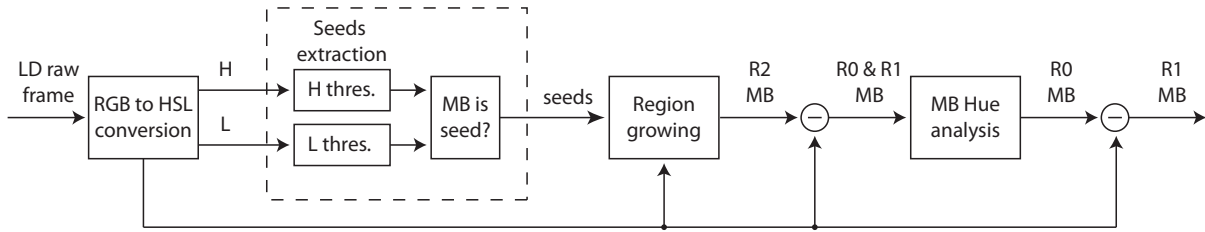


Figure 4.12: Segmentation mechanism.

fication of the macroblocks containing the audience. In wide angle soccer shots, the audience occupies the border of the frame: usually the top rows as well as the left- and rightmost columns, in case the action is approaching the penalty area of a team. In some specific conditions, such as the action taking place close to the side where the camera is mounted, some audience-like items (athletic track) can appear in the bottom rows. For this reason, the macroblocks containing the audience are detected by means of a region growing algorithm [72] with seeds placed on the four corners of the picture. A seed-macroblock may belong to the audience or to the field, depending on its color characteristics. If its color components histogram is compatible with the field dominant color, such seed is discarded, otherwise it is considered as the first macroblock of an audience region. The surrounding macroblocks are then evaluated, and, depending on their color characteristics, can be either attached to the audience region or discarded. The process is terminated when all the border macroblocks of the audience regions have been examined.

Once the audience macroblocks have been detected, the remaining macroblocks belong to R0 and R1. The macroblocks containing a dominant color quota higher than a threshold are associated to the field, the macroblocks that remain unlabeled are, by default, associated to R1. The result of the segmentation process is shown in Figure 4.13(a). A last refinement is necessary to correct some



Figure 4.13: Results of the segmentation process.

macroblocks that have been, possibly, associated to R2 instead of R1. This can happen in case some macroblocks belonging to R1 are next to the audience. As shown in Figure 4.13(b), the region growing mechanism determining R2 includes also players that partly overlap the audience.

However, the audience region cannot be concave or convex. Under this assumption, the rows containing both audience and field macroblocks (such as the fourth row in Figure 4.13(b)) belonging to the field are further analyzed. If a macroblock originally assigned to the audience is surrounded, left and right, by field macroblocks, this means that the macroblock has been incorrectly assigned to

R2 whereas, in reality, it belongs to R1.

4.2.2 Application of FMO

The result of the segmentation process is a map associating each macroblock to the appropriate region. This information is used for optimizing the encoding mechanism, exploiting Flexible Macroblock Ordering (**FMO**). **FMO** is an error resilience feature that allows the encoder to sort the macroblock in orders different than the raster scan.

The standard scan procedure, the *raster scan*, sorts the macroblocks row-wise from the top to the bottom. If the size of one slice, either in bytes or in number of macroblock, has been specified, a new slice is initialized as soon as the condition is fulfilled. This behavior is described in Figure 4.14(a), where a picture in **QCIF** resolution has been encoded limiting the size of a slice to 33 macroblocks.

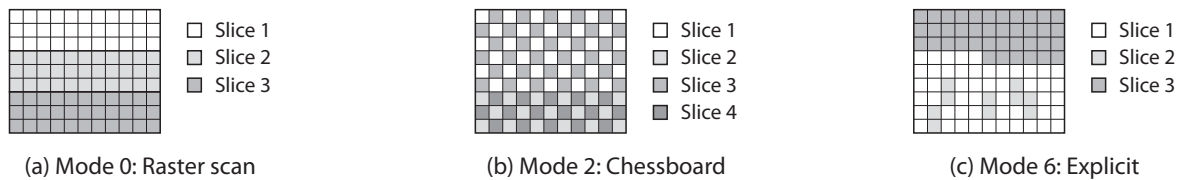


Figure 4.14: Sample FMO approaches.

By means of **FMO** different scanning procedures can be defined. For instance, in Figure 4.14(b), the chessboard scheme is represented. Also in this case, the size of a slice is limited to 33 macroblocks. The slices 1 and 2 belong to the slice group A, whereas the slices 3 and 4 belong to the slice group B. Six different **FMO** mechanisms are defined. In the so called *explicit mode* the macroblocks are associated to a specific slice group according to an explicit map, as shown in Figure 4.14(c).

For the encoding optimization, this last method is the most suitable. The association map between macroblocks and slice groups is the result of the segmentation algorithm. Applying **FMO** in this way already allows for unequal error protection. Different slices are encapsulated in different **NALUs** that, in turn, are encapsulated in different **IP** packets. Although an **IP** packet can contain more than one **NALU**, it is beneficial to aggregate **NALUs** whose payload belongs to the same slice group in the same **IP** packet. Unequal error protection, however, does not provide any direct *rate-subjective distortion* optimization of the encoding mechanism. The number of bits associated to an encoded macroblock strongly depends on the quality of the reference as well as on the selected **QP**. As further explained in Appendix A, the **QP** determines the amount of **DCT** transformed residuals that survive the quantization, that are the coefficients that are not turned to zero.

For optimizing the encoding, the amount of data associated to each region has been investigated. Different soccer sequences in **CIF** resolution have been segmented and the results in terms of code size and quality have been correlated with the segmentation's results. In Figure 4.15 the number of macroblocks associated to the three regions for a representative sequence is depicted. As a **CIF** frame consists of 396 macroblocks, the percentage has been normalized to one. As it can be expected, in wide angle shots, most of the macroblocks are containing the field (R0), around 72%. The R1, containing players, ball and field lines, represents 12% of the frame whereas the grandstands (R2) occupies 16% of the picture. The sequences have been then encoded using the explicit mode of **FMO** defined by the

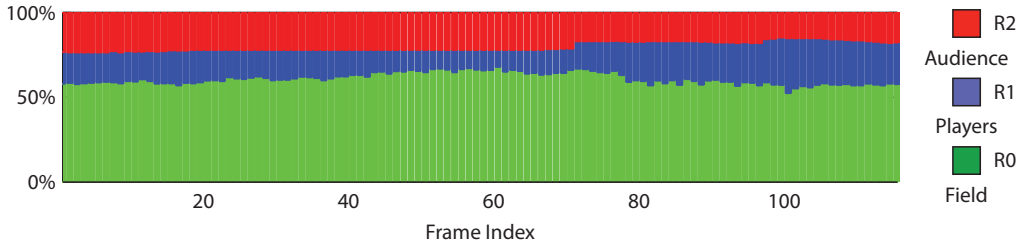


Figure 4.15: Distribution of the macroblocks belonging to the regions R0, R1 and R2.

association map obtained by clustering. Due to the difference between the encoding of the Intra and of the Inter frames, in this work we aim for optimization of the Inter frame encoding. As the Intra frames represent a synchronization mark for recovering the full quality and act as a reference for the following Inter frames, it has been decided to assign to the Intra encoded pictures the highest quality possible.

Analytically, the amount of data required for encoding each region can be correlated with the region's entropy. Each of the regions can be seen as originated by a source emitting symbols \mathbf{s}_i , pixels, with three components $\mathbf{s}_i = \{Y_i, U_i, V_i\}$, the luminance Y_i and the chrominances U_i and V_i . As each component is quantized with 8 bits, the source emits symbols out of an alphabet with cardinality $256 \times 256 \times 256$. The entropy $H(R_i)$ of the region R_i , $i = \{0, 1, 2\}$, if measured in *binits*, is defined as follows:

$$H(R_i) = - \sum_{j=1}^{255^3} p(\mathbf{s}_j) \cdot \log_2(p(\mathbf{s}_j)), \quad (4.4)$$

where $p(\mathbf{s}_j)$ represents the occurrence probability of the symbol \mathbf{s}_j . As

$$\lim_{p \rightarrow 0^+} p \cdot \log(p) = 0, \quad (4.5)$$

the sum can be limited to $N_i = |R_i| < 255^3$, representing the cardinality of the set R_i , that is the amount of symbols that have been actually emitted by the source R_i .

For calculating the entropy of each region, it has been counted how many time a given symbol has occurred in each region of one frame. As each symbol consists of 3 coefficients quantized with 8 bits, most of the symbols are just occurring once. For this reason, it has been decided to rescale the quantization of the coefficients by discarding the least significant bits. The luminance has been quantized using six bits, the chrominances with 5 bits each. For the first frame, the results are shown in Table 4.2 and discussed in the following.

Region	$H(R_i)$	N_i
R0	6.509	367
R1	8.436	2270
R2	9.666	3111

Table 4.2: Entropy of the first frame.

As expected, the region R0, the field, has the smallest entropy, followed by R1 and R2. In order to understand this behavior, it is necessary to consider the symbol's occurrence probability. As discussed

in Figure 4.15, the R0 cover more than 50% of the frame, around 58 000 pixels, R1 and R2 share the remaining area, around 19 000 and 24 000 pixels, respectively.

The entropy is maximized when all the symbols have the same probability, $p(\mathbf{s}_j) = 1/N_i$, this causes the entropy to be equal to

$$H_{\max}(\text{Ri}) = - \sum_j^{N_i} p(\mathbf{s}_j) \cdot \log_2(p(\mathbf{s}_j)) = -N_i \cdot \frac{1}{N_i} \cdot \log_2 \frac{1}{N_i} = \log_2 N_i. \quad (4.6)$$

In the case of equal probable symbols, therefore the entropy increases logarithmically with the cardinality of the set N_i . The actual measured entropies have been compared in Table 4.3 with the one of a source emitting symbols with equal probability.

Region	$H(\text{Ri})$	$H_{\max}(\text{Ri})$	$\Delta H(\text{Ri})$
R0	6.509	8.512	0.236
R1	8.436	11.149	0.243
R2	9.666	11.603	0.167

Table 4.3: Measured entropy compared to the max $H(\text{Ri})$.

The difference between the calculated entropy and the one of a noisy source has been considered in terms of relative entropy reduction:

$$\Delta H(\text{Ri}) = \frac{H_{\max}(\text{Ri}) - H(\text{Ri})}{H_{\max}(\text{Ri})}. \quad (4.7)$$

The different distribution in terms of probability density function (pdf) and cumulative density function (cdf) are shown in Figure 4.16. In both graphs the symbols have been sorted by probability of

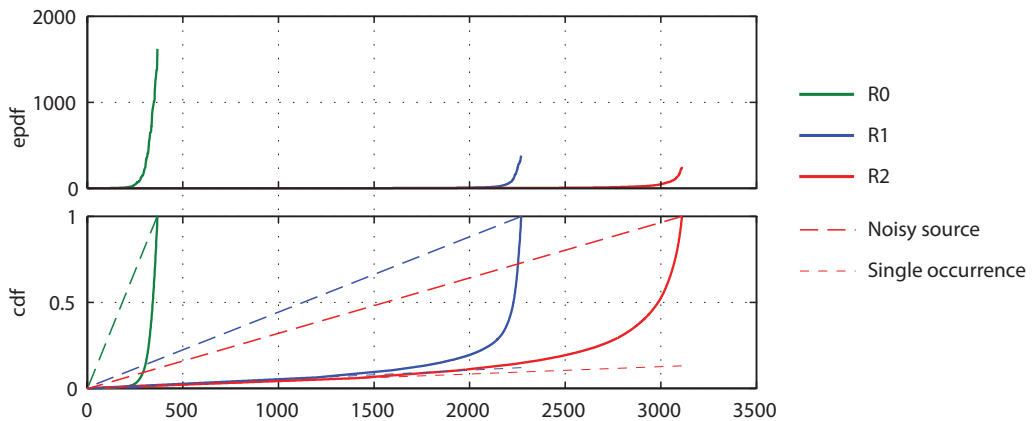


Figure 4.16: Probability and cumulative density function of the symbols.

occurrence. In the lower graph, the dashed line shows the theoretical behavior of one source emitting symbols with equal probability of occurrence equal to $1/N_i$. The smaller the area defined by this line and the cdf curve, the noisier the source. It is also shown, for R1 and R2, the cdf originated by symbols occurring only once. Even though the quantization step has been increased, almost half of the symbols emitted by R1 and R2 have probability of occurrence equal to one.

The amount of data needed for storing each region (or slice group) is drawn in Figure 4.17. Surprisingly, although the audience represents 16 % of the frame, it requires as much as half of the

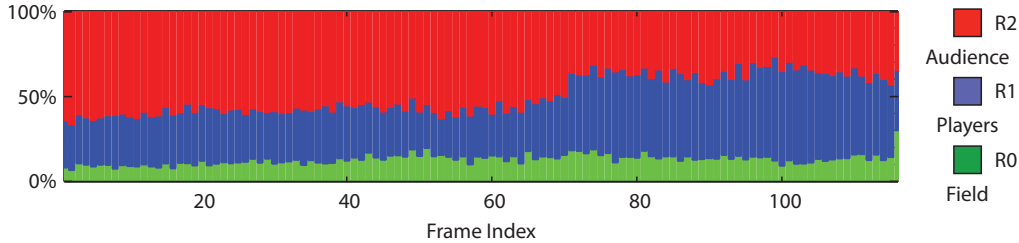


Figure 4.17: Distribution of the data rate associated to regions R0, R1 and R2.

data associated to the whole picture. The amount of data required for encoding a macroblocks of R0 is just 5 % of the data necessary for encoding a macroblock of R2. The field is characterized by a constant color tone, green, without any noticeable high frequency pattern. The only elements that introduce variations on the field are different leveling of the grass and varying light and shadowing conditions. As the prediction is already good (for further details see Section 4.2.3), almost no residuals are associated to the field. The same is, obviously, not valid for the items included in R1. The players, in particular, strongly change their appearance within two consecutive frames causing the temporal prediction to require many additional bits.

It remains then to be explained why the macroblocks of the audience require so many bits to be encoded. The audience for a human viewer, represents an almost static background, whose movement is consistent with the camera. Because of the very low resolution, possible movements of the people sitting on the grandstands cannot be appreciated. Further analysis showed how the high frequency pattern of the audience is responsible of the low efficiency of the temporal prediction.

Because of the spatial downsampling the high frequency pattern of the audience has noise-like characteristics. For human viewers, the audience in two consecutive downsampled frames looks almost the same except for the possible camera movement. From the encoder point of view, however, differences arise at high frequency. When applying temporal prediction, many coefficients are necessary for correcting the high frequency discrepancies between the original macroblock and its prediction, even though these differences are not comprehensible by human viewers.

For this reason, in order to subjectively enhance the user’s experience, the optimization mechanism drawn in Figure 4.18 has been implemented. By means of QP selection, the quality of each region

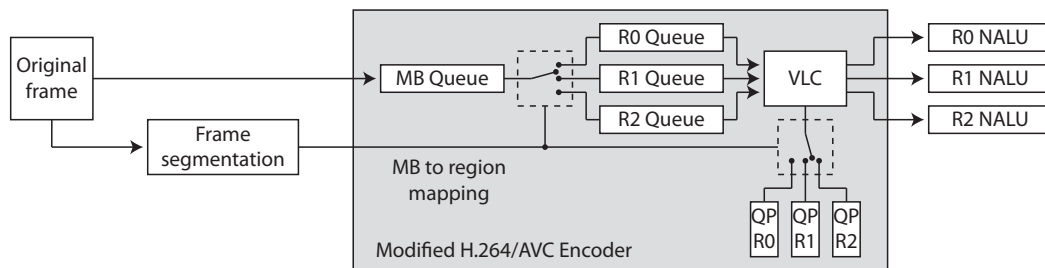


Figure 4.18: Block diagram of the modified encoding mechanism.

can be tuned depending on its subjective importance. In H.264/AVC a similar result can be obtained without FMO by modifying the parameter `mb_qp_delta` (indicating the delta between the global QP and the QP currently used) of each macroblock. This, however, introduces additional overhead for each macroblock and does not allow for unequal error protection. By means of FMO, the parameter `slice_qp_delta`, indicating the delta between the global QP and that of the current slice, is modified once for the whole slice.

Different sets of QPs have been investigated. As a rule of thumb, higher QPs have been assigned to R2 and smaller QPs to R1. As for the field nothing was predictable a priori, the range of QPs varying from 26 to 42 has been analyzed. In Figure 4.19(a) the bit rate reduction when utilizing different QP

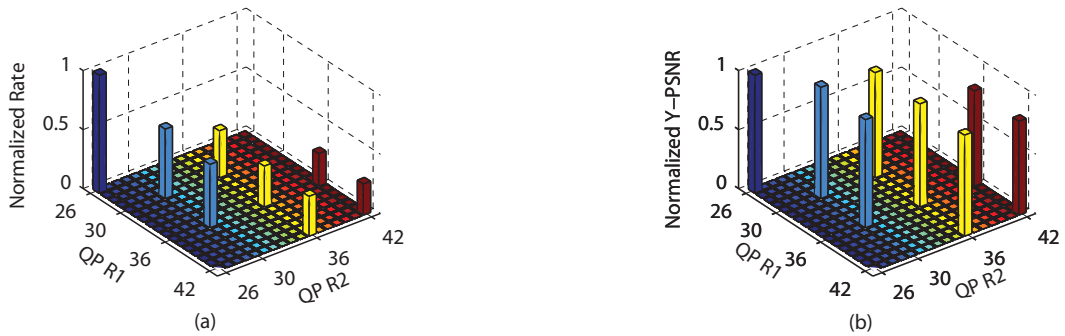


Figure 4.19: Dependency of rate quality on the QPs.

sets is depicted. The values are normalized considering the code size obtained with a constant QP equal to 26 for the three regions.

The dependency between the code size and the QP assigned to R2 is clearly identifiable. As discussed previously, the coefficients of R2 are mostly associated to high frequency corrections. When increasing the QP, more high frequency coefficients are cutted, resulting in decreasing size of the encoded macroblocks. The correlation between size and QP of R0 is much smaller. Increasing the QP of R1 of 12 units (from 30 to 42) while keeping the QP of R2 constant equal to 30, causes a reduction of the code size by 49%. The equivalent modification of the QP of R2 causes a reduction by 15%.

Considering the extreme case, when the QP of the audience is set to 42 and the QP of the field is modified from 36 to 42, the code is reduced by 20%. Modifying the QP of R2 under the same condition, would result in a code reduction by 7%.

A study of the effects of the QP sets on the Y-PSNR of the sequences has been performed as well. Surprisingly, as shown in Figure 4.19(b), the behavior of Y-PSNR as function of the QPs is much different than behavior of the rate as a function of the QPs. Even for the set (42,26,42), where the rate was almost 25% of the original, the Y-PSNR remains about 80% of the original. As observed for the rate, also the objective distortion metric appears to be marginally dependent on the QP applied to the field. The variations are to be attributed to the effect of the quantization of the grandstands.

However, even if the prediction at the encoder is performed minimizing an objective metric as Y-PSNR, this work targets the optimization of the encoding considering the subjective quality as perceived by the observer. Even though some methods in the literature propose mapping of the Y-PSNR to subjective metrics considering the characteristics of the frame and the sequence, in this analysis the subjective impact of the frame's subregions has been analyzed.

In order to test the performance of the method, the user evaluation of some specific sequences encoded selecting different QPs sets has been measured. The quality of the uncompressed sequence has been considered as a reference. Although the MOS score ranges between 1 (poor) and 5 (excellent), the uncompressed sequences had an average evaluation slightly smaller than 4 (good). This is because most of the test persons were not familiar with soccer sequences displayed on small resolution screens. The test subjects were then asked to rate nine versions of the same sequence encoded choosing different QP sets. Three sequences were encoded without FMO, employing the same QP (26, 30 and 36) for the whole frame. Six sequences were encoded using specific QP sets.

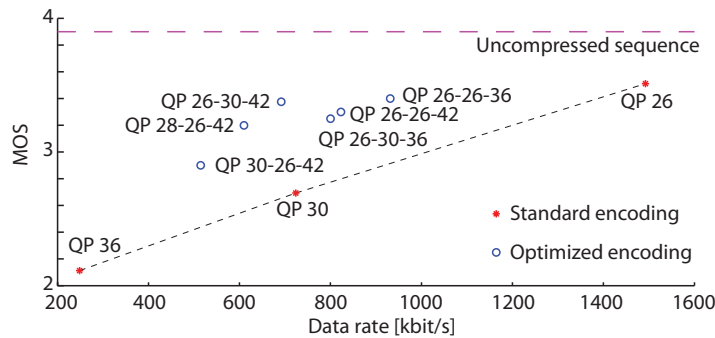


Figure 4.20: Preliminary MOS results.

The preliminary results, drawn in Figure 4.20, describe the rate distortion behavior of the different encoding schemes. Without FMO, the quality appears to be linearly dependent on the encoding rate. This has not to be considered true in general, but rather limited to the considered QP range. Due to the limited amount of possible observations, only the effects of *macro* variation of the parameters can be discussed. In particular, the stronger compression of the audience's macroblocks does not seem to impair the user experience. When comparing the QPs set (26,30,42) with a constant QP equal to 26, the data rate is halved whereas the quality has been reduced by just 0.1 MOS points.

4.2.3 Scene Analysis

The previous analysis has been performed encoding short test sequences, as the method at this stage still suffers of some intrinsic limitations. Basically, increasing the QP means reducing the amount of corrections that the encoder introduces in order to refine the prediction. The method works properly if the reference picture already offers a valid prediction to the frame to be encoded. This is not true in two specific cases: (i) new objects are appearing at the border of the picture because of the camera movement and (ii) the camera zooms in or out. In this section the detection and analysis of these two cases are discussed, in the following section the necessary modifications of the encoder are presented. The two mentioned effects are both related to camera operations, namely pan and zoom, respectively, as depicted in Figure 4.21.

Pan refers to the rotation of the camera on its horizontal plane. In soccer videos the camera usually shoots the area of the field where the ball is present. In wide angle shots, being the shot objects distant from the camera, this movement is appreciated as a rigid camera translation.

The zoom is the camera feature of moving away from (zoom out) or toward (zoom in) a subject.

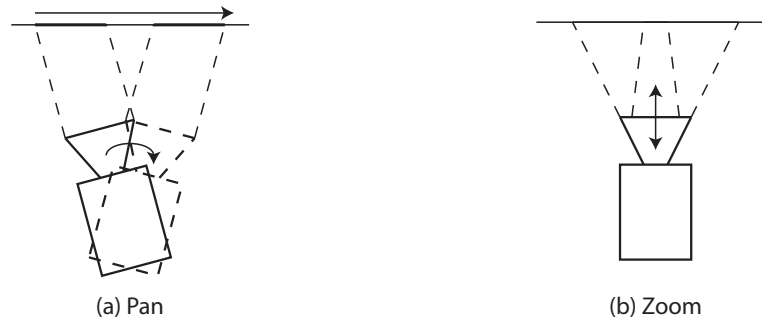


Figure 4.21: Camera operations.

In professional cameras, this action is performed by varying the distance between the lenses. In soccer videos, *zoom in* is used to highlight the field region where the action is taking place. *Zoom out* is used in case the ball moves rapidly and a wider region of interest has to be considered.

Several works in literature address the detection of pan and zoom in video sequences. Most of them exploit the concept of global motion, as defined in [80]. The contributions in [81] and [82] determine the global motion by accurate motion models, involving complex processing such as gradient descent and iterative least square estimation. In this work we exploit the motion information already contained in the motion vectors. This approach was already considered in [83] and [84]. In [84] a probabilistic model is designed to determine the averaged probability of zoom in a frame. Dumitras and Haskell [83] first address the necessity of discriminating the MBs contributing to the estimation of the global motion. In their approach, they weight the contribution from the two biggest regions of the frame, where the motion vectors share similar orientation.

We propose a specific model suitable for soccer sequences. The global motion and zoom are estimated exploiting the direction of the motion vectors, considering the contribution of a set of *reliable* MBs. In order to understand which of the macroblocks can be defined as reliable, the motion estimation applied to the three regions has been analyzed:

Field (R0) It mostly consists of low-frequency green pattern. The video encoder chooses the best temporal prediction minimizing a cost function, weighting the size of the encoded MB with respect to the introduced distortion. Because of the lack of details, the best temporal prediction may be offered by a similar block placed in the vicinity. The cost is often minimized by a zero motion compensation, as the block is copied from the previous picture.

Player and ball (R1) The players and the ball do not move consistently with the camera movement. The motion vectors associated to their MBs are almost uncorrelated. They do not offer a reliable source of global motion estimation.

Grandstands (R2) The audience represents an almost static background of the soccer videos. In low resolution soccer videos, the MBs containing the audience consists mainly of high frequency patterns. Due to the property of the audience to remain static, the motion vectors selected by the encoder share almost the same horizontal and vertical direction.

For this reason, the MBs of R2 have been selected as a reliable source of prediction for the global motion characterizing the shot. In order to demonstrate this, the distribution of the motion vectors in

the three regions has been analyzed. The *absolute* motion vector applied to each MB of the picture has been saved during the encoding and examined offline. In Figure 4.22(a) the direction of each motion

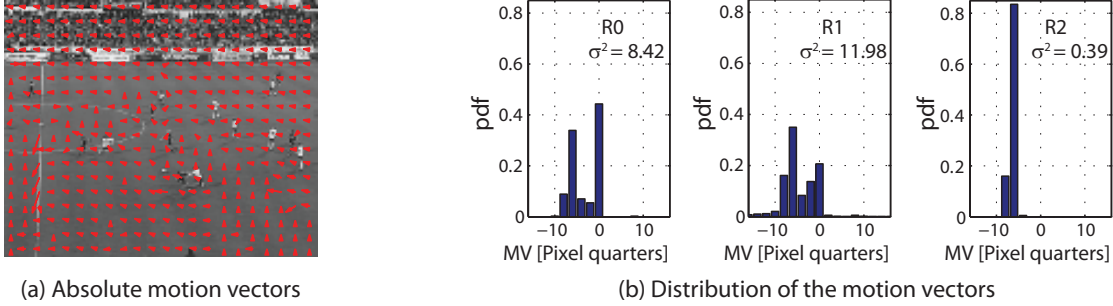


Figure 4.22: Distribution of the horizontal motion vectors.

vector applied to the each macroblock has been graphically represented. It can be noticed that the MBs of R1 do not help discriminating a dominant global direction. Most of the MBs containing the field are pointing to two different main directions. Only the MBs containing the audience are pointing to a single dominant direction.

The histograms in Figure 4.22(b) show the distribution of the horizontal motion vector in the range $[-16, 16]$ quarter of a pixel. Indeed, H.264/AVC apply motion compensation with the resolution of a quarter of a pixel, by means of weighted interpolation.

The two dominant directions in R0 as well as the absence of a dominant peak for R1 can be observed in the histograms in Figure 4.22(a) and 4.22(b), respectively. A global motion component can be recognized for R2, where 99.5% of the motion vectors are comprised within four quarters of a pixel.

To calculate the distribution's variance and mean, the outliers of the motion vectors distribution have been removed by means of statistical data pruning. In a distribution, the outliers are the elements lying at *abnormal* distance from the rest of the data. In the considered case, outliers are caused, for instance, by motion compensation of the MBs at the border of the picture. New elements appearing at the border, due to the camera movement, do not have any correspondent block in the previous pictures. Their associated motion vectors cannot be consistent with the global motion. To discriminate outliers, an implementation of the Grubbs' test [85] has been employed.

Once the global motion has been identified, the motion vectors distribution has been exploited to perform the zoom detection. It has been noticed that the variance of the audience motion vectors was strongly varying over time. In frames where the variance was particularly high, the motion vectors were distributed as in Figure 4.23(a). For each picture row, the motion vectors associated to each MB from left to right has been drawn. In the considered frame, the audience occupies the whole first five rows of the frame. As the frame is in CIF resolution, a single row consists of 22 MBs. H.264/AVC allows the subdivision of a MB up to 4×4 pixels block. This results in a maximum number of 16 motion vectors per MB. Independently of the selected subdivision, it has been decided to consider the motion compensation applied to each 4×4 pixel block, to avoid weighting issues. The motion vectors in Figure 4.23(a) vary linearly their amplitude from the left-most to the right-most MB of each row. The slope of the line remains constant in each row, as indicated by the gray broken lines. Moreover,

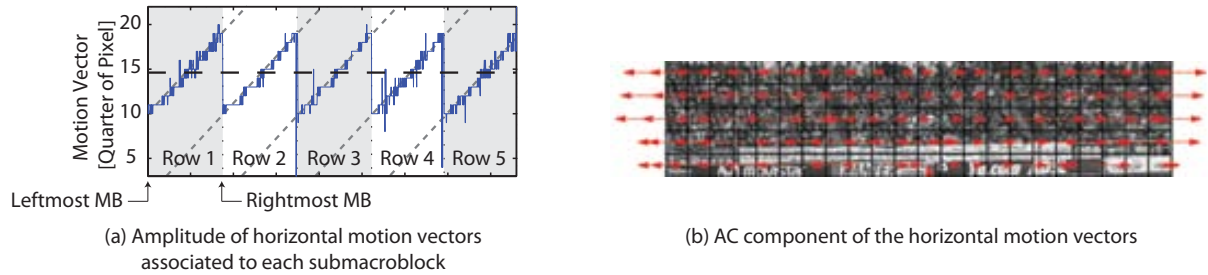


Figure 4.23: Amplitude of the horizontal motion vectors.

the mean of each row corresponds to the global motion, indicated by the bold horizontal broken line.

The amplitude of the motion vectors on the left-most MBs is smaller than the average, whereas the one on the right-most MBs is bigger. Assuming a global motion equal to zero, the motion vectors on the left side of the picture would be negative (pointing to the left) and those on the right side positive (pointing to the right). Equivalently, after removing the global motion component, the relative motion vectors of the audience take on the directions shown in Figure 4.23(b).

In order to prove the effectiveness of the method, we tested our algorithm on 50 soccer sequences extracted from a match of the first Spanish league. The original high resolution MPEG2 video has been deinterlaced and spatially downsampled to the CIF resolution.

Concerning the pan, the speed of the movement has been measured observing the detected global motion, as shown in Figure 4.24. Positive horizontal motion vectors correspond to camera movement to the right direction, while negative to the left. In a similar manner, positive vertical motion vectors indicate downward camera movement while negative upward movement. No relevant correlation was

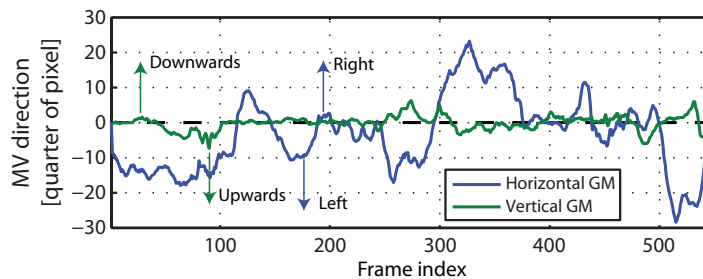


Figure 4.24: Horizontal and vertical pan.

measured between the two considered movements. As expected, the amplitude of the horizontal movement is more relevant than the vertical.

In order to measure the presence of zoom, only the horizontal movement of the camera has been considered: the width of the considered area (the grandstands) is much wider than its height and, as shown before, the horizontal component is much more significant. The trend of the slope of the motion vectors distribution, as depicted in Figure 4.23(a), has been plotted in Figure 4.25.

A smoothed slope curve has been drawn to compensate rough slope variations, particularly around the zero, using a moving window of length five. The trend of the graph describes three main behaviors:

1. Slopes around zero: no zoom has been detected in the considered frame.

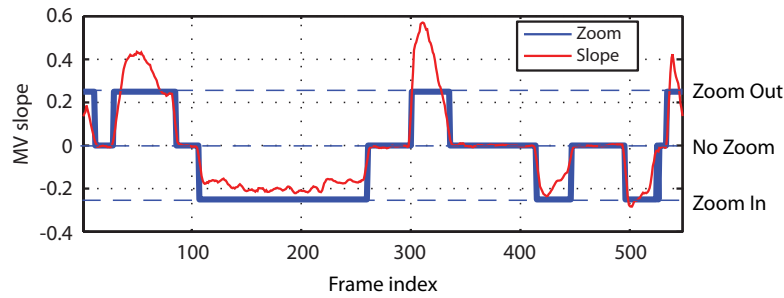


Figure 4.25: Detection of zoom.

2. Slopes bigger than zero: zoom out has been detected.
3. Slopes smaller than zero: zoom in has been detected.

The transition between the three regions is abrupt, proving the robustness of the detector. As a last step, a hard decision block has been implemented. A fixed threshold equal to 0.02 on the smoothed slope has been found as a good compromise between detection capabilities and robustness against false detection.

4.2.4 Optimization of Grandstand's Encoding

The scene analysis presented in Section 4.2.3 is exploited for tuning the encoding mechanism. In detail, the zoom detector is employed for turning on or off the encoding optimization. The Inter prediction in hybrid block based encoder compensates differences between the predicted and the actual block. In case of zoom, the region of the frame contained in the prediction block does not match anymore the actual block to be encoded. For this reason, an accurate reconstruction cannot be performed without applying additional coefficients.

The global motion estimation has been acting, in a first step, for enhancing the encoding of the frame borders. Assume now that the original reference is provided by the first Intra encoded picture of the **GOP**. As the camera moves, new objects, that were not present in the Intra picture, are appearing at the border of the picture. Since there is no adequate reference for these macroblocks, the application of a strong quantization impairs the quality of the reconstructed items. Once the camera has moved enough, the whole audience consists of improperly encoded macroblocks.

The global movement is estimated exploiting the already encoded macroblocks. The encoder is made aware of the amount of new pixels appearing at the frame border. Once the movement in any direction exceeds a given amount of pixels (the threshold has been set equal to eight pixels), or if, equivalently, new objects have not been properly encoded in the last frames (the threshold has been set equal to three frames), the border macroblocks are encoded using the same **QP** chosen for the Intra encoded picture. The schematic representation of the improved method is shown in Figure 4.26. The current implementation relies on a two-pass encoding scheme. The standard encoder is utilized for obtaining the necessary motion information, the optimized decoder produces the stream sent to the users. This solution sounds computationally expensive. However, the considered scheme consists of an encoding procedure that serves several users. Moreover, the computational cost at the transmitter

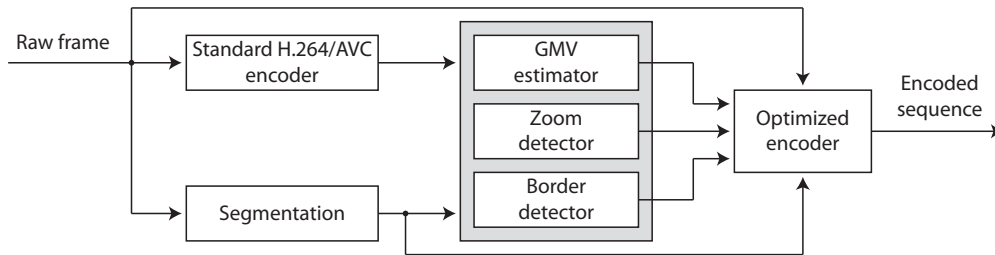


Figure 4.26: Block diagram of the enhanced encoding mechanism.

side is not a major issue, as it can be reasonably assumed that no stringent limitations on hardware and energy consumption are in force.

Even though the encoding of the audience using a higher QP already showed significant advantages in terms of rate-distortion behavior, another subjective consideration is exploited for saving further bits. The audience, as pointed out, represents an almost static background whose apparent movement is driven by the camera movement. Following this observation, it has been considered the application of a rigid movement to the whole audience macroblocks according to the estimated global motion. This can be implemented in H.264/AVC by forcing all the motion vectors of the audience's macroblock to point the direction indicated by the global motion vector. Moreover, in order to prohibit the encoder to apply corrections by means of coefficients, the coded block pattern is forced to be zero. As (i) the absolute motion vector matches the prediction, (ii) there is no block subdivision and (iii) no coefficients are associated to the encoded macroblock, most of the MBs are *skipped*, that means they are reconstructed at the decoder side just by applying motion compensation to the reference picture. For encoding such macroblocks, only few bits are necessary for signaling the amount of the consecutive macroblocks that have been skipped.

The same consideration about the border MBs made at the beginning of this section, is still valid for this enhanced optimization scheme. The new elements appearing at the border of the picture cannot be properly encoded by skipping the macroblocks. As H.264/AVC allows the motion vector to point to a region outside the frame, if no coefficients are applied for refining the prediction, the border macroblocks are not properly displayed. Allowing the encoder to apply the standard Intra encoding to these macroblocks solves the problem.

A last major consideration concludes the analysis of the mechanism. As introduced before, H.264/AVC applies motion compensation with the resolution of a quarter of a pixel. As the frame consists of an integer number of pixels, applying a fractional movement means to interpolate the reference block. However, in this specific approach, no coefficients are used. The fractional movement blurs, therefore, the audience displayed at the user side. The motion compensation is applied only for multiple integers of a pixel. A movement buffer is implemented, in order to accumulate the fractional movement not applied to the motion compensation.

4.2.5 Results

As pointed out in Section 4.2.2, the proposed method aims at the reduction of the required data rate while keeping the subjective quality unaltered. The performance will be, therefore, discussed with

respect to the associated code size as well as with respect to the collected MOS results.

The first results are shown in terms of code saving. The chart in Figure 4.27 shows the code size associated to a test sequence encoded with different coding strategies. The leftmost bar represents

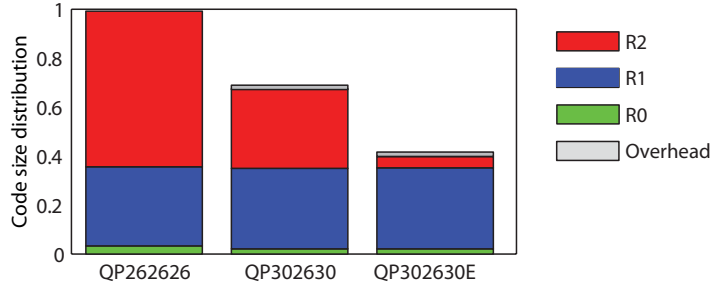


Figure 4.27: Rate comparison using different approaches.

the resulting size when encoding with the standard H.264/*AVC* encoder. The whole frame has been encoded using a single QP equal to 26. It has been considered as reference and, therefore, normalized to one. Without utilizing FMO, the size associated to each region has been calculated. As already discussed, more than 60% of the data is necessary for encoding the macroblocks belonging to the audience.

When comparing the standard encoding mechanism with the proposed mechanism, the overhead of the FMO explicit map has to be considered. The association map is contained into the PPS and, as a new map is necessary for each frame, a new PPS has to be sent before each frame. The current implementation of the standard makes the use of explicit FMO extremely expensive in terms of signalization overhead. Besides the standard PPS information, in average around 100 bits, the region each macroblock belongs to has to be explicitly encoded. As the number of regions is not known a priori, the region index is encoded by means of unsigned exp-Golomb code. The proposed implementation consists of 396 MB subdivided into three regions. Using the exp-Golomb nomenclature, the codeword associated to the region indexes are depicted in Table 4.4. Assuming the macroblocks

Region Index	Exp-Golomb Codeword
0	0
1	010
2	011

Table 4.4: Region index to codeword association.

equally distributed among the three regions, 2.33 bits are necessary for encoding each region. As, in reality, the macroblocks are not equally distributed, and the probability is known a priori, the shorter codeword is assigned to the field, R0. This reduces the average codeword length to two bits. However, the explicit definition of the association is much less effective than a run-length encoding scheme. The audience macroblocks occupy usually the first rows of the frame, the remaining region consists of the field interleaved by macroblocks belonging to R1. Even though smarter methods can be implemented for signaling the encoding map, they would not be standard compliant and not understood by the standard decoder. For this reason, the following results have been obtained when considering the

standard, penalizing, PPS encoding.

The bar QP302630 represents the results obtained by means of the encoding mechanism described in Section 4.2.2. For encoding the field and the audience a higher quantization parameter has been set, if compared to the one used for the player and the ball. This causes a noteworthy reduction of the data associated to R2, being 51% of the original. Since the field consists of low frequency components, the reduction for R0 is not that appreciable. Keeping the same QP settings, the bar QP302630E describe the results when using the method described in Section 4.2.4. Due to perspective distortion and reference picture degradation, each 30 frames the standard Intra encoding has been applied to the whole audience. The proposed method requires, for encoding the audience, 6.17% of the data required by the original encoder.

A detailed description of the behavior of the audience encoding optimization compared to the application of is drawn in Figure 4.28. The two graphs show, for each frame of the sequence, the

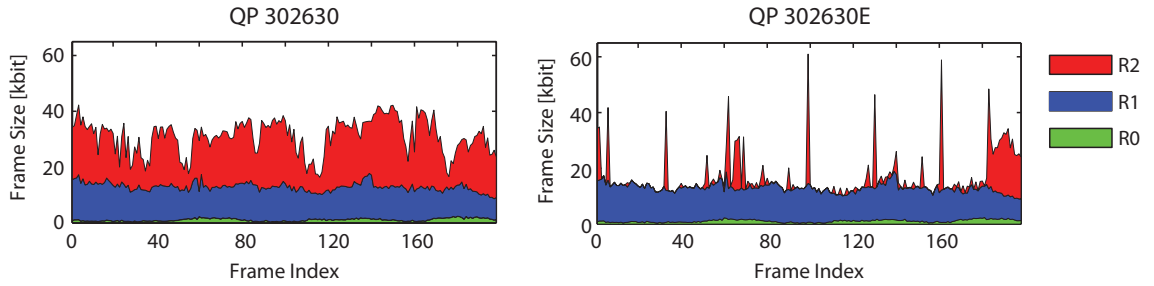


Figure 4.28: Time variant rate distribution.

size associated to each region. Since the proposed method affects only the encoding of the audience, R0 and R1 require *almost* the same amount of data. The behavior of the graph associated to the audience encoded with the proposed method deserves further discussion. In most of the frames, the audience is encoded storing solely the value of the global motion vector. Almost all the macroblocks are skipped. The small peaks are associated to frames whose border macroblocks are encoded as normal P macroblocks. Large peaks are caused by refresh pictures inserted at regular intervals. The size of the refresh pictures is slightly bigger than the size of the same frame encoded without the proposed method. This is caused by the considered reference picture that, being the result of successive picture translations, represents a worst source of prediction.

In this work we address the transmission of soccer video sequences over UMTS networks. The available net video data rate is, excluding the IP, RTP and UDP headers, usually bounded to 220 kbit/s. A preliminary test with a limited number of test users was performed to determine the optimal QP sets for low bit rate. Surprisingly, it has been observed that the quantization parameter of the field has to be kept smaller than the QPs of the R1 and R2. As mentioned before, the macroblocks containing the fields consist mainly of low frequency components, since they usually represent a flat green area. Possible high frequency components are responsible for the transition between different tones of green, due to, for instance, shadows or artificial lights. Cutting these high frequency components cause the reconstructed picture to be affected by blockiness. Even though the QP of the field is significantly smaller than the other two, for the above mentioned reasons, this does not result in considerable increase of the associated rate.

In order to subjectively evaluate the perceived quality of the video, a web page was realized. 22 volunteers were asked to rate four soccer shots encoded with different encoding parameter, the grades are varying from one (bad) to five (excellent). The chosen judgment method is the Absolute Category Rating (ACR). After displaying the original uncompressed sequences as a reference, the test subject rated the encoded versions randomly displayed on screen. This methodology has been standardized by ITU-T under the name of ACR.

Each soccer sequence has been encoded once using the Standard Encoding (SE) with Constant Bit Rate (CBR) enabled and three different sets of QP using the Optimized Encoding (OE) described before.

The results of the investigation is plotted in Figure 4.29. For each sequence, the average MOS for the SE is compared with a single value of OE, the one that has provided the best subjective results, namely QP = 24, 32 and 38 for R0, R1 and R2, respectively. The score assigned to the OE sequences

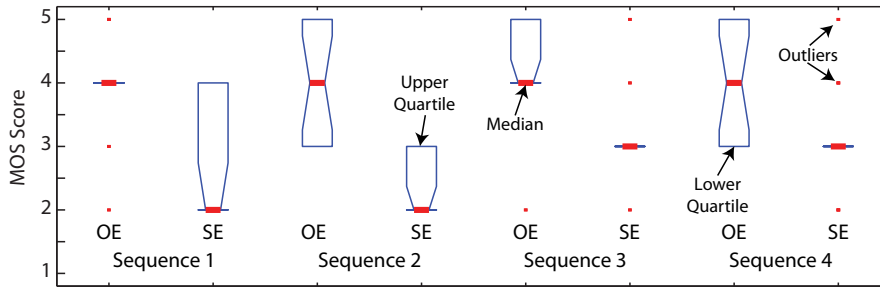


Figure 4.29: Soccer MOS results.

clearly overcomes those measured for SE. In terms of average MOS, we measured an increase of over 1.24 points. We also discriminated two main groups of test persons. The first subset assigned almost the same mark (high or low) to all the sequences. Quite all the outliers were produced by such users. For the second one, a high variance in the marks has been noticed. We assume that the users belonging to this group are those more accustomed to low resolution football streaming and, therefore, already knew the quality to be expected.

4.3 Conclusions and Self Criticism

Two different methods for improving the user experience when observing low resolution soccer video sequences have been presented in this chapter. Although soccer is one of the contents that are more appreciated by nomadic users, it is particularly sensitive to compression artefacts.

The ball carries the most important information for understanding the game. However, as its size in low resolution videos is limited to few pixels, due to the encoding low pass filtering, it can become visible at the user side. By means of a preprocessing mechanism, the ball is detected and replaced by a bigger and brighter template that is still visible after the encoding. The focus is on the ball detection mechanism, performed on the already downsampled sequence. The contribution of this doctoral thesis is the implementation of a clustering based detection mechanism, that aims at the detection of an object, the ball, with specific features, rather than an aggregation of pixels that approximate a set

of ball's template. With respect to the original method, based on MSE calculations, the clustering mechanism requires a smaller computational effort.

The method has been developed considering the setup of a network provider, that usually receives the stream already downsampled by a content provider. However, if involving the content provider in the preprocessing chain, the huge effort spent for detecting the ball in the low resolution sequence is avoided. Several easy and robust ball detection mechanisms are already available for standard resolution schemes.

The movement of the audience is consistent with the camera motion. Moreover, the audience remains static: due to the extreme low resolution, small variations of the people sitting cannot be appreciated. Following these considerations, the Intra frame is encoded with an appropriate quality and acts as a reference for the following Inter encoded frames. A simple rigid translation of the audience is applied, only new elements appearing at the border of the scene are encoded with the standard procedure. This allows the encoder to save data rate that can be utilized for better encoding the remaining regions of the frame.

The current implementation still suffers of a *blinking* artefact, appearing each time the audience macroblocks are refreshed. The pan movement is appreciated as a translation, in reality the distance between the camera and the audience is changing. In order to solve this problem, the audience is regularly refreshed for offering an appropriate reference. The sudden variation of the audience appearance is noticed by the user as blinking, that distracts them from the game and, in some cases, has been reported as an annoying feature.

Chapter 5

Cross-Layer Optimization of Video Streaming

THE Cross Layer Design (CLD) paradigm comprises all the optimization approaches that come through the typical ISO/Open Systems Interconnection (OSI) model of layers with strict boundaries. The definition is, indeed, very generic and embraces a broad variety of works in the literature. It is not specified how many layers have to be involved in the optimization, nor which are the demands, at application or physical layer, that have to be optimized.

Basically, the scope of cross-layer design is the optimization of a given *utility* function. For a video service the utility function comprises, for instance, the user satisfaction combined with the cost for the operator. In this case, maximizing the utility function means maximizing the user's satisfaction minimizing the operator's cost. The optimization is constrained either by a specific minimum satisfactory quality or by the maximum amount of resources assigned to the video streaming.

An outstanding overview of cross-layer design concepts is offered in [86]. The CLD approaches are subdivided into two main groups depending whether a *top-down* or *bottom-up* approach has been applied. The top-down approach is compared to the problem of ordering custom made furniture at a carpenter shop. The application communicates downward its needs and the lower layers try to fulfill its wishes at the minimum cost possible. On the other hand, the *bottom-up* approach is compared to the case of a carpenter shop offering a selection of items the customer can choose from. In this case, the lower layers are communicating upward the set of parameters that can be offered to the application. The *bottom-up* approach aims at the maximization of the quality for a given cost, the *top-down* approach aims at the minimization of the cost for a given quality.

Cross-layer design calls for an additional amount of information shared between layers. As the number of parameters exchanged between layers has to be kept as small as possible, two groups of parameters have been defined in [87]: private and interfacing parameters. The private parameters, also called operating modes, remain confined to the layer they belong to and are not shared with the other layers. The interfacing parameters, or operating points, are visible to the other layers and are affected by the cross layer optimization.

A basic example for video streaming is offered in Figure 5.1. Assume that a given quality has to be assured at the application to the customers. The quality represents the operating mode, as its relevance is only appreciated at application level. For sake of simplicity, consider the quality to be

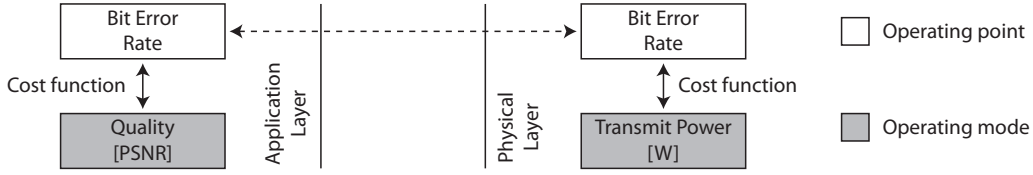


Figure 5.1: Operating modes and operating points.

only dependent on the BER. By means of a cost function, the quality can be mapped to the operating point, the desired BER. This information is then conveyed down, to the physical layer. At the physical layer, the BER may be mapped to an operating mode, such as the transmit power.

If two users are insisting on the same resources, say a limited transmit power, a *bottom-up* approach aims at the maximization of the average quality with the available transmit power. Utilizing a *top-bottom* approach with a specified average quality, the goal of the optimization is the minimization of the necessary transmit power.

In this chapter, two cross-layer investigations are described. In Section 5.1, the channel characteristics are exploited for optimizing the application of SP and SI frames in H.264/AVC streaming over UMTS networks. Following the *bottom-up* approach, the encoding parameters of the video are adapted to the transport block error traces measured in the live network. An application of the *top-down* approach is given in Section 5.2 considering the transmission of video streaming over HSDPA networks. In order to optimize the quality of experience at the application layer, the transmission parameters at the physical layer are adapted for better protecting the most important packets.

5.1 Cross Layer Application of SI/SP frames

SP and SI frames [24] are an error resilience feature introduced in H.264/AVC for limiting the temporal error propagation. As already pointed out in Section 2.2.1, the 3GPP specifications only support the H.264/AVC baseline. The SP and SI frames, however, are not included in the baseline profile, but in the extended profile.

In 3GPP PSS two mechanisms are considered for stopping the temporal error propagation. The first method simply consists of regular insertion of Intra encoded frames, by subdividing the sequence into GOPs. Intra encoded frames are not only useful when, in case of a scene change, the temporal prediction would not be effective, but also when the temporal prediction is impaired by an erroneous reference picture. This effect has already been described in Section 3.1. As one packet of F_2 has been incorrectly received, the corresponding slice of F_2 is incorrectly reconstructed or concealed. Although all the packets of F_3 have been correctly received, it is based upon a corrupted reference (F_2) for performing temporal prediction. This is valid for all the following Inter encoded frames until the end of the GOP.

The length of the temporal error propagation can be tuned modifying the size of the GOP. However, reducing the size of the GOP is not a straightforward solution as the size of the I frames is much bigger than the P frames. In Figure 5.2 the rate-distortion behavior for different GOP sizes is plotted considering different IP error rate probabilities.

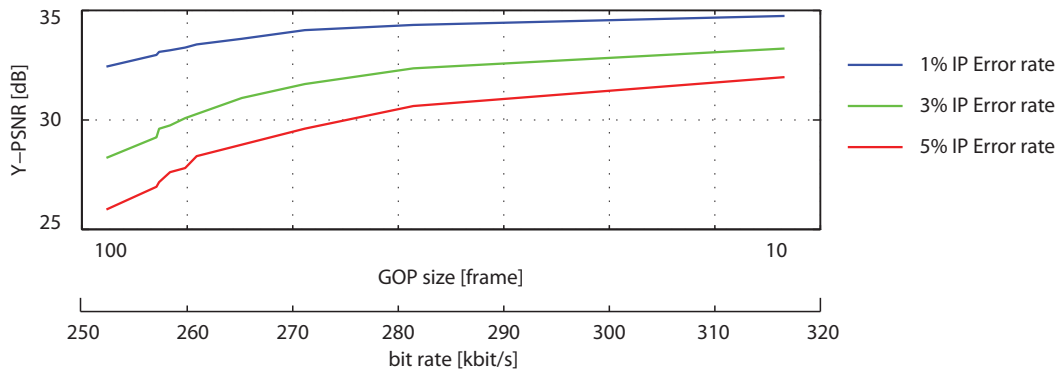


Figure 5.2: Rate distortion behavior for different **GOP** sizes.

The other method consists of retransmitting the corrupted packets. At the receiver side, a playout buffer is implemented as shown in Figure 5.3. Before beginning a streaming session, a *buffering* phase

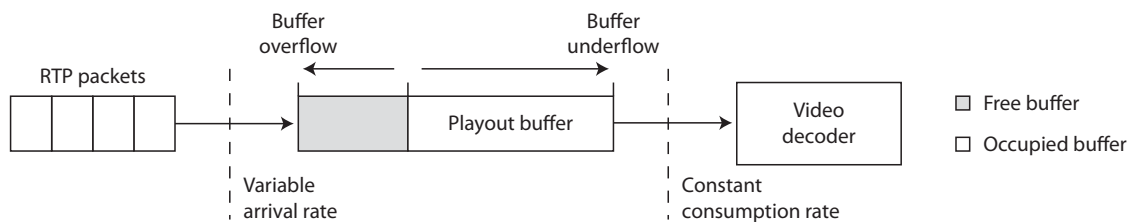


Figure 5.3: Playout buffer of the video decoder.

collects packets for fighting possible jitter. As the end-to-end transmission delay between the streaming server and the end user is not constant, the buffer helps handling packets arriving with higher delay, out-of-order or duplicated.

The size of the playout buffer has to be designed in such a way that buffer underflows and overflows are avoided. Buffer underflow occurs when the buffer becomes empty and there is no data to be played. In case of buffer underflow, the decoder usually freezes and a new buffering phase starts. This effect is particularly annoying for the user, as it interrupts the viewing experience. The other extreme case is the buffer overflow. It occurs in case the user equipment receives more data than it consumes. Buffer overflow is avoidable by establishing an end-to-end connection with appropriate parameters, such as the guaranteed and maximum bandwidth.

If the decoder side implements a playout buffer, damaged packets can be *retransmitted*. As discussed in the introduction of this chapter, the retransmission in **UMTS** occurs at transport block level, rather than **IP** or higher protocols, if the transmission occurs in **AM** mode. A maximum amount of retransmissions per **TB** is set. Since one **IP** packet consists of a collection of transport blocks, even a single damaged transport block forbids the recollection of the whole **IP** packet. If no basic cross layer information is shared between protocols, in case one **TB** is not correctly received, the whole **IP** packet is *lost*. Otherwise, the data link layer delivers to the higher layers the incomplete packet, that may be handled with the techniques described in Chapter 3.

Even assuming an infinite number of allowed **TB** retransmissions, there is a time limit that makes

the packet outdated. This limit is set by the current amount of data stored in the playout buffer. During the retransmission attempts, the decoder continues consuming the packets. If the IP packet containing damaged transport blocks is scheduled for being played, either it is forwarded to the higher layers as damaged or it is assumed to be lost and, therefore, concealed.

The temporal error propagation cannot be avoided relying solely on retransmission. As discussed before, the I frames stop the temporal error propagation. However, the position of the I frames is scheduled a priori during the encoding and cannot be adapted to the need of the decoder. Given a

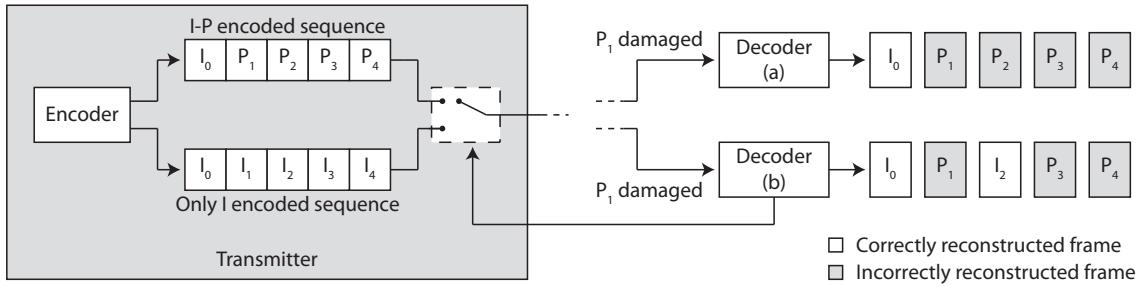


Figure 5.4: Implementation of feedback with only-I encoded sequence.

sequence to be transmitted, the streaming server has available two versions, as depicted in Fig. 5.4. The first consists of Intra and Inter encoded frames organized in GOPs and another consists only of Intra encoded frames. Assuming an available feedback channel at application level, the decoder informs the streaming server in case one packet has not been correctly decoded. The streaming server then reacts by transmitting an Intra encoded version of the following scheduled Inter encoded frame that stops the temporal error propagation. This approach, however, introduces a drift between the encoder and the decoder. The following Inter encoded pictures, in fact, relies on the reference pictures that have been considered at the encoder side. The motion compensation and, in particular, the coefficients have to be applied at the decoder side to the same reference employed at the encoder. Transmitting arbitrarily I frames in place of P frames stops the temporal error propagation caused by the error but generates a drift between the reference pictures at the encoder and at the decoder side.

This drift can be avoided exploiting SP and SI frames, two types of specially encoded frames introduced in H.264/AVC by Karczewicz and Kurceren [24]. They reflect the same prediction type of the P and I frame, respectively. A pair of SP and SI frames is encoded in such a way that the reconstructed image, when decoding an SP or SI frame, offers exactly the same prediction signal to the following pictures. This allows, at the decoder side, to substitute transparently an SP frame with an SI frame.

In Figure 5.5 it is shown how the approach based on the SP and SI frames works. At the encoder side, two sequences are generated. The first, labeled I-P-SP, consists of the standard I, P and of the SP frames, specially encoded P frames that regularly interleave the P frames. In the following, we will refer to S-Frame Distance (SFD) as the distance between two consecutive SP frames. Another equivalent encoded sequence, named I-P-SI, consists of I, P and SI frames. The SI frames occupy the same position of the SP frames in the I-P-SP. As the SP and the SI frames offer the same prediction signal to the following frames, the P frames of the I-P-SP version are identical to those of the I-P-SI

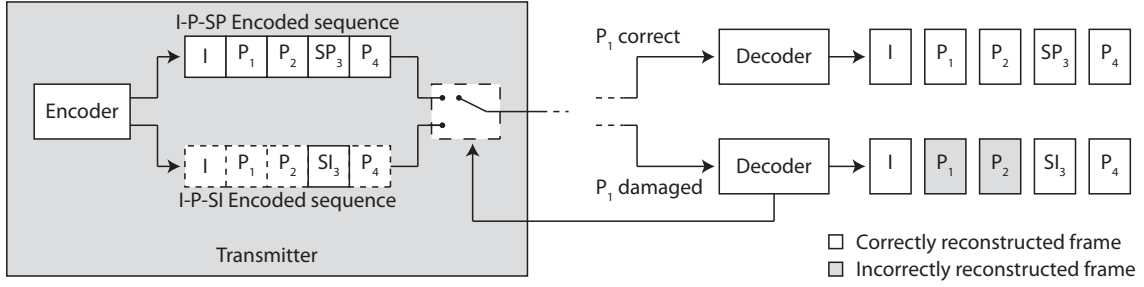


Figure 5.5: Implementation of SP and SI frames.

version. For this reason, it is easier to consider a complete sequence, the I-P-SP, and another one consisting only of the SI frames.

The transmitter starts sending the I-P-SP sequence. Assume now the frame P₁ to be incorrectly received and concealed, even though the packets associated to P₂ are correctly received, the frame is incorrectly reconstructed, since its source of prediction is corrupted. If a feedback channel between the transmitter and the receiver is available, the latter can signalize the damaged packet to the transmitter, which schedules an SI frame in place of the next scheduled SP frame. The special encoding mechanism of the SP and SI frames allows the decoder to reconstruct the very same picture when decoding the one or the other. In the following, a sequence consisting of I, P and either the SP or the SI frames (depending on the receiver's needs) is labeled I-P-S.

Therefore, the SP₃ or SI₃ offer exactly the same reference picture to the frame P₄. But whereas the quality of SP₃ depends on the correctness of P₂, being SI₃ spatially encoded, it stops the temporal error propagation.

In order to ensure an identical reconstructed image, the encoding and decoding of these special frames considerably differ from those of the I and P frames. The two most significant differences are: (i) The difference between the original and the predicted block is calculated in the Direct Cosine Transform (DCT) domain and (ii) The encoding consists of a two stages quantization-dequantization step, using two different Quantization Parameters (QP), QPSP1 and QPSP2. Further details on the implementation of the SP and SI frames in H.264/AVC can be found in [24, 88].

5.1.1 Channel Model

In Section 2.2.1 the protocol stack for 3GPP PSS services has been explained. From the application layer, a PDU, passing through RTP and UDP, is encapsulated into an IP packet and finally partitioned into transport blocks. In Figure 2.2, the protocol stack from application layer to the MAC layer has been described. In order to better understand possible problems arising from transmission errors, in the following the description will move closer to the physical layer, as shown in Figure 5.6.

Once the CRC has been calculated and attached to the transport block, the packet is ready to be sent to the physical layer. The transport block together with the CRC can be segmented or concatenated to fit the channel coder block size. The channel encoding is, usually, performed by a turbo encoder [89] with coding rate equal to 1/3. The coding rate is expressed as the formula $m/(m+n)$, where m is the number of information bits and n is the number of parity bits. By means of

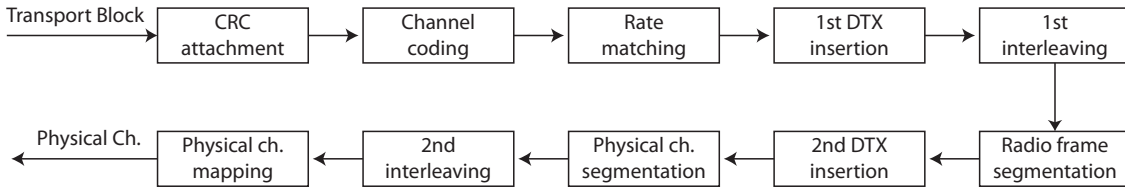


Figure 5.6: Transport block to physical layer mapping.

puncturing, the resulting rate is adapted to the available resources. Afterwards, a first Discontinuous Transmission (DTX) insertion indication is followed by a first interleaving over one coded block, radio frame segmentation and multiplexing of the various transport channels. The Coded Composite Transport Channel (CCTrCH) is then, after a second insertion of DTX indication, segmented into the appropriate physical channels. After a second interleaving over one radio frame, the data bits are mapped onto the correct physical channel.

In the following, a concrete significance will be given to the concept of *transmission errors*. As the data is transmitted onto the physical link, a transmission error is caused by poor channel conditions between the transmitter and the receiver. Assuming that the transmission occurs during a fading hole, the Signal to Noise Ratio (SNR) level at the receiver may be so low that, even after the turbo decoding, the payload of the transport block cannot be reconstructed.

In [23], Karner measured the TB error characteristics of the UMTS DCH of several Austrian mobile operators. The settings considered for such an investigation are compatible with those currently chosen for video streaming. As the bearer has been set to 384 kbit/s, one Transmission Time Interval (TTI) is 10 ms long and contains 12 TB. Each transport block is 336 bits long, comprising the 320 bits of the RLC payload and 16 bits of the RLC AM header.

For the measurements, a UDP data stream in DownLoad (DL) was sent from a PC over the UMTS network to a notebook attached to UMTS terminal as a modem via a USB connection. In Figure 5.7, a schematic illustration of the setup for the measurements in the live networks is given. In the

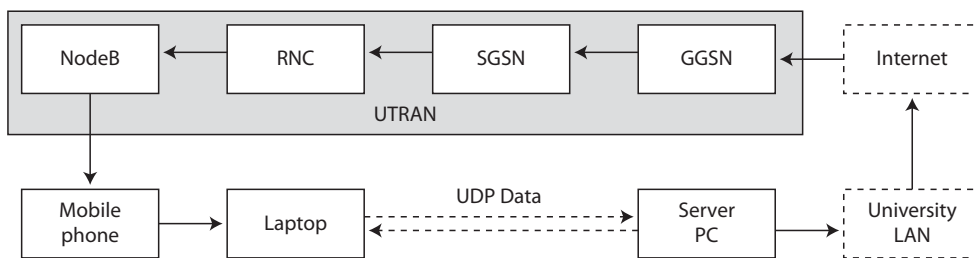


Figure 5.7: Measurement setup.

notebook, the traces were recorded using the TEMS investigation software by Ericsson. The CRC information of the received TBs were parsed from the traces and used for the analysis of the UMTS link error characteristics.

The result of the investigation showed that there is a high correlation between the correctness of the TBs belonging to the same TTI. In non-static scenarios, such as the user moving in a car or by train, the TBs were mostly either all correct or all damaged. In other words, chunks of data with

the size of one TTI have error probability P_{ERR} equal or higher than $1/384 = 2.6 \cdot 10^{-3}$, as each TB contains at least one error, other chunks of data have error probability equal to zero. Whether the considered bit error probabilities (around 10^{-5}) may, in average, be realistic, the assumption of uncorrelated error is unreasonable for mobile communications.

5.1.2 Simulation Setup

For simulation purposes, the performance of the application of **SI** and **SP** frames in H.264/**AVC** with respect to the classical I-P-P scheme has been compared. The transmission of streams encoded with both strategies has been simulated considering the channel realizations described in Section 5.1.1.

In order to generate the encoded sequences, the standard reference software **JM** [57] version 11.0 has been utilized. Although at the time this doctoral thesis has been written, the version 16.0 of the software was available, it has not been employed as the implementation of **SP** and **SI** frames in the newer versions is known to be not properly working (see <https://ipbt.hhi.de/mantis/>).

A football video sequence in Digital Versatile Disc (**DVD**) format consisting of around 1 000 frames has been encoded considering the typical configuration of unicast video streaming over third generation mobile networks. The most stringent constraint is represented by the bandwidth that is limited to 180 kbit/s. To match this restriction, the resolution of the video has been reduced to Quarter Video Graphic Array (**QVGA**) and the sequences have also been decimated in time, reducing the frame rate (**fr**) from 30 to 15 frames per second (**f/s**). The compression level is tuned by means of the quantization parameter, having impact on both the resulting quality and data rate.

For the I-P-P scheme, 17 different **GOP** sizes varying from 2 to 50 frames have been considered. The **GOP** size strongly affects data rate, as the encoded I frames are much bigger than the P frames. Because of the different encoding mechanism, even if the same **QP** for both the I frames and the P frames has been set, the quality of the Intra predicted frames was, in average, higher than the one of the Inter encoded frames. That makes the error free sequence quality a function of the **GOP** size, introducing an unnecessary complexity in the investigation. In order to make the quality of the I-P-P sequence not dependent on the **GOP** size a **QP** of 35 for the P frames and **QP** 37 for the I frames has been chosen.

For the I-P-S sequences, a single I frame was encoded at the beginning of the sequence. Two sets of sequences have been generated, the first containing I-P-**SP** frames and the second containing I-P-**SI** frames. As for the I-P-P scheme, the same 17 different **SFD** from 2 to 50 were considered. The quality and the size of the **SP** and **SI** frames strongly depends on the selected **QPSP1** and **QPSP2**¹. At this step, without knowing the probability of needing an **SI** frame, both **QPSPs** have been set to 33. The **QPSP** of the **SP/SI** frames has been set slightly smaller than the **QP** of the P frames in order to guarantee the same average quality to the reconstructed frames. As suggested in [90] the packet size of the I and P frames has been set equal to 750 bytes, offering a good compromise between header overhead and impact of a lost packet.

The **SP** and **SI** frames were not sliced into packets since the current implementation of the **JM** encoder does support a fixed size of **SP** and **SI** frames in bytes. When fixing the size of the slice in

¹In the following, when referring generically to the quantization parameters used for encoding the **SP** or **SI** frames (both **QPSP1** and **QPSP2**) the word **QPSP** will be used

bytes, the **SP** and **SI** slices contain a different amount of encoded macroblocks because of the different encoding mechanism. As the two encoding procedures are independent and unaware of another, a drift between **SP** and **SI** frames is introduced.

The transmission was simulated inserting the errors at transport block level. For the I-P-P scheme, a damaged packet was removed from the video file in case a transport blocks was marked as flawed. The standard concealment strategies of **JM** (copy-paste with motion compensation) were employed to recover the missing information.

As the standard **JM** decoder is suitable neither for handling errors at **TB** level nor for switching between **SP** and **SI** frames, for the I-P-S scheme the switching mechanism shown in Figure 5.8 has been implemented. The H.264/**AVC** encoder produces the sequences containing the I-P-**SP** and the

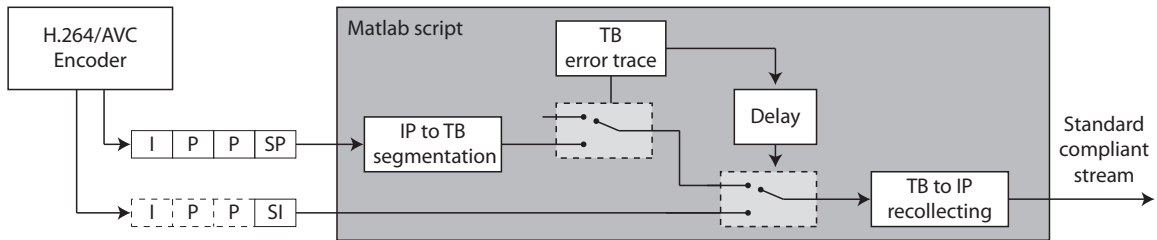


Figure 5.8: Implementation of the switching.

I-P-SI frames. As the transmission of the I-P-**SP** sequence is simulated, each **IP** packet is segmented into transport blocks, which are then compared with the measured error traces. In case a transport block is damaged, the whole **IP** packet is discarded and an error flag is raised. After a given delay, the script substitutes the next scheduled **SP** frames with the corresponding **SI** frame.

A delay time has to be considered in a realistic transmission scenario. It depends both on how the feedback information is transmitted and where the **SI** frames are stored. Two possible implementations are shown in Figure 5.9. The simplest implementation, in Figure 5.9(a), considers the **SI** frames stored

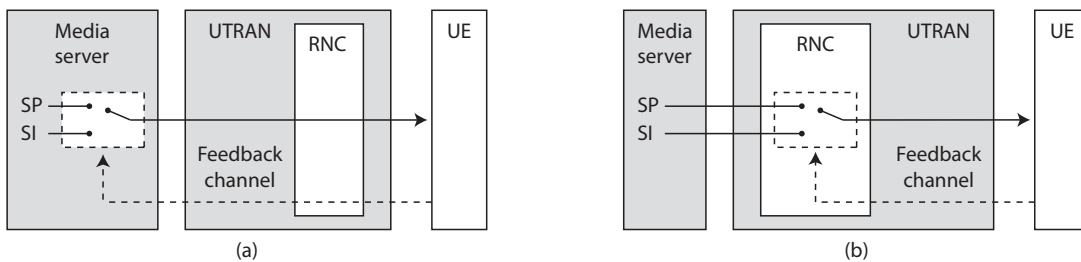


Figure 5.9: Implementation schemes in existing networks.

in the media server where the I-P-**SP** sequence is stored. The **SI** frames are sent in place of **SP** frames if the mobile terminal feedbacks errors in the link. A smarter solution, in Figure 5.9(b), considers all the **SI** frames sent from the media server to the Radio Network Controller (**RNC**) each time the appropriate **SP** frame is scheduled for transmission. This scheme is feasible assuming that no capacity issues have to be considered within the **UTRAN**. Assuming **RNC** improved capabilities, this network element decides which packets are further delivered to the Node-B without the need of forwarding the

feedback information to the media server. In order to cover a variety of technical implementations, different delay times varying from 0.1 to 2 s have been considered. As soon as an SI frame request has been received, the SI frame is sent in place of the next scheduled SP frame. 100 different realizations of each channel have been considered, randomly selecting the starting point of the error trace.

5.1.3 Results

As first outcome of the transmissions, the average temporal error propagation of the two schemes has been compared, as shown in Figure 5.10. The duration of the propagation depends strongly on

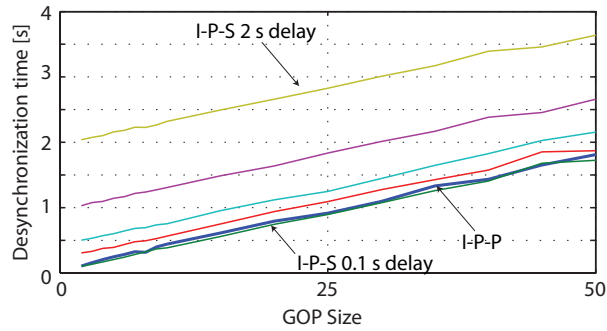


Figure 5.10: Average desynchronized decoding time.

the GOP size and on the feedback delay. The performance of the I-P-P scheme is almost identical to the I-P-S scheme with small delay times. Since, in average, the error occurs in the middle of the GOP, the Propagation Time (PT) increases linearly with increasing GOP or, equivalently S-Frame Distance, being the delay a fixed offset as

$$\text{I-P-P: PT} = \text{GOP}/(2 \cdot fr), \quad (5.1)$$

$$\text{I-P-S: PT} = \text{delay} + \text{SFD}/(2 \cdot fr). \quad (5.2)$$

The graph in Figure 5.11 shows the results of the simulations in terms of rate-distortion when

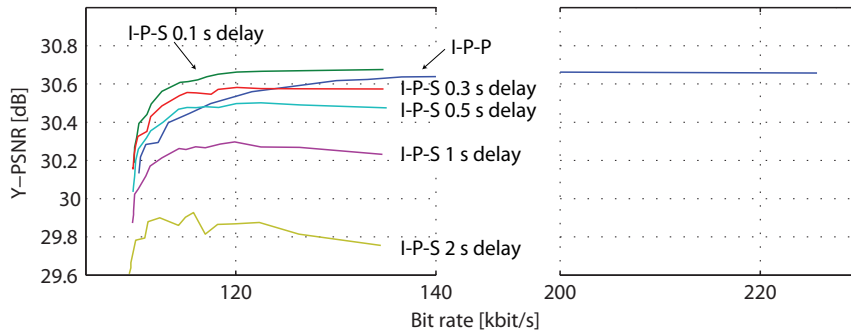


Figure 5.11: Rate-Distortion evaluation.

comparing the I-P-P scheme with the I-P-S. This first graph considers both QPSPs equal to 33. The I-P-SP scheme outperforms the I-P-P scheme only in case the considered delay is around 0.1 s. The performance of the I-P-S scheme decreases considerably with increasing delay and, surprisingly, the

quality decreases with decreasing **SFD**. This is due to the fact that, for high delays, the slightly lower quality of the S frames was dominant over shorter desynchronization times.

This approach, however, does neither take into consideration the channel characteristics nor the delay constraints. In order to optimize the rate-distortion behavior, the QPSP1 and QPSP2 values have to be adapted to the transmission characteristics. In Figure 5.12 the impact of the two QPSPs

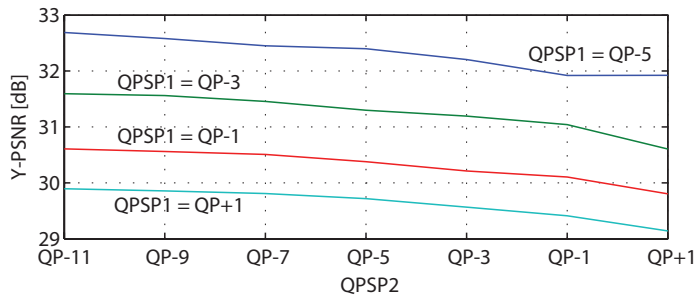


Figure 5.12: Quality of the **SP** and **SI** frames depending on the QPSPs.

on the resulting quality of the **SP** and **SI** frames is shown. The curves show how the quality is mainly driven by the QPSP1, with decreasing QPSP1 values associated to increasing frame quality.

The same experiment has been conducted for the rate of the **SP**, **SI** and P frames and the results are drawn in Figure 5.13. The curves in Figure 5.13(a) show the behavior of the size of the **SP** frames

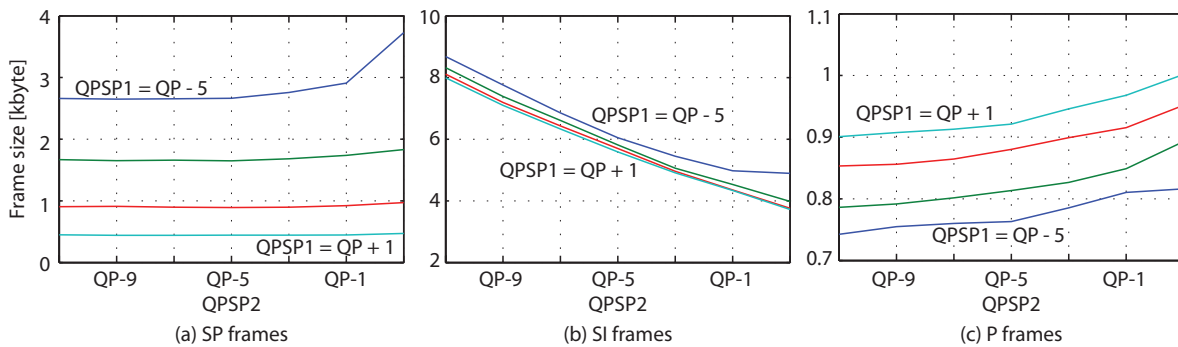


Figure 5.13: Size of the **SP**, **SI** and P frames depending on the QPSPs.

as a function of the QPSPs. As for the quality, the size is mainly depending on the QPSP1, and increasing values of QPSP2 are slightly increasing the **SP** frame size. The latter behavior contradicts the standard assumption of frame size decreasing with increasing **QP**. The size of the **SI** frames is drawn in Figure 5.13(b). The size of the **SI** strongly depends on the QPSP2 and it is only slightly dependent on the QPSP1. The quality of the **SP** and **SI** frames indirectly influences the size of the reconstructed P frames, as drawn in Figure 5.13(c). As for decreasing QPSPs the quality of the reconstructed **SP** and **SI** frames increases, they offer a better source of prediction for the following P frames. This decreases the amount of residuals that are necessary for reconstructing the picture.

The rate-distortion behavior has already been analytically solved by Setton and Girod in [90]. The optimal QPSP1 and QPSP2 values have been designed as a function of the **QP** of the P frames as well as the switching probability.

As the suitable **QP** of the **P** frames has been already defined, the probability of switching for the proposed channel measurements has been considered. The application of **SI** frames has been modeled as a state transition model with four states, as indicated in Figure 5.14(a). The transition probabilities

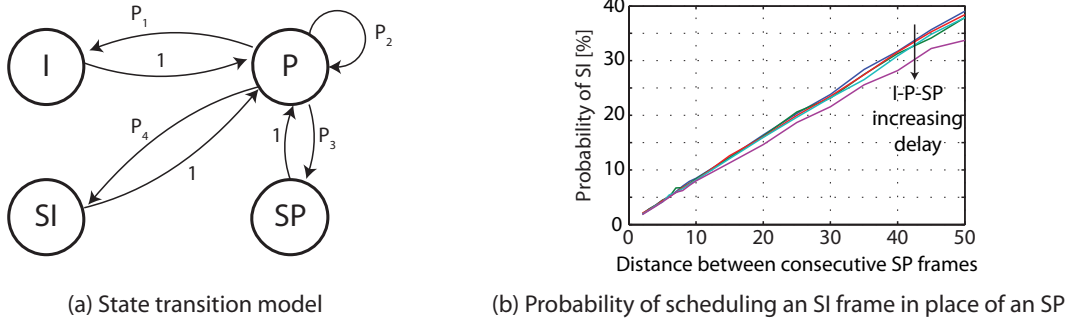


Figure 5.14: Modeling of the switching process.

P are defined as follows:

- $P_1 = f(\text{GOP}) = \frac{1}{\text{GOP}}$,
- $P_2 = f(\text{SFD}, \text{GOP}) = \frac{\text{GOP} - 1}{\text{GOP}} \cdot \frac{\text{SFD} - 1}{\text{SFD}}$,
- $P_3 = f(\text{SFD}, \text{GOP}, \text{delay}) = \frac{\text{GOP} - 1}{\text{GOP}} \cdot \frac{1}{\text{SFD}} \cdot P_{\text{SI}}$,
- $P_4 = f(\text{SFD}, \text{GOP}, \text{delay}) = \frac{\text{GOP} - 1}{\text{GOP}} \cdot \frac{1}{\text{SFD}} \cdot (1 - P_{\text{SI}})$.

They depend on the **GOP** size, on the S-Frame distance as well as on the probability of sending an **SI** frame in place of an **SP** frame (P_{SI}).

The probability of transmitting an **SI** frame in place of an **SP** frame has been measured and the result of the investigation is depicted in Figure 5.14(b). For distances smaller than 15 frames between S frames, the probability of scheduling an **SI** frame remains below 10%. For increasing **SD**, the probability increases, since the probability that an error has occurred between two consecutive **SP** frames becomes higher. It reaches the 40% for $\text{SD} = 50$. The probability is depending on the considered delay as well. With increasing delay, more **SI** frames are not sent because of feedback response issues, therefore the probability is slightly diminishing with increasing delay.

The graphs in Figure 5.15 compare the rate-distortion behaviour of the proposed schemes, when optimizing the QPSP2 as suggested in [90]. Major improvements are appreciated when using the refined I-P-S scheme. The quality, which previously saturated for increasing **SD**, is now increasing for shorter distance between consecutive S frames. I-P-S schemes with feedback delays around 0.5s still provide better performance than the classical I-P-P scheme.

The performed simulations bring us to the conclusion that the **SP** and **SI** frames are beneficial under specific conditions. As shown in Figure 5.15 their implementation is convenient as soon as the feedback delay remains smaller than 0.75s. This time is sufficient to convey the feedback information to the **RNC**, but it remains questionable if the whole core network till the Mobile Station (**MS**) may be crossed in this interval. Although outside the scope of this work, complexity and power consumption issues have to be considered in order to select the optimal **SD**.

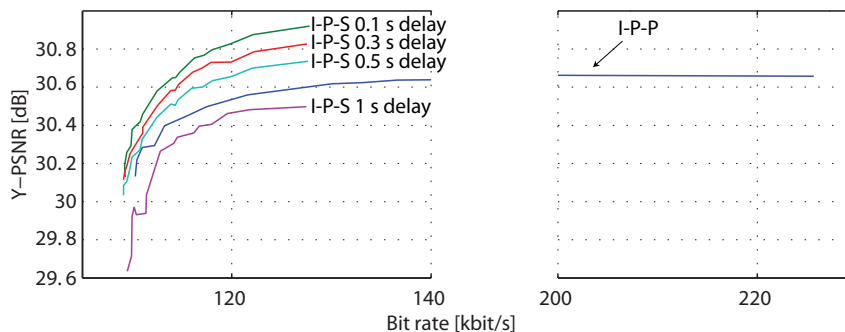


Figure 5.15: Rate-Distortion evaluation.

5.2 Optimization of Video Streaming over HSDPA Networks

This section presents an approach for optimizing video streaming in HSDPA networks. This work has been carried out together with Martin Wrulich and Philipp Svoboda [28, 29]. Mixing the expertise in our respective own field of interests, it has been possible to optimize the whole chain from the application layer to the physical one. Cross-layer design in standardized environment, such as the 3GPP world, usually requires additional information conveyed through the layers assuming channels not described in the standard. In this work, a fully standard compliant cross-layer optimization is proposed, exploiting the capabilities of the network elements of a standard HSDPA network (3GPP release 7).

This section is structured as follows: The traffic classes defined both in IP and 3GPP are described in Section 5.2.1. The implementation of the proposed mechanism for HSDPA networks is described in Section 5.2.2. As, at the time this doctoral thesis has been written, the method has not been tested in real networks, the HSDPA system level simulator used is presented in Section 5.2.3. The obtained results are shown in Section 5.2.4.

5.2.1 Traffic Classes in IP and 3G Networks

The IP world, standardized by the IETF, and the 3G world, standardized by 3GPP, separately defined their own traffic classification schemes.

In IP, the traffic class of a packet is signaled in the third byte of the IPv4 header, as shown in Figure 5.16. In [91] this byte has been reserved for signaling the ToS. The ToS indirectly indicates the quality of the desired service. As the IP has been designed for fixed networks, the network gateways decide how to route a specific packet depending on its ToS value.

The 8 bits associated to the ToS byte are defined in the following manner:

- **Bits 0–2:** Precedence.
- **Bit 3:** Delay. 0 = Normal Delay, 1 = Low Delay.
- **Bits 4:** Throughput. 0 = Normal Throughput, 1 = High Throughput.
- **Bits 5:** Reliability. 0 = Normal Reliability, 1 = High Reliability.
- **Bits 6–7:** Not specified, for future use.

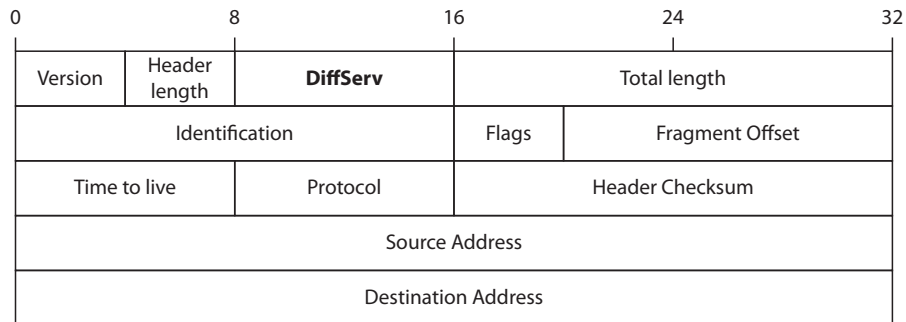


Figure 5.16: IPv4 header.

The IP Precedence field was assigned to various uses, including network control traffic, routing traffic, and various levels of privilege. In [91], the notion of Precedence was defined as “*An independent measure of the importance of this datagram.*” However, with increasing traffic demands and networks complexity, network gateway vendors started implementing more sophisticated queuing mechanisms considering ToS as one of the variables considered for filtering, together with IP source and destination addresses, UDP and TCP ports and other header fields.

In [92], Kilkki explains why a priority method based on ToS was not suitable for the evolving Internet network (already in 1999). Such a model was valid until the network was featuring reciprocity. In a reciprocal network, every user is allowed to send a large amount of important packets since (i) all the other users are allowed to do the same when necessary, (ii) no one is wasting resources by sending useless packets and (iii) many users decrease their sending rate when the capacity limits have been reached. Because of the heavily fragmentation of the current Internet structure, the three conditions, valid within the items of a group, cannot be extended to the whole network.

In [93], the ToS header byte has been assigned to the definition of Differentiated Services (DiffServ). The DiffServ Code Point (DSCP) fields occupies the first six bits of the third header byte, the last two remain unused or, in some specific application, reserved for Explicit Congestion Notification (ECN). DiffServ represents only a traffic classification and differentiation tool, as the implementation of the handling mechanism is left to the network provider. Network issues in a DiffServ architecture are handled in [94].

Even though up to 64 different DSCP are defined, [95] suggests, and does not require, a specified set of encoding.

- **Default Forwarding (DF)** [93, 96]. DSCP: 000000. It is basically the *best-effort* traffic category. It guarantees that the packets are accepted and can be configured for ensuring a given bandwidth. A single DSCP as well as a single queue is reserved for this category.
- **Assured Forwarding (AF)** [97]. DSCP: see Table 5.1. AF allows the network operator to bind the traffic produced by a given user to the amount stipulated in the contract. As the traffic is metered as it enters the network, it is marked according to the arrival rate. If the traffic exceeds the stipulated limit, the packet’s dropping rate of the user will increase in case of congestion. Four different AF classes have been defined, and each class is subdivided into three drop priorities. Within each class, the packets are dropped depending on the drop priority. Within different classes, balanced queue servicing ensure a higher rate of fairness with respect

	Class 1	Class 2	Class 3	Class 4
Low Drop	001010	010010	011010	100010
Medium Drop	001100	010100	011100	100100
High Drop	001110	010110	011110	100110

Table 5.1: AF DSCP.

to strict priority queuing.

- **Expedited Forwarding (EF)** [98]. **DSCP**: 101110. This traffic class is suitable for applications requiring low delay, low loss and low jitter. **EF** marked packets have to be preferred to any other class of packets.
- **Class Selector (CS)**. **DSCP**: XXX000. Provides backward compatibility with the ToS priority field as defined in [93].

The effectiveness of a DiffServ compliant network strongly depends on the reliability and the fairness of the marking point. The prioritization of **EF** packets, for instance, works properly if the amount of **EF** traffic does not exceed 30% of the capacity of the link. For this reason, a set of DiffServ elements belonging to a network, labeled *DiffServ domain* in Figure 5.17, need to be protected by

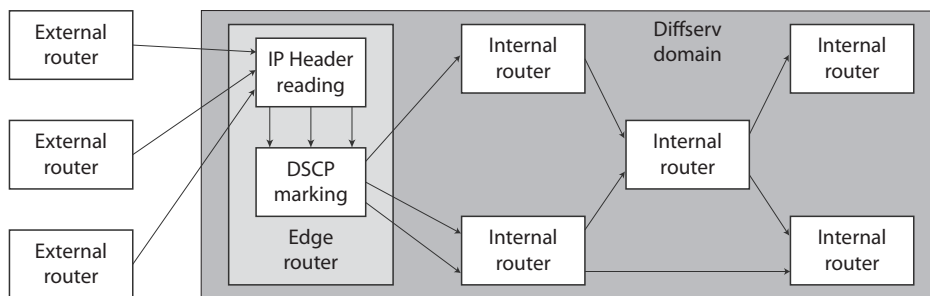


Figure 5.17: DiffServ domain.

DSCP misuses coming from the outside. At the border of the DiffServ domain, the edge routers implement admission control, policing and other mechanisms for controlling the traffic classification. The **DSCP** set outside the network rarely survive the edge, and new policies are specific of each DiffServ domain. A DiffServ aware router implements a Per Hop Behavior (**PHB**) that defines the forwarding options of a given **DSCP**.

Independently, the **3GPP** specified its own traffic classification mechanism in [99]. In **UMTS** the traffic classes are strictly related with QoS parameters, a concept that has been intentionally omitted in the definition of **IP** traffic classes. The QoS in **UMTS** is obtained by establishing a bearer service with specified characteristics and functionalities. Beside the control signaling and the user plane transport, the service bearer attributes also comprise, among others, maximum and guaranteed bit rate, traffic handling priority and transfer delay.

Four traffic classes have been defined in UMTS Rel. 99:

- **Conversational**: Is suitable for VoIP and videoconferencing applications. Involves a real time conversation between two (or more) end users. Stringent constraint on low delay and low jitter are specified.

- **Streaming:** Covers the one-way connection between an application server (video or audio) and a human end user. The focus is given on guaranteeing low jitter, since a fixed delay can be compensated by the playout buffer of the application.
- **Interactive:** Comprises human interaction with a remote server, such as web browsing. More effort is given on the preservation of the payload content rather than on the transmission time, as soon as it remains reasonable for the customer.
- **Background:** This category collects the background application. The end user is not expecting data within a certain time from, such as **FTP** downloading or email downloading. The time constraints are relaxed.

In the following, a brief overview of the literature discussing the possibility of implementing QoS by means of DiffServ in **UMTS** networks will be offered. Several scientific publications [26, 27, 100–102] discuss the best strategy how to map **UMTS** QoS classes to DiffServ traffic classes. The promising results show how an appropriate mapping privileging applications, such as **VoIP** and video streaming, are beneficial. However, these contributions just consider the definition of QoS on the core network and **UTRAN** network elements. In [100] the traffic bottleneck has been assumed to be between the Gateway GPRS Support Node (**GGSN**) and the Serving GPRS Support Node (**SGSN**). The characteristics of the wireless link between NodeB and user terminal has not been taken into consideration. In [103], the problem of differentiating packets within a single transmission flow is addressed. For a video telephony scenario, the authors discuss the possibility of handling differently the packets containing audio from those containing video, because of the respective traffic characterization and impact on the QoE. Still, the results are presented in terms of weighted delay and jitter.

5.2.2 Implementation of Secondary PDP Context in HSDPA Networks

An implementation of cross layer optimization of Video services over Institute of Electrical and Electronics Engineers (**IEEE**) 802.11e [104] wireless networks has been provided in [105, 106]. The **IEEE** 802.11e defines the QoS requirement for the **IEEE** 802.11 standard, known for being a best-effort service. The two works exploited the access classes defined in the **IEEE** 802.11e to give more priority to packets containing the most valuable payloads. However, the contribution offered in this doctoral thesis involves a much more complex access network, the **UTRAN**, and a scheduling algorithm, placed at the NodeB, which cannot be customized for specific applications.

In the following, the practical implementation of the cross-layer optimization is described. The proposed scheme involves almost all the protocol stack layers, as depicted in Figure 5.18. However, the

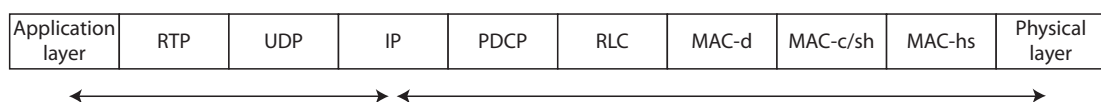


Figure 5.18: Cross layer optimization protocol stack.

mechanism has been subdivided into two steps. The first addresses the signalization of the importance of a video **NALU** from the application layer until the **IP** layer. This information is then exploited for guaranteeing a better QoS to the logical channel transmitting the most important packets.

The server side application, in this study the video encoder, already discriminates between packets having different importance. In this context, the term *importance* refers to the impact the packet has on the perceived quality at the user side, in case it is not received. As already pointed out, Intra encoded frame loss have a higher impact on the user side quality. They are not only used when, in case of scene change, the temporal error prediction would not be efficient, but also for stopping the temporal error propagation.

The importance of a video **NALU** is signaled in the two bit reserved for NALU Reference Indicator (**NRI**). This field is originally intended for signaling whether the encoded slice is referenced for temporal prediction or not. This information is not considered by the decoder but rather helps developing unequal error protection schemes at the transmitter side. Modifying this field, therefore, does not corrupt the structure of the video stream, which can still be decoded by a standard compliant H.264/**AVC** decoder.

As described in Section 2.2.1, the **NALU** is further encapsulated into the **RTP**, **UDP** and **IP** protocols, as shown in Figure 5.19. The DiffServ header byte is set according to three parameters:

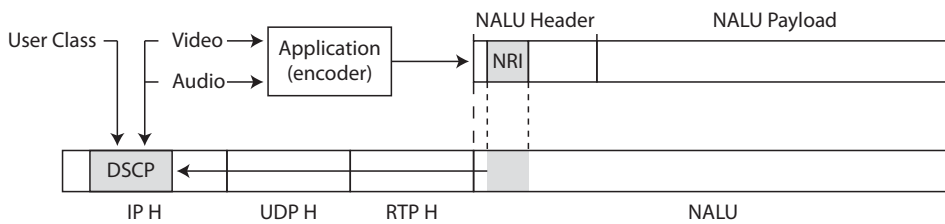


Figure 5.19: DSCP marking.

the application class, the user class and the importance of the packet as set by the encoder. The first two features are not strictly related to the cross-layer optimization. However, more priority is given to applications, such as video and/or audio encoders, whose services have stringent delay, jitter or bandwidth constraints. For video streaming applications, this operation can be performed by the video server itself, as it is either included in the DiffServ domain or it is considered to be a trusted source.

In the following, it will be explained how the DiffServ priority information is exploited for guaranteeing better transmission parameters to the most important packets, that is the second step of the cross layer optimization. In Figure 5.20, the protocol stack of **UMTS**, at the user plane, is shown. The end-to-end **IP** connection is established between the **GGSN**, in the core network, and the user equipment. The other network elements in between cannot understand the **IP** protocol. As shown in Section B-2, the **GGSN** is responsible of establishing a logical connection, named Packet Data Protocol (**PDP**) context, with the user terminal.

There exist two different categories of **PDP** contexts, primary and secondary. Once a primary **PDP** context has been activated, GPRS Tunneling Protocol (**GTP**) 1 in Figure 5.21, a parallel logical connection with the **GGSN** can be activated. The user terminal requests the activation of another primary **PDP** context or the attachment of a secondary **PDP** context to the already available primary **PDP** context. Each logical connection associated to the **PDP** context is established by means of **GTP**.

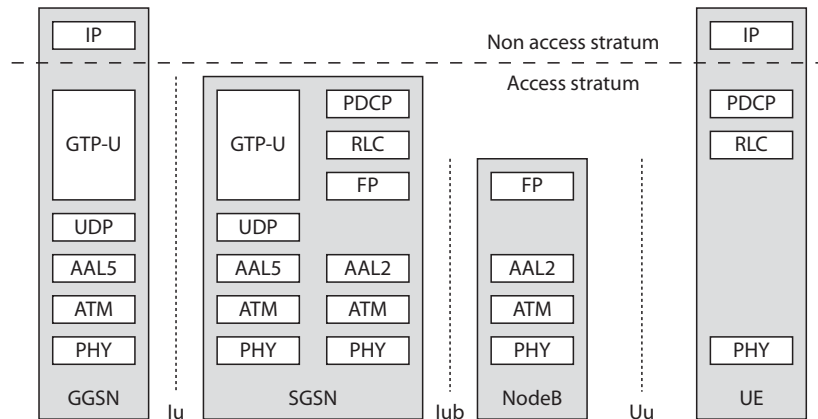


Figure 5.20: UMTS protocol stack (user plane).

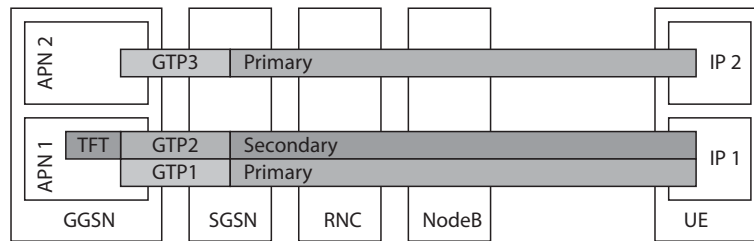


Figure 5.21: Example of two primaries and one secondary PDP context.

Another primary PDP context, GTP 3 in Figure 5.21, has its own IP address and it is usually associated to a different application. This is the case of Blackberry push email service in parallel to standard web browsing. The secondary PDP context can be attached to an already established primary PDP context, GTP 2 in Figure 5.21. Multiple secondary PDP contexts share the same IP address of the primary PDP context they are attached to, but can have different QoS settings. In order to subdivide the traffic produced by the same application into the two GTP tunnels, a Traffic Flow Template (TFT) filtering is performed. The filtering rules are based on one or more parameters of the IP header, such as source address, IP protocol number, destination port (range), source port (range), IPsec security parameter index and the type of service. This last attribute has been employed for the proposed cross layer optimization.

The complete implementation of the method is summarized in Figure 5.22. The DSCP of the

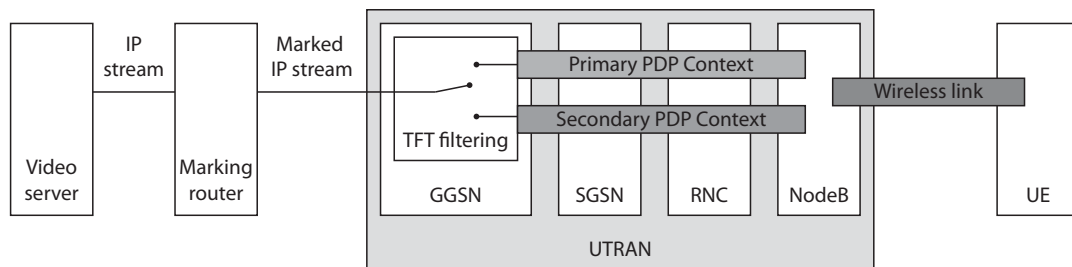


Figure 5.22: Block description of the proposed method.

packets streamed by the video server are marked by an edge router of the network, depending on the NRI field of the NALU header, on the user class and on the application class. Once the packets reach the GGSN, they are filtered into a primary and a secondary PDP context. The primary PDP context has higher QoS settings and it is used for transmitting the I frames. The P frames are transmitted in the secondary PDP context, with lower QoS settings. As the establishment of a secondary PDP context requires some seconds, it is advisable to transmit the unfiltered sequence with the better QoS and, once the filtering is active, downgrade the least important packets to the secondary PDP context.

The main advantage of the proposed method is that the cross layer optimization is totally transparent at the user side. As the two logical channels are both transmitting to the same IP address, the protocol stack of the user equipment is unaware of the performed filtering. The whole method is, therefore, standard compliant and can be implemented between any network and any mobile terminal capable of establishing a secondary PDP context.

5.2.3 HSDPA System Level Simulator

A system level simulator has been utilized in order to assess the performance of the proposed method. As the complexity of system level would increase when evaluating the radio link between the base station and the mobile terminal, a simple but accurate link model has been considered.

The simulator is capable of simulating classical HSDPA networks as well as the enhanced version utilizing MIMO for increased data rates. This work focuses on the classical single antenna HSDPA without the possibility for spatial multiplexing in the downlink [107].

The physical-layer modeling utilized in the system-level simulator accounts for MMSE equalization at the receiver side and accurately reproduces the inter-code interference in the multi-code operation of the shared downlink channel of HSDPA. The simulator is able to generate the cell deployment according to the desired configuration and deals with a large variety of user set-ups. A basic overview of the simulation methodology is depicted in Figure 5.23. The HSDPA cell deployment considered

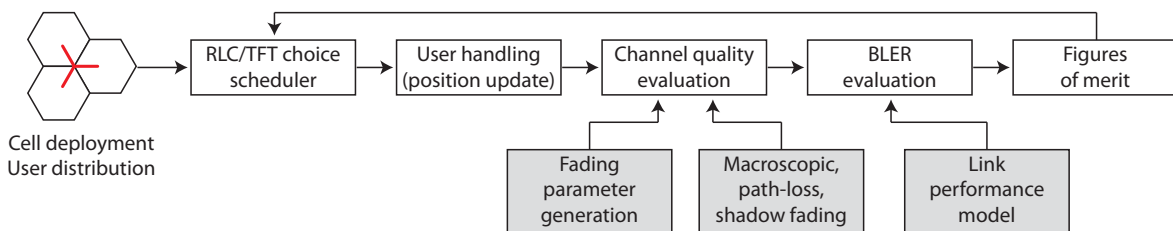


Figure 5.23: Overview of the system level simulator.

in the simulator consists of 19 three-sector sites, corresponding to the layout type 1 of [108]. The simulator allows for the power of the neighboring base stations to be controlled independently, such that the network to be simulated can be stripped down to seven three sector sites or even a single cell scenario. Different propagation models are available in the simulator, in this work the well known Walfish-Ikegami model [109] representing urban micro cell scenarios has been chosen. Radio link control as well as scheduling in the MAC-hs is simulated only for the target sector, thus keeping the computational effort manageable. This, however, requires the simulation to get along without handovers (because in the case of a handover, the associated algorithms—residing in the RNC—would

have to coordinate the radio link control of two sectors). In this work we set up the user mobility such that no handover will occur during the simulation of a video transmission.

The basic simulation procedure is as follows (see Figure 5.23): The first step of the simulation invokes the network generation, such as the cell deployment, and the user generation according to the selected User Equipment (UE) capability class together with their positioning. Also the fading parameters (describing the physical-layer) suitable for the scenario are loaded, the shadow fading traces are generated and the data necessary for the link-performance model (describing the decoding performance) is loaded. In the main simulation loop, according to the feedback of the UE in the target cell, the RLC and the MAC-hs scheduler decide upon the user to be served and the transmission settings of this transmission. After this decision an update of the user position takes place. With the position being known, the macro-scale pathloss and the effective antenna gain can be calculated. The Signal to Interference and Noise Ratio (SINR) in the current transmission is then evaluated and consequently the correctness of the received packet is determined according to the link-performance model. The user feedback is then formed of the ACKnowledgment (ACK)/Not ACKnowledgment (NACK) report and the Channel Quality Indicator (CQI) for the current transmission evaluated pursuant to the mapping of the UE capability class. At the end of the simulation time, the resulting data is collected and statistically evaluated.

5.2.4 Results

As the establishment of a primary and a secondary PDP context cannot be implemented employing the system level simulator, a strategy for adapting the transmission setting to the packet priority has been proposed.

As a first step, the interfaces between the video codec and the simulator have been designed, as shown in Figure 5.24. The video encoder, both in RTP and Annex B output mode, produces a

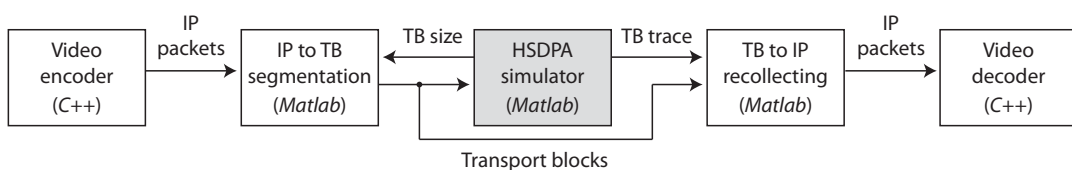


Figure 5.24: Interfaces between the video codec and the simulator.

stream that can be easily segmented into NALUs. After encapsulating the NALU into the appropriate protocol headers, the IP packets are queued for transmission. As the system level simulator produces traces for transport blocks, the IP packets have to be segmented. The size of the transport block in HSDPA is not fixed a priori, as in UMTS, but it is rather adaptively decided by the scheduler depending on the reported user CQI, as explained in Section B-4.

The output of the simulator is the error trace of the transmitted transport blocks. In case a transport block has not been successfully transmitted, the whole IP packet the TB belongs to is discarded. The input of the video decoder is the reassembled sequence, not comprising the IP packets that have not been entirely received. The JM software is already able to recognize missing packets, as an RTP header field, sequence number, can be exploited for sorting out the received packets. In case

a packet has not been received, the corresponding macroblocks are concealed.

This implementation still cannot handle packets with different priorities. In order to simulate an unequal packet handling, the mapping of **CQI** to modulation scheme and transport block size has been modified. The standard mapping, as specified in [110], is trained in order to guarantee a transport block error rate equal to 10%. In case of high reported **CQIs**, the size of the transport block is increased and better performing modulation schemes, till 64 Quadrature Amplitude Modulation (**QAM**), are considered. For smaller **CQIs**, smaller transport blocks are transmitted considering conservative modulation schemes, such as 4 **QAM**.

In order to better protect the most important packets, the standard scheduler is modified introducing a remapping block, as shown in Figure 5.25(a). The proposed approach relies on a remapping

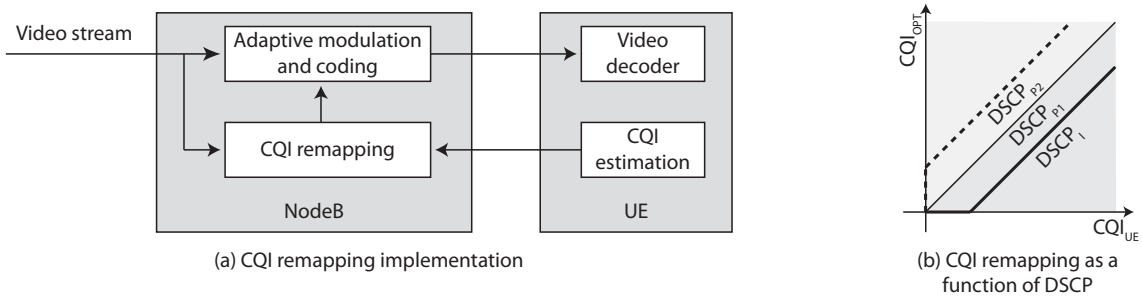


Figure 5.25: Implementation of the remapping scheme.

function Φ that, given as input the reported user **CQI**, CQI_{UE} , and the **DSCP** of the packet to be transmitted, generates the optimal **CQI**, CQI_{OPT} , for better protecting the most valuable payloads. The function Φ is defined as follows:

$$CQI_{OPT} = \begin{cases} CQI_{UE} - 1, & \text{I slice} \\ CQI_{UE}, & \text{P slice} \in [1, GOP_{size} - 4] \\ CQI_{UE} + 1, & \text{P slice} \in (GOP_{size} - 4, GOP_{size}] \end{cases} \quad (5.3)$$

Graphically, this behavior is described in Figure 5.25(b). For the most important packets, the conservative remapping forces the scheduler to choose a smaller transport block size and a less efficient modulation scheme that results in an average transport block error rate smaller than 10%. In Equation (5.3), an aggressive remapping is considered as well. As the conservative remapping reduces the throughput of the **TTIs** where the I frames are transmitted, the overall cell throughput is reduced as well. In order to offer a fair comparison, the cell throughput is balanced by a more aggressive remapping performed for the packets that have the smaller impact on the perceived quality. The behavior between missing packets and their position in the **GOP** is drawn in Figure 5.26. Different sequences have been encoded with a fixed **GOP** size equal to 65 frames. Depending on the sequence characteristic, the impact of the position of the missing packet is different. For more static sequences, like Hall, the quality degradation lies around 2.5 dB. Since the same QP has been used for all the sequences, those involving more movement, like container, are characterized by smaller quality. As the concealment does not perform as good as in the static case, the quality ranges between 28 dB for errors introduced at the beginning of the **GOP** and 32 for those introduced at the end.

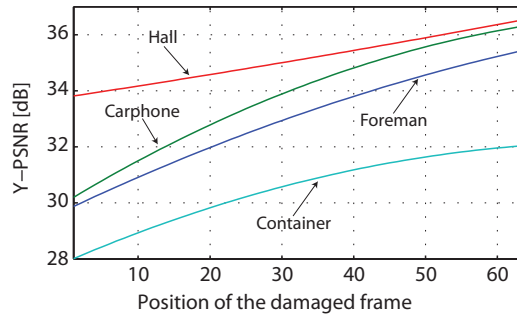


Figure 5.26: Quality as a function of the position of the corrupted frame.

As a general observation, the last frames of a **GOP** do not impair the objective quality as much as the first ones, as the temporal error propagation is much shorter. As the size of an I frame is comparable to the size of four P frames, in order to offer a fair comparison in terms of cell throughput, a more aggressive remapping has been chosen for the last frames of the **GOP**.

The results are shown comparing the proposed Cross-Layer Content Aware (**CLCA**) scheme and the standard Round Robin (**RR**) algorithm. In Figure 5.27(a) the transport block error probabilities of the two schemes are shown. The transport block error probability of the **RR** scheme matches the expected 10%. The results of the **CLCA** approach have been separately shown for I frames and for P frames. By means of **CQI** remapping, the transport block error probability of the I frames has been reduced by a factor four and lies around 2.7%. The average **TB** error probability for the P frames lies slightly over 10%, as the aggressive remapping performed for the last frames of the **GOP** increases the transport error probability.

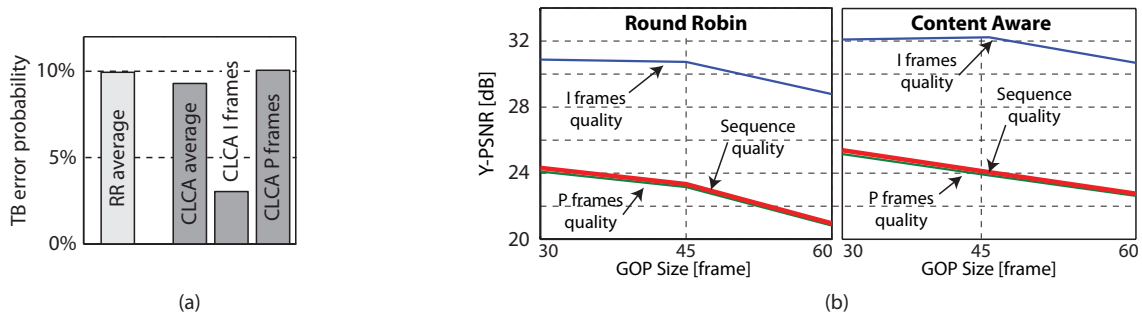


Figure 5.27: Transport block error probability and quality comparison without retransmission.

As the whole cross layer optimization aims at the enhancement of the perceived video quality, the results are shown in terms of **Y-PSNR**. The curves shown in Figure 5.27(b) show the resulting quality in case no retransmission is allowed.

As expected, the lower error probability of the **TB** containing the I frames is beneficial in terms of video quality. The average I frames quality measured when considering the proposed method is over 1 dB higher than the quality obtained with the **RR**. Although no positive remapping is performed for the P frames, their average quality benefits for the cross-layer optimization. The quality of a P frame is also dependent on the quality of the reference, because of the already discussed temporal error propagation. As the I frames stop the temporal error propagation, by decreasing the amount of

corrupted I frames, the probability of a P frame of having a reliable source of prediction is increased.

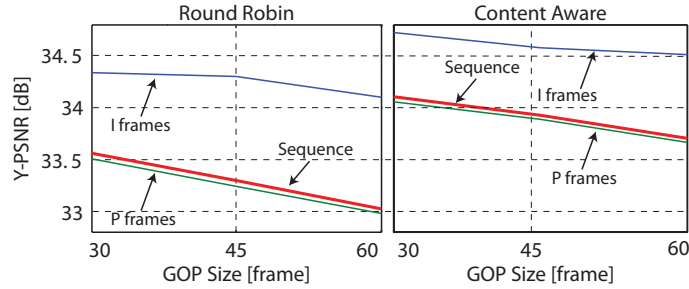


Figure 5.28: Quality comparison with retransmission.

Similar results are obtained when considering the possibility of retransmission, as shown in Figure 5.28. The Hybrid Automatic Retransmission reQuest (HARQ) does not decrease the transport block error probability on the single transmission, but rather, by combining corrupted versions of the same transport block, increases the net probability of having available an error free version of the transport block at the receiver side. For this reason, the improvement brought by the cross-layer optimization is limited to 0.6 dB.

5.3 Conclusions and Self Criticism

Cross-layer techniques for video streaming represent an interesting research field where the delivery of an encoded sequence is observed without the limitation of the classical ISO/OSI protocol approach. The layers can communicate and share information with distant layers aiming at the optimization of a specific *utility function* valid for the whole process. In this chapter the utility function, the perceived video quality, has been optimized considering either the channel characteristics of the UMTS DCH or the access network of the HSDPA.

The first proposed approach consists of the application of SP and SI frames to H.264/AVC encoded sequences. These two kinds of frame are not allowed in the 3GPP PSS standard, as they cause additional computational complexity at the decoder side. The current mobile devices, however, are equipped with processors and batteries capable of more complex tasks, if compared with the mobile phone capabilities considered at the time of the standardization of 3GPP PSS. The new wireless cellular network standards, like LTE, can realistically include more complex H.264/AVC profiles, such as the main profile.

The benefits arising by the application of the SP and SI frames, as shown in Section 5.1.3, should also be compared with an increasing level of complexity, when considering devices with limited capabilities. Such an investigation, however, cannot be performed utilizing a development software such as JM. Another issue regards the usage of packet retransmission in place of SP and SI frames. Retransmitting the damaged packet is convenient when limiting the observation to the net data rate. Retransmission, however, has to cope for strict delay constraints set by the playout buffer. In case the round trip time is higher than the time span covered by the playout buffer, it can happen that the retransmitted packet arrives too late. SP and SI frames, despite their increase in complexity and average data rate, relax the delay constraints. The decoding resynchronization occurs by replacing

packets *in the future*, whereas retransmission requires the replacement of a specific damaged packet. Hybrid broadcasting schemes can be therefore implemented by means of **SP** and **SI** frames. One single sequence, consisting of **I**, **P** and **SP** frames is transmitted to all the users. In case one user receives damaged packets, the appropriate **SI** frame can be transmitted exploiting a unicast parallel channel.

The second cross-layer proposed approach consists on the exploitation of different logical channels for transmitting the packets to the user. This idea has been realized by assigning a priority index to each packet in order to guarantee a better handling to the most important payloads. The signalization of the priority from the application to the **IP** layer is straightforward. However, there is no direct way for signalizing the priority of the **PDU** in the lower layers and the nodeB is not able to handle packet belonging to the same stream differently. It has been therefore decided to move the decision back to the **GGSN**, where the **IP** protocol is understood. An appropriate filtering of the packet allows the allocation of each packet to the appropriate logical channel. The results shown have been obtained utilizing a simulator.

In order to validate the simulation, the proposed scheme has been implemented by means of general purposes PCs attached to the reference cell of mobilkom austria. The media server consists of a PC running **VLC** as a streaming server. **VLC** produces a unicast video stream in MPEG Transport Stream (**MPEG-TS**) format. A script developed in Java² is responsible of modifying the **IP** headers of the packets streamed to the receiver. The script acts like a socket, intercepting the packets and modifying their ToS byte according to the **NRI** field set in the encoded sequence by a customized version of the **JM** encoder. This PC is then attached to the **GGSN** of the mobilkom austria's reference cell. At the receiver side, an **HSDPA** modem is placed into a *shielding box* of the reference cell. The modem is attached via USB to the receiver PC, where **VLC** is acting as a streaming client. The receiver PC is in charge of establishing the **PDP** contexts, as explained in Appendix B-2. The tested modems possess two COM ports. The first COM port is utilized for communicate the AT commands that (i) establish the primary **PDP** context and (ii) define the **TFT** filtering. Once the primary **PDP** context has been activated, the secondary **PDP** context can be attached. As the first COM port is now occupied by the primary **PDP** context, the secondary **PDP** context is activated through the second COM port. When realizing the transmission, the data transmitted through the secondary **PDP** context are not received at the receiver side. In order to further investigate this problem, the ICMP protocol (ping messages) data has been filtered and transmitted through the secondary **PDP** context. As all the interfaces of the reference cell have been monitored, the Internet Control Message Protocol (**ICMP**) packets were correctly filtered and transmitted by the antenna through the Iu interface. As the two available COM ports were already occupied by the **PDP** contexts, there was no possibility of monitoring the modem interface with tools such as Wireshark. The most realistic assumption is that, although written in the specifications, the modems were not fully supporting the secondary **PDP** context. Even though they were able to correctly establish the secondary **PDP** context, the modems were probably not able to recollect the data transmitted through the second logical channel.

²For the realization of the script, Elena Recas de Buen has to be thanked

Chapter 6

Conclusions

THIS doctoral thesis dealt with the optimization of video services under specific transmission environments, that are the UMTS and HSDPA wireless networks. For performing meaningful investigations, three factors have been taken into account when performing the optimization: the characteristics of the wireless channel, the way human users evaluate the content as well as the topology of the access network. The structure of the thesis reflects this subdivision, even though the interactions between elements of different nature have to be considered in place of strict boundaries.

6.1 Results of this Thesis

As a first result, it has been demonstrated how the detection of errors within corrupted packets is a tool for enhancing the quality of video streaming over an error prone transmission channel. If compared to the classical approach relying on the correctness of the whole application datagram, as signaled by the UDP checksum, the exploitation of the correct payload segment limits the concealment to the corrupted part. As the position of the error is not known a priori, different error detection mechanisms have been implemented and analyzed. They all show significant improvements if compared to the standard handling mechanism. As the proposed more sophisticated mechanisms may increase the computational complexity and/or introduce distortion on the encoded stream, costs and benefits of each proposal have been discussed. A final discussion has been reserved to the suitability of the considered channel model. An uncorrelated error model, such as the binary symmetric channel, does not reflect the real behavior of the wireless channel. As real measurements showed a strong correlation between occurring errors, an error model based on the correctness of the transport blocks is more appropriate. This information can be extracted by the transport block's parity bits and conveyed up to the application layer for performing error detection.

As the video application is consumed by human users, the optimization of a specific video service, soccer video streaming, has been performed considering subjective aspects. If the video codec cannot discriminate between different regions of a single frame, human users react differently to specific elements of the image. Active elements, such as the players and the ball (i), are much more meaningful for the understanding of the game if compared to background items, such as the field (ii) and the

audience (iii). The rate distribution assigned to them does not reflect the subjective importance of each region: the encoding of the audience requires more than the half of the total data rate associated to a frame. To optimize the encoding, the three regions have been identified in each frame and differently handled by the video encoder. The data rate associated to the encoded audience region, has been minimized by means of a rigid translation of a reference picture, reflecting the camera movement. Significant improvements have been measured by means of a subjective distortion measure, the MOS, where human users were asked to rate the displayed video sequences.

The last investigation concerns the cross-layer optimization of video services considering the specific characteristics of the transmission channel and the access network. A first bottom-up approach deals with the application of special encoded frames (SP and SI) for reducing the temporal error propagation, caused by frames utilizing corrupted references for temporal prediction. As those special frames are sent on demand, benefits have been shown for feedback delays remaining smaller than the typical UMTS round trip times.

A top-down optimization mechanisms has been implemented for signaling the importance of a application payload as close as possible to the physical layer of an HSDPA network. The network elements of HSDPA are currently not discriminating between packets belonging to a single application. However, different packets belonging to the same application have different impact on the user experience. The signalization has been split in two steps. As first, the information about the importance of one application payload, as signaled by the application itself, is stored into the DiffServ field of the IP header. In UMTS, the end-to-end IP connection is set up between the GGSN and the user terminal, the network elements in between do not implement the IP protocol. At the GGSN a logical connection, PDP context, with a given QoS class is established with the mobile terminal. At the same time, a parallel logical connection with a different QoS class can be set up. By means of TFT filtering based on the content of the IP header, a packet is transmitted over the one or the other logical connection. The overall quality of the video stream sent over the HSDPA network has been enhanced when better handling the packets containing the encoded I frames.

6.2 Open Points and future Work

In this thesis three main standards have been considered: UMTS and its evolution HSDPA as wireless communication standards and H.264/AVC as video codec. At the time this thesis has be finalized, the HSDPA and the H.264/AVC have reached their maturity. The respective standardization organization are currently working on their successors.

In December 2009 the Swedish mobile operator TeliaSonera launched the first public LTE mobile service. LTE is a standard defined by the 3GPP as a successor of UMTS and HSDPA and represent the last step of the bridge leading from the third to the fourth generation of mobile telecommunication systems. Major changes in the air interface and in the access network let LTE offer data rates up to 100 Mb/s in downlink and 50 Mb/s in unplug. For this reason, the data rate limit considered for video applications in this thesis, around 220 kb/s, has to be considered outdated. Moreover, the smart phones currently available in the market support resolutions higher than CIF. Most of the

entertainment and business phones are nowadays full-touch or with fold-out keyboard, allowing for Half VGA (HVGA) or VGA resolution.

At the same time, the JVT has started the definition of the H.265/High-efficiency Video Coding (HVC). As in the case of H.264/AVC, the (promised) improvements are obtained by further improving the current video codec functionalities, without any relevant breakthrough. A lot of speculations have been made in the last years about the replacement of H.264/AVC in mobile environment. In a first moment, the Scalable Video Coding (SVC) was considered as a solution for utilizing a single content for serving devices with different capabilities. However, the scalability is paid in terms of increasing complexity and decreasing coding efficiency. Benefits and costs of scalable video coding are investigated in [111]. Afterward, a second paradigm, the Distributed Video Coding (DVC), has been considered. Despite an initial enthusiasm in the scientific community, the theoretical benefits have, up to now, never been verified by any real implementation.

Realistically, future research topics should investigate the transmission of H.264/AVC streams over LTE. LTE is advertised as a full-IP network, this may enables simpler and finer optimization mechanisms for better handling packets whose importance is signalized at IP level. As the users have shown a limited interest on video streamed to their mobile phones, the definition of video stream should include more generic video services, such as Youtube, accessed by a laptop equipped with a datacard. Moreover, further subjective investigation should focus on users reaction to transmission errors, as the Y-PSNR has already shown major flaws.

Appendix A

H.264/AVC

H.264 Advanced Video Coding (**AVC**) is, at the time of writing of this thesis, the state of the art and best performing video codec for commercial applications. The standard has been jointly defined by two standard organizations, the Moving Picture Expert Group (**MPEG**) of the International Standard Organisation (**ISO**) and the Video Coding Expert Group (**VCEG**) of the International Telecommunication Unit standardization sector (**ITU-T**). The partnership is commonly known as Joint Video Team (**JVT**). In the past, the two standard organizations defined their own video standards.

The **ITU-T** developed video standards oriented to telecommunication services. In 1990, H.261 [112] was released for video transmissions over Integrated Services Digital Network (**ISDN**) lines. In the same year the **ISO** was finalizing the development of the MPEG-1 [113], a video coding standard suitable for storage applications as well as for high data rate broadcasting. In 1994 an enhancement of the previous standards was jointly defined under the name of MPEG-2 (Part 2) [114] for **ISO** and H.262 [115] for **ITU-T**, increasing the coding efficiency and the flexibility. The MPEG-2/H.262 is still currently employed for Digital Video Broadcasting Satellite (**DVB-S**) transmissions and for Digital Versatile Disc (**DVD**) storage. One year later, in 1995, the first version of the **ITU-T** H.263 [49] has been standardized. H.263 was designed for low bit rate videoconferencing and was the video codec formerly utilized for the encoding Flash Video files. In 1998, **ISO** finalized the MPEG-4 (Part 2) [50]. The video standard was designed for ensuring flexibility, serving applications ranging from TeleVision (**TV**) broadcasting to low data rate internet transmission. Different coding libraries based on the MPEG-4 standard became common for storage of video content to be shared in Peer to Peer (**P2P**) networks, such as Xvid and DivX.

Finally, in 2003, the **ITU-T** H.264 [52] or **ISO** MPEG-4 (Part 10) [50] was jointly developed and by the end of 2009 it is still the state of the art video codec. As for its predecessors, the codec does not provide any major breakthrough, but fulfills to achieve efficient coding rate and flexibility by further enhancing the features of the previous standards. The specifications defined in the standard refer to the decoding process. Therefore, whereas the decoder is bounded by the specifications of the standard, complete freedom has been left to the implementation of the encoder, as soon as it produces a stream understandable by the decoder.

The standard is organized in *profiles* and *levels*. The profiles and levels are conformance points for allowing compatibility between different devices having defined software or hardware capabilities. A profile defines a specific set of allowed coding tools. Different profiles are specified in the standard:

baseline, main, extended and, recently, high definition profiles. The Third Generation Partnership Project (3GPP) defined the H.264/AVC as optional codec in its baseline profile. The baseline profile does not allow computational demanding decoding features. The levels place a set of constraints on key parameters of the bitstream, such as the allowed bit rate, resolution as well as the decoding speed rate.

Although the encoder is not covered by the standard, it is conceptually advised to describe the functions of the video encoder. Its functionalities can be subdivided into the Video Coding Layer (VCL) and the Network Abstraction Layer (NAL). The former comprises the encoding related features whereas the latter guarantees *network friendliness* to the encoded stream.

A-1 Network Abstraction Layer

H.264/AVC addresses a wide range of video applications. It can be utilized for storage applications, such as Blu Ray discs or generic ISO MP4, as well as for transmission applications, such as real time or non real time streaming and terrestrial or satellite broadcasting. The NAL functionalities aim at the formatting of the encoded stream, as produced by the VCL, for matching the requirements of the different applications.

The encoded stream consists of a collection of Network Abstraction Layer Unit (NALU)s. Without loss of generality, for streaming services the NALU is the payload of the Real Time Protocol (RTP) packet, although for some specific applications several NALUs are encapsulated into a single RTP packet. Each NALU consists of an one-byte header and the payload, a chunk of the encoded stream produced by the VCL. The NAL header contains three fields: (i) a forbidden bit, that has to be zero, (ii) a NAL reference indicator (two bits), signaling whether the slice is referenced for prediction (see Section A-2) and (iii) the NAL unit type, indicating the type of the NALU contained in the payload.

The NALUs are classified into VCL and non-VCL NAL units. VCL NALUs contain information directly related to the encoded content of the picture whereas the non-VLC NALUs contain more generic information, such as parameters sets and supplemental enhanced information.

The standard defines two types of parameter sets. The Sequence Parameter Set (SPS) contains information regarding the whole sequence, such as the resolution (the size in pixel of each frame), the frame rate (the number of pictures displayed each second), the profile and the level selected by the encoder. The Picture Parameter Set (PPS) contains information shared by a set of pictures, possibly the whole sequence. It contains, for instance, the number of slices contained in the picture and how are they distributed, the size of the reference picture buffer and the default quantization parameter. The hierarchical structure of the parameter set and VCL NALUs is sketched in Figure A.1. Each

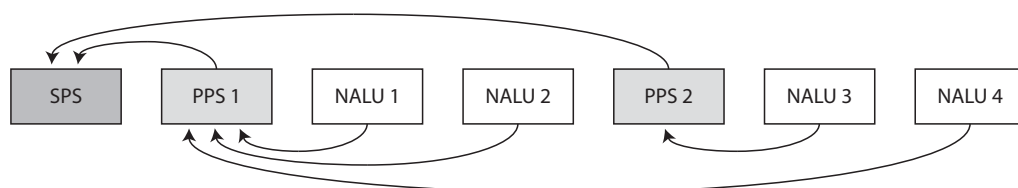


Figure A.1: Hierarchical dependency of SPS, PPS and NALU.

encoded picture references the **PPS** containing the appropriate parameter set. Each **PPS**, in turn, references an **SPS**. Usually an encoded sequence contains just a single **SPS** and a single **PPS** that, for broadcast applications, are sent regularly, allowing new users to randomly access the sequence. In Chapter 4, an application exploiting multiple **PPS** has been presented.

A-2 Video Coding Layer

Similarly to the video codecs previously mentioned, the H.264/**AVC** is a *hybrid block based video codec*. The term *block based* refers to the fact that each picture is subdivided in square blocks called *macroblocks*. The encoder exploits the temporal correlation between consecutive pictures and the spatial correlation within a single picture. In this extent the encoder is also defined hybrid.

A sequence picture can be encoded as a single entity defined *frame* or subdivided in even and odd rows to be displayed in an interlaced manner. Each subsampled picture is labeled as *field*, top and bottom, respectively. In this work the interlaced mode is not considered, therefore the words picture and frame have to be considered as synonymous.

As the Human Visual System (**HVS**) receives the brightness information differently from the chrominance stimulus, in video coding the Luminance Chrominance-blue Chrominance-red (**YCbCr**) is preferred to the common Red Green Blue (**RGB**) color space. Moreover, since the **HVS** is more sensible to the luminance than chrominance, the chrominance is subsampled. The most common sampling is the 4:2:0, four pixels are described using 4 levels of luminance and two of chrominance as shown in Figure A.2. In other words each frame is split into one luminance frame with the same

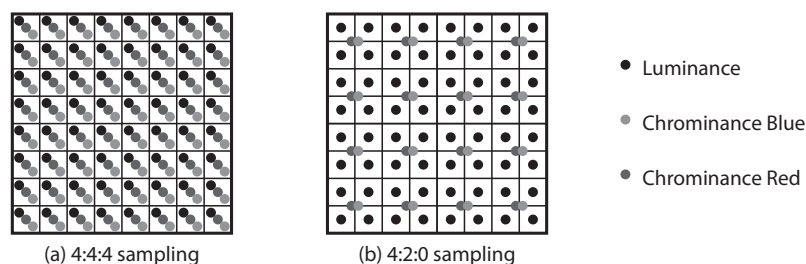


Figure A.2: Distribution of luminance and chrominance.

resolution and two frames having half the width and half the height of the original resolution. In H.264/**AVC** the macroblocks have a fixed size of 16×16 pixels in the luminance frame and 4×4 in the chrominance frames.

A set of macroblocks belonging to a single frame are defined as a *picture slice*. A picture slice is *self contained*, since it can be interpreted without the necessity of accessing information belonging to other slices. Usually a slice contains a given number of macroblocks sorted in a raster scan order. In H.264/**AVC**, however, the Flexible Macroblock Ordering (**FMO**) allows more freedom when partitioning the picture into slices. For more details about **FMO**, refer to Section 4.2.2.

The standard defines two types of encoding, one exploiting the spatial correlation between macroblocks belonging to the same picture, Intra Predicted frame (**I**) encoding, and one exploiting the temporal correlation between macroblocks belonging to consecutive pictures, Inter Predicted frame (**P**)

encoding. A picture can be either Intra or Inter encoded. In the former case, all the macroblocks are Intra encoded, in the latter both encoding modes are allowed. The first frame of a sequence is Intra encoded, as no reference for temporal prediction is available. As the temporal prediction is much more effective than the spatial one, a sequence is segmented into Group Of Pictures (GOPs). A GOP consists of an Intra encoded picture and all the following Inter encoded pictures until the following Intra encoded picture, that represents the first picture of the following GOP.

In both cases the encoded process can be summarized as follows: for each macroblock its best prediction is searched either in the same picture or in the previous reconstructed frames, the element-wise difference block is built and further processed by means of a two-dimensional (vertical and horizontal) Discrete Cosine Transformation (DCT) and quantized. The block is then raster scanned and entropy encoded.

All the parameters are encoded trying to minimize the number of bits associated to the resulting codeword. Since the codewords have no fixed length, on this extent the coding style is defined as Variable Length Coding (VLC). Moreover, the encoding process is also Context Adaptive (CA), as it is influenced by the previously encoded elements. In H.264/AVC, two entropy coding modes are defined, the Context Adaptive Variable Length Coding (CAVLC) and the Context Adaptive Binary Arithmetic Coding (CABAC). Since only the former is allowed in the baseline profile, the following discussion describes the CAVLC encoding process.

A-2.1 Intra Frame Prediction

In Intra predicted frames, each macroblock is predicted using the already encoded neighboring macroblocks as reference. The luminance macroblock can be subdivided into sixteen 4×4 subblocks or encoded as a whole. In Figure A.3 the prediction modes in case the macroblock is subdivided into

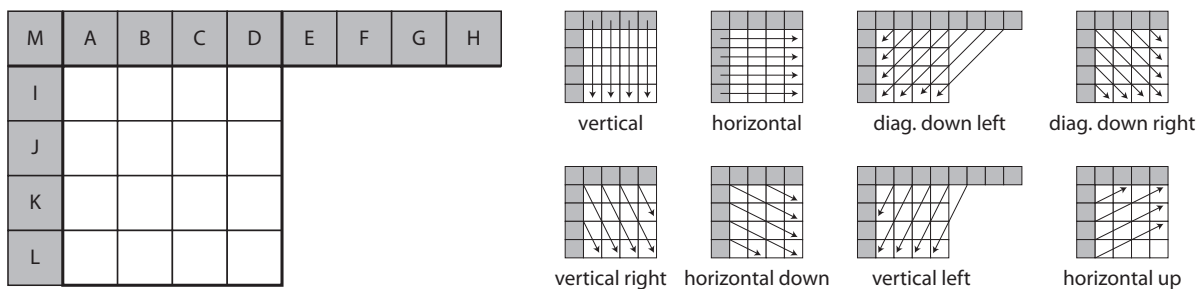


Figure A.3: Intra prediction mode.

subblocks is shown. The rightmost column (pixels I to L) and bottom line (A to H) of the neighboring macroblocks are exploited to build a prediction exploiting one of the nine available modes. In order to avoid drifts between the encoder and the decoder, the reconstructed blocks (containing the quantization loss) are used for prediction. Each mode defines a directional interpolation of the neighboring pixels, in the horizontal, vertical and diagonal direction.

The pixels of the neighboring macroblocks can be utilized as a source of prediction only if they belong to the same slice as the macroblock to be predicted. Therefore, only a subset of the nine prediction modes can be usually exploited. If none of the neighboring macroblocks is available for

prediction, a Direct Current (**DC**) mode is allowed by the standard. An equivalent set of Intra prediction modes is also defined for the whole 16×16 pixels macroblock. For predicting the whole macroblock, only four modes are defined: vertical, horizontal, diagonal and **DC**. Similarly, the two 8×8 chroma blocks are predicted exploiting the neighboring already encoded chroma blocks. The same four prediction modes defined for the 16×16 luma prediction modes are allowed.

The Intra frame prediction is encoded signaling how the macroblock is subdivided and the best selection mode chosen for each block.

A-2.2 Inter Frame Prediction

The Inter prediction mode exploits the strong correlation between consecutive pictures. A macroblock can be predicted as a whole item or can be subdivided iteratively into smaller blocks, as drawn in Figure A.4. The best prediction is searched either only in the previous pictures (P inter prediction)

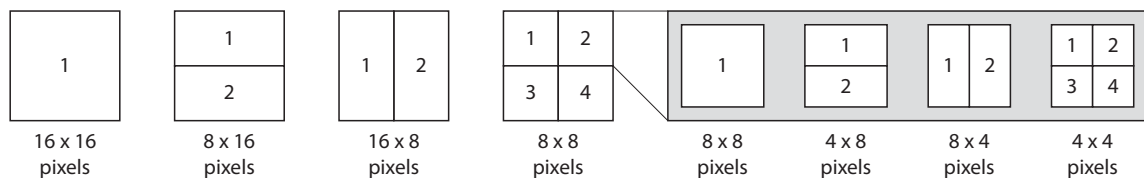


Figure A.4: Intra prediction mode macroblock subdivisions.

or also in the following pictures (B inter prediction mode). In both cases the already encoded version of the frames is used as a reference. This avoids drift at the decoder side, where the original frame is not available.

The best prediction is searched for following a spiral path starting from the position of the macroblock to be encoded. The size of the region where the search is performed depends on time and computational constraints: the bigger the search region, the more expensive the search process.

The position of the best prediction with respect to the current block is signaled by means of *motion vectors*. In H.264/**AVC** a motion vector is predicted for each block considering the motion vectors already defined for the neighboring blocks. Therefore, rather than the absolute components, the difference between the best and its prediction is encoded.

H.264/**AVC** defines the so called *sub-pixel motion compensation*, the luminance motion vectors have the resolution of a quarter of a pixel. Since the luminance and the chrominance blocks still have integer resolution, a prediction is generated by interpolating the luminance (or chrominance) samples. The interpolation is performed using a six-tap Finite Impulse Response (**FIR**) filter, each interpolated sample consists of the weighted sum of six integer samples. The Inter frame prediction is encoded signaling the subblock subdivision, and, for each subblock, the position of the reference picture in the buffer as well as the relative motion vector components are encoded.

A-2.3 Transform and Quantization

Once the best prediction has been chosen, independently from the prediction type and the macroblock subdivision, each predicted 4×4 block is elementwise subtracted from the corresponding original block, obtaining the 16 elements *residuals* matrix **R**.

The matrix \mathbf{R} is then further processed by means of an horizontal and vertical modified DCT obtaining the matrix $\mathbf{T} = \text{DCT}_V(\text{DCT}_H(\mathbf{R}))$. The modified DCT defined in H.264/AVC is an integer transformation calculated with summations and shifts (multiplication free). The inverse transformation is also defined, avoiding mismatches between encoder and decoder. In case the macroblock has been Intra encoded and has not been subdivided, a modified transformation is defined in order to further decorrelate the luminance DC components. A similar transformation process is defined for transforming the 2×2 chrominance blocks associated to each 4×4 luminance block.

The last step before the entropy encoding is the quantization of the transformed residuals. The H.264/AVC makes use of a scalar quantizer that has been specifically designed to avoid divisions and floating point algebra. Each element of the quantized transformed residual matrix \mathbf{Q} can be expressed as $Q_{i,j} = \text{round}(T_{i,j}/Q_{\text{step}})$, where Q_{step} depends on the Quantization Parameter (QP). The smaller the QP, the smaller the Q_{step} and, therefore, the less coefficients are quantized to zero. The elements of the matrix \mathbf{Q} are in the following called *coefficients*¹.

After the transformation and quantization, the encoder performs the reconstruction of the block by means of inverse quantization, inverse transformation and summing up the residual with the *predicted* block. This step is necessary to avoid drifts between the reference used by the encoder and by the decoder. For both encoding strategies, it is necessary that the prediction information is provided by the reconstructed rather than by the original block. As the Intra prediction uses as reference a block belonging to the current picture, the update process has to be performed on-the-fly. For the Inter encoded blocks, the reconstructed blocks are stored in a picture buffer, relaxing the time constraints.

A drawback of the block-based encoders is the blockiness effect in the reconstructed pictures. In H.264/AVC an effective deblocking filter has been defined. It compares the values of the reconstructed pixels at the border of two neighboring 4×4 blocks. If the difference is higher than a given threshold depending on the current QP, the borders are filtered.

A-3 Entropy Coding in H.264/AVC

H.264/AVC makes extensive use of entropy encoding mechanisms. Entropy encoding is a lossless data compression mechanisms aiming at the reduction of the average codeword length, by linking the length of each codeword to the probability of occurrence of the respective value. The word *value* refers to the decimal representation of the element to be binary encoded whereas *codeword* refers to its binary equivalent. Since the length of the codeword is not fixed a priori, entropy coding is a variable length coding. For this reason, in order to avoid synchronization words between consecutive codewords, the entropy encoding must generate a *prefix code*. A code system is defined *prefix code* if none of the valid codewords is a prefix of another valid codeword. This property is requested for correctly understand the boundaries of the variable length codewords.

Assume now an alphabet of values $V = \{v_1, v_2, \dots, v_n\}$ each with a known probability of occurrence

¹In some of the author's publications and in some literature, the elements of the matrix \mathbf{Q} are called residuals. However, in this doctoral thesis, the word coefficient is preferred to be consistent with the official literature and standard nomenclature.

$p(v_1) \leq p(v_2) \leq \dots \leq p(v_n)$. Defining the codeword alphabet as $C(V) = \{c_1, c_2, \dots, c_n\}$, where c_1 is the binary representation of the value v_1 , the coding efficiency is maximized of the average when the codeword length

$$\bar{l} = \sum_{i=1}^n p(v_i) \cdot l_i \quad (\text{A.1})$$

is minimized, being l_i the length of the codeword c_i . Shannon defined the information content of a symbol v_i as

$$h(v_i) = \log \left(\frac{1}{p(v_i)} \right). \quad (\text{A.2})$$

The entropy of a source producing the symbols V can be defined as

$$H(V) = \sum_{i=1}^n p(v_i) \cdot \log \left(\frac{1}{p(v_i)} \right) = - \sum_{i=1}^n p(v_i) \cdot \log(p(v_i)). \quad (\text{A.3})$$

The entropy of the source defines also the shortest average codeword length that can be achievable by entropy coding. With the assumption of known value's probabilities, this result is achieved with a Huffman code.

Unfortunately, for video coding application, the probabilities of occurrence are not known a priori, but some assumptions can be made. Small values are assumed to occur more frequently than larger ones, and, therefore, are encoded choosing shorter codewords. If larger values occur more frequently, the original values are remapped in such a way that this property is still in force. In the following, the three entropy encoding mechanisms used in H.264/AVC are presented.

- **Tabled Codewords.** H.264/AVC defines several look-up tables for different encoded elements, where a given value is associated to each codeword. The relation between the codeword and the resulting value is not unique but it usually depends on other information available at the decoder side. Since the scope of the entropy encoding is to associate shorter codewords to symbols with higher probability, the tables are accordingly designed, modifying the association map depending on the value of the previously decoded parameters.
- **Exp-Golomb codewords.** Exp-Golomb is a class of parametric universal code. In H.264/AVC the parameter k of the exp-Golomb code is fixed to zero, resulting in a structure similar to the Elias- γ [116] code. Whereas the Elias- γ code can only encode values larger than zero, the exp-Golomb can encode the value zero as well. The generic exp-Golomb encoded codeword has the following structure:

$$\underbrace{0_1 \dots 0_M}_M 1 \underbrace{b_1 \dots b_M}_M. \quad (\text{A.4})$$

The length of the codeword is variable and equal to $2M + 1$. The first M zeros and the first one represent the *prefix* of the codeword. The last M bits are the *info* field of the codeword. The decoding of an exp-Golomb codeword takes place by obtaining the associated `codeNum` field:

$$\text{codeNum} = 2^M - 1 + \text{info}. \quad (\text{A.5})$$

From `codeNum`, the encoded value is finally reconstructed, depending on the mapping type (unsigned, signed and truncated).

- **CAVLC level.** The **CAVLC** is, in a narrower sense, employed to encode the transformed quantized residuals. As discussed before, the entropy coding aims the minimization of the codeword length depending on the probability of the associated value. Therefore, in H.264/AVC seven different VLC- N routines have been defined. The parameter N is chosen in the range $[0,6]$ depending on the values of the previously decoded residuals. Under the assumption that small values are associated to high frequency components, the first residual is encoded with the VLC-0 routine. A VLC-0 codeword has the following structure:

$$\underbrace{0_1 \dots 0_M}_M 1.$$

The parameter M contains both the value as well as the sign of the encoded value. For the following residuals, an appropriate VLC- N routine is chosen depending if the previous decoded element l_{i-1} is larger than a given threshold, as described in the Table A.1:

VLC- N	Threshold
0	0
1	3
2	6
3	12
4	24
5	48
6	NA

Table A.1: Thresholds for VLC- N routines.

The codewords have the following generic structure:

$$\underbrace{0_1 \dots 0_M}_M 1 \underbrace{i_1 \dots i_{N-1}}_{N-1} s.$$

Similarly to the exp-Golomb encoded words, the first $M + 1$ bits represent the prefix of the codeword. The info field contains a number of bits equal to the parameter N minus one. An additional bit signalizes the sign of the value.

A-4 Elements encoded into a NALU

In Sections A-2.1 and A-2.2 it has been described how the Intra and Inter macroblock prediction are performed and which encoded parameters are stored in the NALUs. In the following, the remaining encoded items are presented and discussed. Once the best prediction has been found, each 4×4 residual matrix is transformed and quantized. These two operations are standard and do not produce any additional code overhead.

The first element to be coded after the prediction is the Coded Block Pattern (CBP). The CBP signalizes which of the four 8×8 luminance block and which chrominance blocks contains, at least,

one non-zero coefficient. The coefficients of each 4×4 subblock are raster scanned and sorted as shown in Figure A.5, from the highest to the lowest frequency. The *coefficient token* signalsizes the number

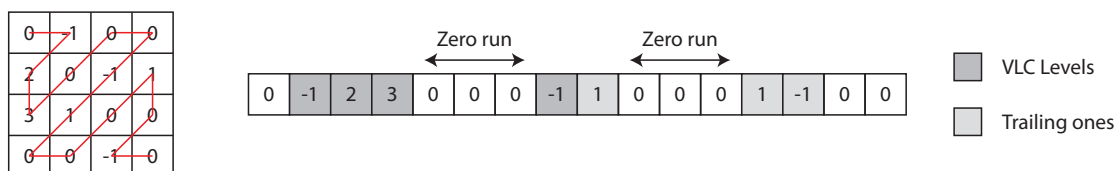


Figure A.5: Sorting of coefficients.

of non-zero coefficients and the number of *trailing ones*. The number of trailing ones is the number of consecutive coefficients having absolute value equal to one, starting from the end of the scanned coefficients sequence. If the first non-zero coefficient has an absolute value different than one, the block has no trailing ones, even though some of them appear in lower frequency. The number of trailing ones ranges between zero and three, if a block has four or more consecutive ones, the first three are labeled as trailing and the following are normal non-zero coefficients. The number of non-zero coefficients varies between 0 and 16. The codeword associated to the coefficient token is stored on a table and its entries depend on the number of non-zero coefficients of the neighboring macroblocks. In this extent, many encoded coefficients in H.264/AVC are *context adaptive* since the codeword associated to the value to be encoded depends on the previously encoded values.

Once the number of the non-zero coefficients and trailing ones is known, the values are encoded into codewords. Since the absolute value of the trailing ones is fixed, their sign is encoded with a single bit. The other non-zero coefficients are encoded as shown in Section A-3. Subsequently, the total number of zeros between the DC position and the first non-zero coefficient is stored in the Total Zeros parameter. The Zero Run signalizes how many zero coefficients are present between each non-zero coefficient and the following one. A similar approach is followed for encoding the coefficients of the chrominance blocks.

A-5 Decoding

The decoding process follows backwards the instruction contained in the encoded sequence as produced by the encoder. First, the best prediction is built depending on the block prediction mode. Since the reconstructed block has been used as a reference at the encoder side, if no errors have occurred during the transmission, the decoded block is the same as the reconstructed one.

The prediction has to be corrected by means of residuals. As the coefficients are available at the decoder side, they have to be inversely quantized and inversely transformed. The so obtained residuals are summed up to the prediction and the reconstructed picture is smoothed by means of a deblocking filter.

A-6 JM Reference Software

In this doctoral thesis, it has been made extensive use of the Joint Model (JM) [57] reference software. The JM is a freely available software implementing the functionalities of a standard compliant H.264/AVC decoder and of a H.264/AVC encoder producing standard compliant encoded sequences. The software has been developed by a team of researcher actively involved in the standardization group of H.264/AVC and managed by Karsten Suehring and Alexis Tourapis. The code has been entirely written in American National Standards Institute (ANSI) C and does not aim at optimizing the scripts but rather on making the code understandable by other researchers. The flexibility and simplicity are paid in terms of computational complexity. Using a standard general purpose computer running Windows XP, encoding a P frame in Common Intermediate Format (CIF) resolution takes more than one second.

The code is offered as *best effort*, some software bugs are already known whereas a bug tracker allows the users to report bugs or missing features. The implementation of the ideas described in this thesis required often massive modification of the reference software, possibly including MATLAB pre- and post-processing.

Appendix B

UMTS and HSPA Overview

The reliable transmission of data packets over wireless cellular network represents one of the most challenging task already arose in the late nineties. The earlier Global System for Mobile communications (GSM) standard [117] already allowed for a wireless data service. As the connection was circuit switched, a given amount of resources were allocated for establishing a wireless communication channel between one user and one base station. This approach does not suit the bursty characteristics of Internet traffic, causing an suboptimal usage of the air interface as well as an unfair billing system, as the connection is charged depending on the time and not on the volume.

General packet radio service (GPRS) [118] was a bearer system introduced in GSM for improving the wireless access to data networks. On the one hand, GPRS simplified the network access, from several second to less than one second, on the other hand it increased the data rate, from 9.6 kbit/s to an Integrated Services Digital Network (ISDN) like connection speed.

Universal Mobile Telecommunication System (UMTS) [119] was introduced in the late ninety's to cover the increasing need of data bandwidth for mobile users. The peak data rate per user has been increased up to 386 kbit/s whereas the Round Trip Time (RTT) has been reduced from 1000 ms to 140 ms. In 2006, an enhancement of UMTS has been released with the name of High Speed Packet Access (HSPA). The data rate has been further increased up to 14.4 Mbit/s (at the time this doctoral thesis has been written). Currently, mobile internet does not only address mobile phones, but also common laptops equipped with an appropriate modem, or datacard. This calls for an internet experience that fulfills the expectations of common home users.

In the following, an overview of the four mentioned wireless cellular systems is provided. The brief descriptions do not claim to be exhaustive explanations of the systems. They are rather supposed to let people not familiar with cellular network issues understand the content of this thesis.

B-1 GSM

Wireless communication systems for nomadic users are usually referred to as *cellular networks*. The radio area coverage is, in fact, structured in *cells*. The architecture of a single Public Land Mobile Network (PLMN) is depicted in Figure B.1.

A cell is the radio area covered by a fixed transmission antenna. One or more transmission antennas are controlled by a network element named Base Transceiver Controller (BTS). The BTS is responsible

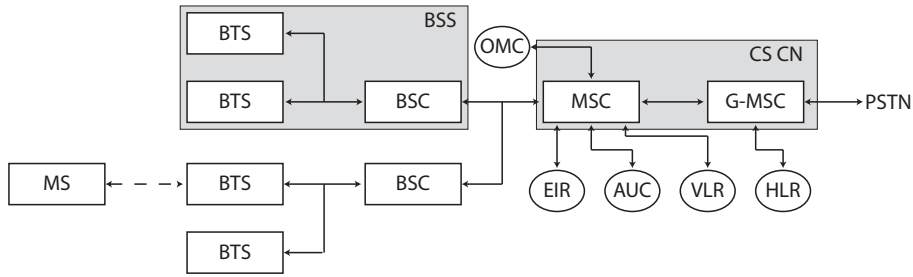


Figure B.1: Structure of a GSM network.

for the cell radio coverage as well as for the communication functionalities (codification, ciphering and modulation) that allow the information exchange with the Mobile Station (**MS**) in the served cell. Several **BTS** are controlled by a single Base Station Controller (**BSC**). For the traffic coming from the **MSs** the **BSC** acts as a concentrator. For the traffic coming from the Mobile-services Switching Center (**MSC**), it acts as a router toward the different **BTSs**. The **BSC** manages the frequency allocation among the controlled **BTS** as well as mobility issues like handover. In case a user crosses the boundaries between two cells, the **BTS** manages the transition of the call between the old and the new cell the user belongs to. The Base Station Subsystem (**BSS**) consists of one **BSC** and of the controlled **BTS**.

The traffic between mobile users located in different **BSSs** is handled by the **MSC**, whereas the connection with elements outside the **GSM** network is managed by the Gateway Mobile-services Switching Center (**GMSC**). These two elements represent the Circuit Switched (**CS**) Core Network (**CN**). The **MSC** is the switching center for the radio services, such as calls control, internetworking toward external networks and user mobility between cells belonging to different **BSSs**. The hierarchical structure of the **GSM** network starts with the definition of administrative regions, consisting of one or more Location Area (**LA**). This, in turn, consists of several cells groups, where each group is controlled by a **BSC**.

Further network elements belonging to the Network SubSystem (**NSS**) are the Home Location Register (**HLR**), Visitor Location Register (**VLR**), Equipment Identity Center (**EIC**), Operation and Maintenance Center (**OMC**) and AUthentication Center (**AUC**). The **HLR** is a database containing the permanent and temporary data of all the users registered with a network operator. The data contains all the subscription details, such as the services allowed for the user and the supplementary services subscribed. The last known position of the user as well as the status of his terminal are stored as temporary information. A single **PLMN** can contain more than one **HLR**. A single International Mobile Subscriber Identity (**IMSI**), the identifier of each Subscriber Identity Module (**SIM**) card, can be associated to a single **HLR** at a time. The International Mobile Equipment Identity (**IMEI**) code of the **MS** can be saved in the **HLR** but it is not used for discriminating users. Each time an user receives a call, the **HLR** is queried to determine the current user’s position and his status. The **HLR** also contains the secret encryption key associated to a user, even though this information can be deciphered only by the **AUC**. The **AUC** is a database containing in an encrypted form the authentication keys, in order to avoid fraudulent actions. Each time the user accesses the network, the **AUC** asks him to prove his identity. An algorithm is stored both on the read only memory of the **SIM** card as well as on the

AUC. As they both generate a code, the authentication center compares the one provided by the user with the one locally generated.

Temporary user information is stored in the **VLR**, a temporary database of the users currently located in the **LAs** served of responsibility of the selected **VLR**. The **VLR** stores the **IMSI** of the users located into the served **LAs** and is also responsible of tracking possible changes, such as new user attached to one cell, user changing **LA** and user detaching from the cell. Each change is communicated back to the **HLR**. Finally, the **OMC** is responsible for controlling operation and configuration of the network. It is used for building up a pattern of subscribers preferences and expectations, as well as applying specific billing strategies depending on the network load or day time.

B-2 GPRS

GPRS, also known as the 2.5 generation of cellular networks, is an enhancement of the **GSM** enabling packet oriented mobile data services. By means of **GPRS**, new services were offered to the users, such as Multimedia Messaging Service (**MMS**) and Wireless Markup Language (**WML**) pages or email services. **WML** is a markup language specifically designed for Wireless Application Protocol (**WAP**) capable mobile phones. In order to enable these new functionalities, new elements have been included in the **GSM** network. As these elements were specifically designed for **GPRS**, they are defined **GPRS** Support Node. The Packet Switched (**PS**) **CN** of the **GPRS** consists of Gateway **GPRS** Support Node (**GGSN**)s and Serving **GPRS** Support Node (**SGSN**)s. In Figure B.2, the enhanced network able to deliver packets within the network and to external Packet Data Network is depicted.

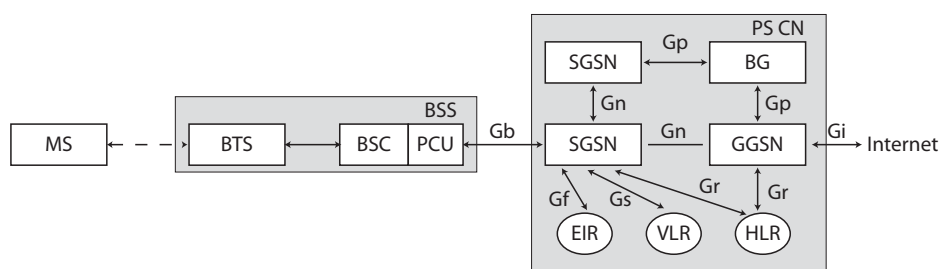


Figure B.2: Structure of a GPRS network.

The **GGSN** is the interface between the **GPRS** network and the external network, like the Internet. The **GGSN** acts like a access router, as the structure of the elements attached to the **GPRS** is hidden to the external network. For packet incoming through the **Gi** interface, the **GGSN** converts the Internet Protocol (**IP**) address of the packet into the **GSM** identification of the receiver.

The **GGSN** establishes a connection with the mobile users by creating a Packet Data Protocol (**PDP**) context. The protocol can be **IP**, **X25** or **FrameRelay**. A schematic description of the **PDP** context establishment is drawn in Figure B.3. In case a mobile station initializes a **GPRS** session, it has to register with the **SGSN** it is attached to. The **MS** asks for a specific **PDP** type, **PDP** address, requested **QoS**. The **SGSN** communicates the request to the **HLR**. If the user is authorized to access the **GPRS** service, a user entry is created in the **VLR** and the **SGSN** sends a request to the **GGSN**. The **GGSN** creates an entry in the **PDP** contexts table, in order to correctly route the packets addressed to

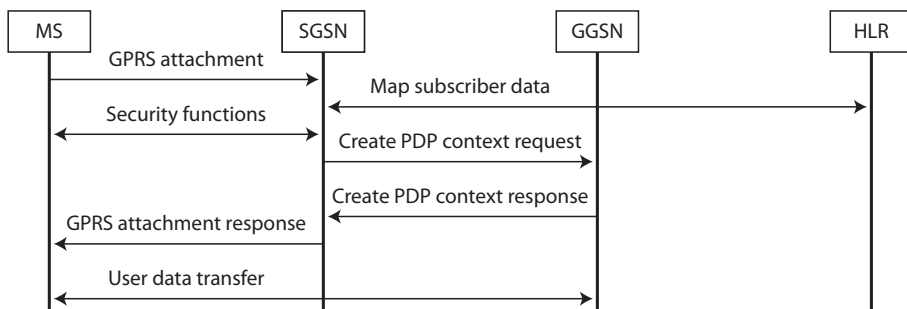


Figure B.3: Establishment of a PDP context.

the user. A **PDP** address (either static or dynamic) together with a positive response are transmitted back to through the **SGSN** to the user.

The external **IP** packets sent to a **MS** are filtered by the **GGSN**, that acts as a firewall. In case the selected user has a valid entry in the **PDP** contexts table, the packets are further encapsulated the GPRS Tunneling Protocol (**GTP**) and transmitted to the appropriate **SGSN** through the Gn interface. The **SGSN** is responsible for further delivering the **GTP** packets to the mobile user. It manages the routing of the packets, mobility issues, authentication and billing of the connection. In order to handle user mobility, the **SGSN** is connected to the **VLR** (Gs interface) and to the **HLR** (Gr interface). The **SGSN** is further attached to the BSC through the Gb interface. An additional network element, the Packet Communication Unit (**PCU**), has been added in order to allocate channels between data and voice. One **SGSN** communicates with a **GGSN** belonging to an external **PLMN** by means of the Gp interface, using the **GTP** protocol.

B-3 UMTS

The **UMTS** [120] has been standardized by the Third Generation Partnership Project (**3GPP**) in the year 2000 (*Release 99*). In the *Release 5*, finalized in 2002, two main features have been added to the standard: (i) the IP Multimedia Subsystem (**IMS**), describing a framework for delivering **IP** multimedia services, and (ii) the cell throughput has been increased by the introduction of the High Speed Download Packet Access (**HSDPA**) from the 384 kbit/s of Release 99 up to 14.4 Mbit/s.

Before the introduction of **UMTS**, the reduced bandwidth forced the **GPRS** customer to utilize the wireless connection for basic Internet services, such as **WAP** browsing or email. As the performance of the wireless transmission technologies are approaching the needs, in terms of bandwidth and delay, of the standard fixed-line customers, also a broader range of Internet services are utilized by nomadic users. The limited amount of available resources calls for a traffic classification and prioritisation.

The **GPRS** offered a soft transition between the second and third generation wireless networks. The core networks of the **UMTS**, both **CS** and **PS**, consists of the same elements of the **GSM** and **GPRS**, respectively. The main differences are located in the UMTS Terrestrial Radio Access Network (**UTRAN**) of the **UMTS**, as shown in Figure B.4

The radio resource management's tasks are covered by the Radio Network Controller (**RNC**), that substitutes the BSC of the **GSM**. The **RNC** is in charge of handling handovers, power control, packet

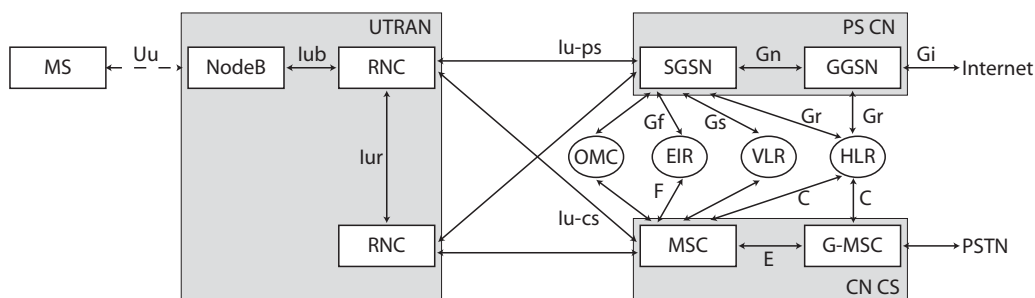


Figure B.4: Structure of a UMTS network.

scheduling and allocating channels as well as the scrambling codes to the users. Each RNC is assigned to a single MSC for CS communications and to a single SGSN for PS communications. An RNC can communicate directly with other RNCs through the Iur interface.

The NodeB is the counterpart of the BTS of the GSM. The NodeB includes a Code Division Multiple Access (CDMA) receiver that converts the signals of the radio interface into a data stream or, in the opposite direction, it prepares the incoming data for being transported over the radio interface. Although the target power control values are set in the RNC, the NodeB performs the inner loop power control, guaranteeing that all the users are receiving the same signal strength.

B-4 HSPA

HSPA is an upgrade of the UMTS technology usually labeled as the 3.5 generation cellular wireless network. It consists of an enhanced downlink, HSDPA, and uplink, High Speed Upload Packet Access (HSUPA), transmission mode. The structure of the HSDPA network is the same as in UMTS, as depicted in Figure B.4. The elements belonging to the PS CN have been not modified, major improvements are present in the UTRAN. The tasks of the RNC and the NodeB have been differently distributed. More functionalities have been moved from the RNC to the NodeB, in particular Adaptive Modulation and Coding (AMC), fast scheduling and fast retransmission. HSDPA also introduces a new transport channel, the High Speed Downlink Shared CHannel (HS-DSCH), which is shared by all the HSDPA users. This replaces the UMTS Dedicated CHannel (DCH), each of which was reserved to a single user.

AMC is the ability of adapting the modulation and coding format to the channel conditions experienced by the users. Assuming that a Transmission Time Interval (TTI) (in HSDPA the length of a TTI is reduced to 2 ms) is reserved to a single user, the NodeB decides the more appropriate transmission parameters depending on the user's reported Channel Quality Indicator (CQI). Whether in UMTS only 4-QAM with a fixed code rate was allowed, in HSDPA 16-QAM and 64-QAM are also enabled.

The scheduling algorithm has been moved from the RNC to the NodeB. As the intelligence has been moved closer to the air interface, the network reacts faster to varying channel conditions and user needs. The re-transmission mechanism has been moved from the RNC to the NodeB as well. This shortens the response time and, moreover, is combined with Hybrid Automatic Retransmission reQuest (HARQ) techniques. If one transport block is not correctly received, or if the MS does not

provide an ACKnowledgment (**ACK**) message within the time limit, the NodeB transmits the same block with incremental redundancy. By combining the multiple versions of the same chunks of data, the **MS** is able to reconstruct the payload.

Appendix C

Abbreviations and Symbols

3GPP	Third Generation Partnership Project
3G	Third Generation
ACK	ACKnowledgment
ACR	Absolute Category Rating
AF	Assured Forwarding
AM	Acknowledgment Mode
AMC	Adaptive Modulation and Coding
ANSI	American National Standards Institute
AUC	AUthentication Center
AVC	Advanced Video Coding
BER	Bit Error Ratio
BSC	Binary Symmetric Channel
BSS	Base Station Subsystem
BTS	Base Transceiver Controller
CLCA	Cross-Layer Content Aware
CA	Context Adaptive
CABAC	Context Adaptive Binary Arithmetic Coding
CAVLC	Context Adaptive Variable Length Coding
CBP	Coded Block Pattern
CBR	Constant Bit Rate
CCTrCH	Coded Composite Transport Channel
cdf	cumulative density function
CDMA	Code Division Multiple Access
CE	Contextual Error
CIF	Common Intermediate Format
CLD	Cross Layer Design
CIS	Class Selector

APPENDIX C. ABBREVIATIONS AND SYMBOLS

CN	Core Network
CRC	Cyclic Redundancy Check
CQI	Channel Quality Indicator
CS	Circuit Switched
DC	Direct Current
DCT	Discrete Cosine Transformation
DCH	Dedicated CHannel
DF	Default Forwarding
DiffServ	Differentiated Services
DL	DownLoad
DP	Data Partitioning
DSCP	DiffServ Code Point
DSL	Digital Subscriber Line
DTX	Discontinuous Transmission
DVB	Digital Video Broadcasting
DVB-H	Digital Video Broadcasting Handheld
DVB-S	Digital Video Broadcasting Satellite
DVB-SH	Digital Video Broadcasting Satellite to Handheld
DVB-T	Digital Video Broadcasting Terrestrial
DVC	Distributed Video Coding
DVD	Digital Versatile Disc
ECN	Explicit Congestion Notification
EF	Expedited Forwarding
EG	Exp-Golomb codewords
EIC	Equipment Identity Center
ETSI	European Telecommunication Standard Institute
FEC	Forward Error Correction
FEW	Force Even Watermarking
FIR	Finite Impulse Response
FL	Fixed Length
FMO	Flexible Macroblock Ordering
FOW	Force Odd Watermarking
FTP	File Transport Protocol
GGSN	Gateway GPRS Support Node
GOB	Groups Of Blocks
GOP	Group Of Picture
GMSC	Gateway Mobile-services Switching Center
GPRS	General packet radio service

GSM	Global System for Mobile communications
GTP	GPRS Tunneling Protocol
HARQ	Hybrid Automatic Retransmission reQuest
HLR	Home Location Register
HSL	Hue Saturation Lightness
HSV	Hue Saturation Value
HS-DSCH	High Speed Downlink Shared CHannel
HSDPA	High Speed Download Packet Access
HSPA	High Speed Packet Access
HSUPA	High Speed Upload Packet Access
HVC	High-efficiency Video Coding
HVGA	Half VGA
HVS	Human Visual System
I	Intra Predicted frame
IC	Illegal Codewords
ICMP	Internet Control Message Protocol
IDR	Instantaneous Decoding Refresh
IEEE	Institute of Electrical and Electronics Engineers
IETF	International Engineering Task Force
IGMP	Internet Group Management Protocol
IMEI	International Mobile Equipment Identity
IMS	IP Multimedia Subsystem
IMSI	International Mobile Subscriber Identity
IP	Internet Protocol
IPTV	Internet Protocol TeleVision
ISO	International Standard Organisation
ISDN	Integrated Services Digital Network
ITU-T	International Telecommunication Unit standardization sector
JM	Joint Model
JVT	Joint Video Team
LA	Location Area
LWSE	Least Weighted Sum Error
MAC	Medium Access Control
MB	MacroBlock
MBLC	MacroBlock Level Concealment
MBMS	Multimedia Broadcast Multicast Service
MIMO	Multiple Input Multiple Output
MMS	Multimedia Messaging Service

APPENDIX C. ABBREVIATIONS AND SYMBOLS

MOS	Mean Opinion Score
MPE	MultiProtocol Encapsulation
MPEG	Moving Picture Expert Group
MPEG-TS	MPEG Transport Stream
MS	Mobile Station
MSC	Mobile-services Switching Center
MSE	Mean Square Error
MMSE	Minimum Mean Square Error
MTU	Maximum Transfer Unit
NACK	Not ACKnowledgment
NAL	Network Abstraction Layer
NALU	Network Abstraction Layer Unit
NRI	NALU Reference Indicator
NSS	Network SubSystem
OE	Optimized Encoding
OMC	Operation and Maintenance Center
OR	Out of Range codewords
OSI	Open Systems Interconnection
P	Inter Predicted frame
P2P	Peer to Peer
PCU	Packet Communication Unit
PDP	Packet Data Protocol
PDU	Protocol Data Unit
PHB	Per Hop Behavior
PIM	Protocol Independent Multicast
PLMN	Public Land Mobile Network
PPS	Picture Parameter Set
PS	Packet Switched
PT	Propagation Time
PSS	Packet Switched Services
QAM	Quadrature Amplitude Modulation
QCIF	Quarter Common Intermediate Format
QP	Quantization Parameter
QoE	Quality of Experience
QoS	Quality of Service
QVGA	Quarter Video Graphic Array
RBW	Relation Based Watermarking
RFC	Request For Comment

RGB	Red Green Blue
RLC	Radio Link Control
RNC	Radio Network Controller
ROI	Region Of Interest
RR	Round Robin
RTP	Real Time Protocol
RTT	Round Trip Time
RTSP	Real Time Stream Protocol
SDU	Service Data Unit
SD	Straight Decoding
SDP	Real Time Stream Protocol
SE	Standard Encoding
SFD	S-Frame Distance
SGSN	Serving GPRS Support Node
SH	Slice Header
SI	Switching I frame
SIM	Subscriber Identity Module
SINR	Signal to Interference and Noise Ratio
SISO	Single Input Single Output
SLC	Slice Level Concealment
SNR	Signal to Noise Ratio
SP	Switching P frame
SPS	Sequence Parameter Set
SVC	Scalable Video Coding
TB	Transport Block
TC	Tabled Codewords
TCP	Transport Control Protocol
TE	Tabled Encoding
TFT	Traffic Flow Template
ToS	Type of Service
TS	Technical Specification
TTI	Transmission Time Interval
TV	TeleVision
UDP	Universal Datagram Protocol
UE	User Equipment
UEP	Unequal Error Protection
UM	Unacknowledgement Mode
UMTS	Universal Mobile Telecommunication System

APPENDIX C. ABBREVIATIONS AND SYMBOLS

UTRAN	UMTS Terrestrial Radio Access Network
VCEG	Video Coding Expert Group
VCL	Video Coding Layer
VGA	Video Graphic Array
VLC	Variable Length Coding
VLR	Visitor Location Register
VoIP	Voice Over IP
WAP	Wireless Application Protocol
WML	Wireless Markup Language
YCbCr	Luminance Chrominance-blue Chrominance-red
Y-PSNR	Luminance Peak Signal to Noise Ratio

Appendix D

List of files used

- Chapter 3
 - `foreman`. (QCIF resolution)
 - `carphone`. (QCIF resolution)
- Chapter 4
 - Section 4.1, Section 4.2. Various sequences grabbed from the Austrian football league `fussball_[1,20]`. Files contained in the folder `football_opt_1`. (QCIF and CIF resolution)
 - Section 4.2 Various sequences grabbed from the Spanish football league `m_[1,10]`. Files contained in the folder `football_opt_2`. (CIF resolution)
- Chapter 5
 - Section 5.1. `foreman` (CIF resolution).
 - Section 5.2. `foreman` (QCIF resolution).

Appendix E

Acknowledgements

Perhaps you will ask me, “Why are there no other drawings in this book as magnificent and impressive as this drawing of the baobabs?”

Antoine de Saint-Exupéry

The acknowledgments belong to the few pages, together with the book cover, which will be read by every person handling, even by chance, this thesis. I will, therefore, commit myself to make your reading experience as magnificent and impressive as possible.

Inizierò col ringraziare le persone che mai potrò ringraziare abbastanza, vale a dire Babbo, Eleonora e Mamma (in ordine alfabetico). Se ora sono qui a scrivere queste pagine lo devo a voi.

I want now to thank the people who made the last four years an amazing experience. *In primis* I thank my supervisor Professor Markus Rupp for giving me this unique opportunity, for always supporting my ideas (no matter how weird they sounded) and for the creative discussions we had. I am also grateful to mobilkom austria ag for financially supporting my work and for the interesting feedbacks.

I want to thank some of the colleagues and friends of the mobile communication group. I will start with Wolfgang: I had the chance to learn a lot from him, both on a Monday to Friday basis as well as during the weekends. Turning his advices into practice has always been a challenging but fruitful experience. I would also like to mention Philipp: he has been the living embodiment of competence, wisdom and kindness in one single human being. It has been also a great pleasure for me to work and share expertise with Martin. My gratitude goes also to Olivia, who guided me in the very first and most critical part of my doctorate.

I am grateful to the following colleagues. Günter, for always being willing to help me with whatever language problem I have had (no more than a couple of them) and for his friendship. Josep, even though he stole half of my vital space, it has been a pleasure to share the room and the music with him. Elena, for sharing with me her Teas and Coffees as well as for the fruitful cheap Spanish crash courses

APPENDIX E. ACKNOWLEDGEMENTS

she gave me. Alex, for the thousands lunches and the few drinks we have had together. Christian, for having helped me in building the MIMO equalizer that I am currently still using. Michal Michal, for all the enthralling chats about life, the universe and everything. The people of the MIMO lab, Qi and Sibbi, for the instructive technical discussions during my visits to the lab.

Outside of the University environment, there are some more people who are significant to me and who I want to thank. My quite older brother Alessandro, Marina and the little Andrea, for having succeeded to let me feel home while at 1147 km (as the crow flies) from home. Leonardo, Carla, Giulia and Francesco, for loving me the way I am and making me believe in myself in some particular moments. Laura, for taking out the most genuine part of myself and for demonstrating me that, no matter how far away you live, you will always remain tight bounded to your roots. Zeppe, for showing me that, even if the time passes by, there is nothing bad on remaining the same people we have been 11 years ago. Gianluca, for being a person who I will always can count on, no matter how big will be the physical distance. Natalie and Monika, for having listened carefully to all my legendary and detailed tales. Danilo, for having addressed me that day on the plane, thus improving significantly my life quality. Domenico, for being such a great friend, always ready to complain for everything that I have done. Nicola, for his kind advices and the sushi we have shared. The little Giammy, for being the greatest fan of my German skills.

Finally, I thank the person that I should thank every day for having blessed me with this life.

List of Figures

1.1	Organization of this thesis.	2
2.1	PSS services and protocols.	10
2.2	PSS protocol stack.	10
2.3	Overhead of the IP/UDP/RTP headers.	11
3.1	Temporal error propagation.	16
3.2	Decoding desynchronization at bitstream level.	17
3.3	H.264/AVC decoder blocks.	19
3.4	Error handling mechanisms.	22
3.5	Cumulative distortion using the three considered approaches.	22
3.6	Y-PSNR comparison using different handling mechanisms and QPs.	23
3.7	Bit error probability mapped to packet error probability.	24
3.8	MSE of corrupted I slices depending on the quality of the previous P frame.	26
3.9	MSE after transmission of P and I frames (QP=28).	26
3.10	Average MSE after transmission (QP=28).	27
3.11	Detected errors: distance between error occurrence and error detection, $k_d - k_e$	28
3.12	Undetected errors: distance between error appearance and end of slice, $k_f - k_e$	28
3.13	Undetected errors: MSE calculated in the range $k_f - k_e$	29
3.14	Visual impairments caused by undetected errors.	29
3.15	Errors detected by visual impairments detection: $k_d - k_e$	32
3.16	Transmission scheme with watermarking.	33
3.17	Watermarking of DCT coefficients.	33
3.18	Comparison between FEW and FOW.	35
3.19	Error detection probability and detection distance of watermarking with three coefficients for each MB.	36
3.20	Quality of watermarking with three coefficients for each MB.	36
3.21	Error probability at bit level.	38
3.22	Implementation of smart NALU sorting.	38
3.23	Sorting mechanism.	39
3.24	Smart sorting results.	40
3.25	Error probability at transport block level.	42
4.1	Transmission chain for soccer videos.	46
4.2	Preprocessing of low definition soccer video sequences.	46
4.3	Dominant color replacement.	47
4.4	Partial occlusions handling.	48
4.5	Clustering based detection mechanism.	49
4.6	Clustering mechanism.	50
4.7	Cluster map.	51
4.8	Penalizing functions for 9×9 pixel cluster.	51

LIST OF FIGURES

4.9	Characteristics of different soccer shots.	53
4.10	Normalized histogram of the HSV components.	55
4.11	Field identification.	55
4.12	Segmentation mechanism.	56
4.13	Results of the segmentation process.	56
4.14	Sample FMO approaches.	57
4.15	Distribution of the macroblocks belonging to the regions R0, R1 and R2.	58
4.16	Probability and cumulative density function of the symbols.	59
4.17	Distribution of the data rate associated to regions R0, R1 and R2.	60
4.18	Block diagram of the modified encoding mechanism.	60
4.19	Dependency of rate quality on the QPs.	61
4.20	Preliminary MOS results.	62
4.21	Camera operations.	63
4.22	Distribution of the horizontal motion vectors.	64
4.23	Amplitude of the horizontal motion vectors.	65
4.24	Horizontal and vertical pan.	65
4.25	Detection of zoom.	66
4.26	Block diagram of the enhanced encoding mechanism.	67
4.27	Rate comparison using different approaches.	68
4.28	Time variant rate distribution.	69
4.29	Soccer MOS results.	70
5.1	Operating modes and operating points.	74
5.2	Rate distortion behavior for different Group Of Picture (GOP) sizes.	75
5.3	Playout buffer of the video decoder.	75
5.4	Implementation of feedback with only-I encoded sequence.	76
5.5	Implementation of SP and SI frames.	77
5.6	Transport block to physical layer mapping.	78
5.7	Measurement setup.	78
5.8	Implementation of the switching.	80
5.9	Implementation schemes in existing networks.	80
5.10	Average desynchronized decoding time.	81
5.11	Rate-Distortion evaluation.	81
5.12	Quality of the Switching P frame (SP) and Switching I frame (SI) frames depending on the QPSPs.	82
5.13	Size of the SP, SI and P frames depending on the QPSPs.	82
5.14	Modeling of the switching process.	83
5.15	Rate-Distortion evaluation.	84
5.16	IPv4 header.	85
5.17	DiffServ domain.	86
5.18	Cross layer optimization protocol stack.	87
5.19	DSCP marking.	88
5.20	UMTS protocol stack (user plane).	89
5.21	Example of two primaries and one secondary PDP context.	89
5.22	Block description of the proposed method.	89
5.23	Overview of the system level simulator.	90
5.24	Interfaces between the video codec and the simulator.	91
5.25	Implementation of the remapping scheme.	92
5.26	Quality as a function of the position of the position of the corrupted frame.	93
5.27	Transport block error probability and quality comparison without retransmission.	93
5.28	Quality comparison with retransmission.	94

A.1	Hierarchical dependency of Sequence Parameter Set (SPS), Picture Parameter Set (PPS) and Network Abstraction Layer Unit (NALU).	102
A.2	Distribution of luminance and chrominance.	103
A.3	Intra prediction mode.	104
A.4	Intra prediction mode macroblock subdivisions.	105
A.5	Sorting of coefficients.	109
B.1	Structure of a GSM network.	112
B.2	Structure of a GPRS network.	113
B.3	Establishment of a PDP context.	114
B.4	Structure of a UMTS network.	115

LIST OF FIGURES

Bibliography

- [1] I. E. Richardson, *H.264 and MPEG-4 Video Compression: Video Coding for Next Generation Multimedia*, 1st ed. Wiley, August 2003.
<http://www.amazon.ca/exec/obidos/redirect?tag=citeulike09-20&path=ASIN/0470848375>
- [2] R. Schaefer, T. Wiegand, and H. Schwarz, “**The emerging H.264/AVC standard**,” EBU Technical review, Tech. Rep., 2003.
- [3] T. Wiegand, G. Sullivan, G. Bjontegaard, and A. Luthra, “**Overview of the H.264/AVC video coding standard**,” *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 13, no. 7, pp. 560–576, July 2003, doi: [10.1109/TCSVT.2003.815165](https://doi.org/10.1109/TCSVT.2003.815165).
- [4] ITU-T, “**ITU-T Recommendation X.641: Information technology - Quality of Service: Framework**,” July 1998.
- [5] ITU-T, “**ITU-T Study Group 12: Definition of Quality of Experience (QoE)**,” March 2004.
- [6] S. Wenger, “**H.264/AVC over IP**,” *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 13, no. 7, pp. 645–656, July 2003, doi: [10.1109/TCSVT.2003.814966](https://doi.org/10.1109/TCSVT.2003.814966).
- [7] L. Superiori, O. Nemethova, and M. Rupp, “**Performance of a H.264/AVC Error Detection Algorithm Based on Syntax Analysis**,” in *Proceedings of International Conference on Advances in Mobile Computing and Multimedia (MoMM)*, pp. 1–10, Yogyakarta, Indonesien, December 2006.
- [8] L. Superiori, O. Nemethova, and M. Rupp, “**Performance of a H.264/AVC Error Detection Algorithm Based on Syntax Check**,” *Journal of Mobile Multimedia*, vol. 3, no. 4, pp. 314–330, 2007.
- [9] L. Superiori, C. Weidmann, and O. Nemethova, “**Error detection mechanisms for encoded video streams**,” in *Video and Multimedia Transmissions over Cellular Networks: Analysis, Modelling and Optimization in Live 3G Mobile Networks*, M. Rupp, Ed., pp. 126–158, 2009.
- [10] L. Superiori, O. Nemethova, and M. Rupp, “**An H.264/AVC Error Detection Algorithm Based on Syntax Analysis**,” in *Multimedia Transcoding in Mobile and Wireless Networks*, pp. 215–234, 2008.
- [11] M. Barni, F. Bartolini, and P. Bianco, “**Performance of syntax-based error detection in H.263 video coding: a quantitative analysis**,” B. Vasudev, T. R. Hsing, A. G. Tescher, and R. L. Stevenson, Eds., vol. 3974, no. 1, pp. 949–956, 2000.
<http://link.aip.org/link/?PSI/3974/949/1>
- [12] L. Superiori, O. Nemethova, and M. Rupp, “**Detection of Visual Impairments in the Visual Domain**,” in *Picture Coding Symposium Proceedings*, Lissabon, Portugal, November 2007.
- [13] E. R. Rodriguez, L. Superiori, O. Nemethova, and M. Rupp, “**Performance of Watermarking as an Error Detection Mechanism for Corrupted H.264/AVC Video Sequences**,” in *Proceedings of the 17th European Signal Processing Conference (EUSIPCO 2009)*, pp. 2206–2210, Glasgow, Scotland, August 2009.
- [14] L. Superiori and M. Rupp, “**Smart Sorting of H.264/AVC Encoded Sequences for Applications over UMTS Networks**,” in *Proceedings of IEEE International Symposium on Consumer Electronics*, Algarve,

BIBLIOGRAPHY

April 2008.

- [15] M. Wrulich, O. Nemethova, L. Superiori, and M. Rupp, “**Ball Appearance Improvement in Low-Resolution Soccer Videos,**” *Elektrotechnik und Informationstechnik (e&i)*, vol. Digitales Fernsehen, no. 10, pp. 337–345, 2007.
- [16] M. Wrulich, L. Superiori, O. Nemethova, and M. Rupp, “**A Robust Preprocessing Algorithm for Low-Resolution Soccer Videos,**” in *Proceedings of the ACM Multimedia 2007*, Germany, Augsburg, September 2007.
- [17] L. Superiori, O. Nemethova, and M. Rupp, “**Clustering-based Object Detection for Low-resolution Video Streaming,**” in *Proceedings of IEEE International Symposium on Broadband Multimedia Systems and Broadcasting*, Orlando, USA, March 2007.
- [18] H. Knoche, M. Papaleo, M. A. Sasse, and A. Vanelli-Coralli, “**The kindest Cut: Enhancing the User Experience of mobile TV through adequate Zooming,**” in *MULTIMEDIA '07: Proceedings of the 15th international conference on Multimedia*, pp. 87–96, New York, NY, USA, 2007, doi: <http://doi.acm.org/10.1145/1291233.1291254>.
- [19] K. Seo, J. Ko, I. Ahn, and C. Kim, “**An Intelligent Display Scheme of Soccer Video on Mobile Devices,**” vol. 17, no. 10, pp. 1395–1401, Oct. 2007, doi: [10.1109/TCSVT.2007.903775](https://doi.org/10.1109/TCSVT.2007.903775).
- [20] L. Superiori and M. Rupp, “**Encoding Optimization of low Resolution Soccer Video Sequences,**” in *Proceedings of IEEE International Conference on Multimedia and Expo*, Hannover, June 2008.
- [21] L. Superiori, A. F. Perez, and M. Rupp, “**Optimization of Audience Encoding in Low-Resolution Soccer Video Sequences,**” in *Proceedings of Asilomar Conference on Signals, Systems and Computers*, Asilomar, California, USA, October 2008.
- [22] L. Superiori and M. Rupp, “**Detection of Pan and Zoom in Soccer Sequences based on H.264/AVC Motion Information,**” in *Proceedings of WIAMIS 2009*, London, May 2009.
- [23] W. Karner, “**Link Error Analysis and Modeling for Cross-Layer Design in UMTS Mobile Communication Networks,**” Ph.D. dissertation, Vienna UT, 2007.
- [24] M. Karczewicz and R. Kurceren, “**The SP- and SI-frames design for H.264/AVC,**” *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 13, no. 7, pp. 637–644, July 2003, doi: [10.1109/TCSVT.2003.814969](https://doi.org/10.1109/TCSVT.2003.814969).
- [25] L. Superiori, W. Karner, and M. Rupp, “**Analysis of Video Streaming with SP and SI Frames in UMTS Mobile Networks,**” in *Proceedings of International Conference on Advances in Mobile Computing and Multimedia (MoMM)*, Kuala Lumpur, Malaysia, December 2009.
- [26] F. Agharebparast and V. Leung, “**QoS support in the UMTS/GPRS backbone network using DiffServ,**” in *Global Telecommunications Conference, 2002. GLOBECOM '02. IEEE*, vol. 2, pp. 1440–1444 vol.2, Nov. 2002, doi: [10.1109/GLOCOM.2002.1188436](https://doi.org/10.1109/GLOCOM.2002.1188436).
- [27] H. Wang, D. Prasad, O. Teyeb, and H.-P. Schwefel, “**Performance Enhancements of UMTS networks using end-to-end QoS provisioning.**” in *International Symposium on Wireless Personal Multimedia Communications*, Aalborg Universitet, 2005.
- [28] L. Superiori, M. Wrulich, P. Svoboda, and M. Rupp, “**Cross-Layer Optimization of Video Services over HSDPA Networks,**” in *Proceedings of Mobilight 2009*, Athens, May 2009.
- [29] L. Superiori, M. Wrulich, P. Svoboda, M. Rupp, J. Fabini, W. Karner, and M. Steinbauer, “**Content-Aware Scheduling for Video Streaming over HSDPA Networks,**” in *Proceedings of IEEE Workshop on Cross Layer Design 2009*, Palma de Mallorca, June 2009.
- [30] ETSI, “**Digital Video Broadcasting (DVB); Framing structure, channel coding and modulation for 11/12 GHz satellite services (ETSI EN 300 42, v1.1.2),**” European Telecommunications Standards

- Institute, Aug. 1997.
- [31] ETSI, “**Digital Video Broadcasting (DVB); Framing structure, channel coding and modulation for digital terrestrial television (ETSI EN 30, 744 v1.6.1)**,” European Telecommunications Standards Institute, Jan. 2009.
- [32] ETSI, “**Digital Video Broadcasting (DVB); DVB-SH Implementation Guidelines (ETSI TS 102 584 V1.1.1)**,” European Telecommunications Standards Institute, Dec. 2008.
- [33] ETSI, “**Digital Video Broadcasting (DVB); Transmission System for Handheld Terminals (DVB-H) (ETSI EN 302 304 V1.1.1)**,” European Telecommunications Standards Institute, Nov. 2004.
- [34] B. Cain, S. Deering, I. Kouvelas, B. Fenner, and A. Thyagarajan, “**Internet Group Management Protocol, Version 3**,” RFC 3376 (Proposed Standard), IETF, Tech. Rep. 3376, Oct. 2002, updated by RFC 4604.
<http://www.ietf.org/rfc/rfc3376.txt>
- [35] A. Adams, J. Nicholas, and W. Siadak, “**Protocol Independent Multicast - Dense Mode (PIM-DM): Protocol Specification (Revised)**,” RFC 3973 (Experimental), Tech. Rep. 3973, Jan. 2005.
<http://www.ietf.org/rfc/rfc3973.txt>
- [36] 3GPP, “**Transparent end-to-end transparent streaming service; Protocols and codecs**,” 3rd Generation Partnership Project (3GPP), TS 26.234, 2008.
http://www.3gpp.org/ftp/Specs/archive/26_series/26.234/
- [37] A.-V. T. W. Group, H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, “**RTP: A Transport Protocol for Real-Time Applications**,” RFC 1889 (Proposed Standard), Internet Engineering Task Force, Jan. 1996, obsoleted by RFC 3550.
<http://www.ietf.org/rfc/rfc1889.txt>
- [38] A.-V. T. W. Group and H. Schulzrinne, “**RTP Profile for Audio and Video Conferences with Minimal Control**,” RFC 1890 (Proposed Standard), Internet Engineering Task Force, Jan. 1996, obsoleted by RFC 3551.
<http://www.ietf.org/rfc/rfc1890.txt>
- [39] S. Wenger, M. Hannuksela, T. Stockhammer, M. Westerlund, and D. Singer, “**RTP Payload Format for H.264 Video**,” RFC 3984 (Proposed Standard), IETF, Tech. Rep. 3984, Feb. 2005.
<http://www.ietf.org/rfc/rfc3984.txt>
- [40] J. Postel, “**User Datagram Protocol**,” RFC 768 (Standard), IETF, Tech. Rep. 768, Aug. 1980.
<http://www.ietf.org/rfc/rfc768.txt>
- [41] J. Postel, “**Transmission Control Protocol**,” RFC 793 (Standard), IETF, Tech. Rep. 793, Sept. 1981, updated by RFCs 1122, 3168.
<http://www.ietf.org/rfc/rfc793.txt>
- [42] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, “**RTP: A Transport Protocol for Real-Time Applications**,” RFC 3550 (Standard), Tech. Rep. 3550, July 2003, updated by RFC 5506.
<http://www.ietf.org/rfc/rfc3550.txt>
- [43] 3GPP, “**Transparent end-to-end packet switched streaming service (PSS); Real-time Transport Protocol (RTP) usage model**,” 3rd Generation Partnership Project (3GPP), TR 26.937, 2008.
http://www.3gpp.org/ftp/Specs/archive/26_series/26.937/
- [44] 3GPP, “**Radio interface protocol architecture**,” 3rd Generation Partnership Project (3GPP), TS 25.301, Sept. 2008.
<http://www.3gpp.org/ftp/Specs/html-info/25301.htm>
- [45] 3GPP, “**Radio Link Control (RLC) protocol specification**,” 3rd Generation Partnership Project (3GPP),

- TS 25.322, Sept. 2008.
<http://www.3gpp.org/ftp/Specs/html-info/25322.htm>
- [46] 3GPP, “**Medium Access Control (MAC) protocol specification**,” 3rd Generation Partnership Project (3GPP), TS 25.321, Sept. 2008.
<http://www.3gpp.org/ftp/Specs/html-info/25321.htm>
- [47] 3GPP, “**End-to-end transparent streaming service; General description**,” 3rd Generation Partnership Project (3GPP), TS 26.233, 2008.
http://www.3gpp.org/ftp/Specs/archive/26_series/26.233/
- [48] 3GPP, “**Packet switched conversational multimedia applications; Default codecs**,” 3rd Generation Partnership Project (3GPP), TS 26.235, 2008.
http://www.3gpp.org/ftp/Specs/archive/26_series/26.235/
- [49] ITU-T, “**ITU-T Recommendation H.263 : Video coding for low bit rate communication**,” International Telecommunications Union, January 2005.
<http://www.itu.int/rec/T-REC-H.263-200501-I/en>
- [50] ISO/IEC, “**ISO/IEC 14496-2:2004 Information technology – Coding of audio-visual objects – Part 2: Visual**,” International Standard Organization IEC, 2004.
- [51] ISO/IEC, “**ISO/IEC 14496-10:2009 Information technology – Coding of audio-visual objects – Part 10: Advanced Video Coding**,” International Standard Organization IEC, 2009.
- [52] ITU-T, “**ITU-T Recommendation H.264 : Advanced video coding for generic audiovisual services**,” International Telecommunications Union, November 2007.
<http://www.itu.int/rec/T-REC-H.264-200711-I/en>
- [53] 3GPP, “**Introduction of the Multimedia Broadcast/Multicast Service (MBMS) in the Radio Access Network (RAN); Stage 2**,” 3rd Generation Partnership Project (3GPP), TS 25.346, Mar. 2008.
<http://www.3gpp.org/ftp/Specs/html-info/25346.htm>
- [54] D. Dostal, “**Handy-tv in oesterreich**,” Master’s thesis, Universtitaet Wien, Fakultaet fr Sozialwissenschaften, 2009.
- [55] “**Oesterreich: Mobiletv fuer drei viertel der handynutzer uninteressant**.”
<http://www.areamobile.de/news/9732-oesterreich-mobiletv-fuer-drei-viertel-der-handynutzer-uninteressant>
- [56] T. Stockhammer, M. Hannuksela, and T. Wiegand, “**H.264/AVC in wireless Environments**,” *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 13, no. 7, pp. 657–673, July 2003, doi: [10.1109/TCSVT.2003.815167](https://doi.org/10.1109/TCSVT.2003.815167).
- [57] “**H.264/AVC JM Reference Software**,” Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG, August 2008.
<http://iphome.hhi.de/suehring/tml/>
- [58] M. Chen, Y. He, and R. Lagendijk, “**A fragile watermark error detection scheme for wireless video communications**,” *Multimedia, IEEE Transactions on*, vol. 7, no. 2, pp. 201–211, April 2005, doi: [10.1109/TMM.2005.843367](https://doi.org/10.1109/TMM.2005.843367).
- [59] O. Nemethova, G. Forte, and M. Rupp, “**Robust Error Detection for H.264/AVC Using Relation Based Fragile Watermarking**,” in *Proceedings of International Conference on Systems, Signals and Image Processing (IWSSIP)*, Budapest, Ungarn, September 2006.
- [60] T. Stockhammer and M. Bystrom, “**H.264/AVC data partitioning for mobile video communication**,” in *Image Processing, 2004. ICIP '04. 2004 International Conference on*, vol. 1, pp. 545–548 Vol. 1, Oct. 2004, doi: [10.1109/ICIP.2004.1418812](https://doi.org/10.1109/ICIP.2004.1418812).
- [61] W. Karner, O. Nemethova, and M. Rupp, “**The Impact of Link Error Modeling on the Quality of**

- Streamed Video in Wireless Networks,**” in *Proceedings of the 3rd International Symposium on Wireless Communications Systems 2006 (ISWCS 2006)*, Valencia, Spanien, September 2006.
- [62] L. Superiori, O. Nemethova, W. Karner, and M. Rupp, “**Cross-Layer Detection of Visual Impairments in H.264/AVC Video Sequences streamed over UMTS Networks,**” in *Proceedings of IEEE 1st International Workshop on Cross Layer Design*, Jinan, Shandong, China, September 2007.
- [63] O. Nemethova, M. Zahumensky, and M. Rupp, “**Preprocessing of Ball Game Video Sequences for Robust Transmission over Mobile Network,**” in *Proceedings of the 9th CDMA International Conference (CIC 2004)*, Seoul, Korea, October 2004.
- [64] T. D’Orazio, N. Ancona, G. Cicirelli, and M. Nitti, “**A Ball Detection Algorithm for Real Soccer Image Sequences,**” *Pattern Recognition, International Conference on*, vol. 1, p. 10210, 2002, doi: <http://doi.ieeecomputersociety.org/10.1109/ICPR.2002.1044654>.
- [65] M. Leo, T. D’Orazio, and A. Distanto, “**Independent Component Analysis for Ball Recognition in Soccer Images,**” in *Proceedings of the IASTED International Conference on Intelligent Systems & Control ISC 2003*, 2003.
- [66] S. Jiang, Q. Ye, W. Gao, and T. Huang, “**A new method to segment playfield and its applications in match analysis in sports video,**” in *MULTIMEDIA ’04: Proceedings of the 12th annual ACM international conference on Multimedia*, pp. 292–295, New York, NY, USA, 2004, doi: <http://doi.acm.org/10.1145/1027527.1027594>.
- [67] X. Yu, Q. Tian, and K. W. Wan, “**A novel ball detection framework for real soccer video,**” in *Multimedia and Expo, 2003. ICME ’03. Proceedings. 2003 International Conference on*, vol. 2, pp. II–265–8 vol.2, July 2003, doi: [10.1109/ICME.2003.1221604](http://doi.ieeecomputersociety.org/10.1109/ICME.2003.1221604).
- [68] X. Yu, C. Xu, H. W. Leong, Q. Tian, Q. Tang, and K. W. Wan, “**Trajectory-based ball detection and tracking with applications to semantic analysis of broadcast soccer video,**” in *MULTIMEDIA ’03: Proceedings of the eleventh ACM international conference on Multimedia*, pp. 11–20, New York, NY, USA, 2003, doi: <http://doi.acm.org/10.1145/957013.957018>.
- [69] Z. Wasik and A. Saffiotti, “**Robust Color Segmentation for the RoboCup Domain,**” vol. 2, p. 20651, Los Alamitos, CA, USA, 2002, doi: <http://doi.ieeecomputersociety.org/10.1109/ICPR.2002.1048386>.
- [70] T. Gevers and A. W. M. Smeulders, “**A comparative study of several color models for color image invariant retrieval,**” in *In Proceedings of the first international workshop on Image databases and multimedia search (IDB-MMS 96)*, pp. 17–26, 1996.
- [71] M. Fairchild, “**Color Appearance Models: CIECAM02 and Beyond,**” Tutorial slides for IS&T/SID 12th Color Imaging Conference, November 2004.
- [72] W. K. Pratt, *Digital Image Processing: PIKS Inside*. New York, NY, USA: John Wiley & Sons, Inc., 2001.
- [73] R. Adams and L. Bischof, “**Seeded Region Growing,**” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 6, pp. 641–647, 1994, doi: <http://doi.ieeecomputersociety.org/10.1109/34.295913>.
- [74] J. R. Ohm, *Multimedia Communication Technology*. Springer, 2004.
- [75] N. Vandenbroucke, L. Macaire, and J.-G. Postaire, “**Color Image Segmentation by Supervised Pixel Classification in a Color Texture Feature Space: Application to Soccer Image Segmentation,**” *Pattern Recognition, International Conference on*, vol. 3, p. 3625, 2000, doi: <http://doi.ieeecomputersociety.org/10.1109/ICPR.2000.903622>.
- [76] Y.-Q. Yang, N. Liu, W. Gu, and X.-Y. Liang, “**An Object Segmentation Approach Based on United Color Models for Soccer Video,**” in *Machine Learning and Cybernetics, 2006 International Conference*

BIBLIOGRAPHY

- on, pp. 3960–3963, Aug. 2006, doi: [10.1109/ICMLC.2006.258790](https://doi.org/10.1109/ICMLC.2006.258790).
- [77] Z. Niu, X. Gao, D. Tao, and X. Li, “**Semantic Video Shot Segmentation Based on Color Ratio Feature and SVM**,” in *Cyberworlds, 2008 International Conference on*, pp. 157–162, Sept. 2008, doi: [10.1109/CW.2008.27](https://doi.org/10.1109/CW.2008.27).
- [78] P. Mazzeo, P. Spagnolo, M. Leo, and T. D’Orazio, “**Visual Players Detection and Tracking in Soccer Matches**,” in *Advanced Video and Signal Based Surveillance, 2008. AVSS ’08. IEEE Fifth International Conference on*, pp. 326–333, Sept. 2008, doi: [10.1109/AVSS.2008.33](https://doi.org/10.1109/AVSS.2008.33).
- [79] Z. Xu and P. Shi, “**Segmentation of players and team discrimination in soccer videos**,” in *VLSI Design and Video Technology, 2005. Proceedings of 2005 IEEE International Workshop on*, pp. 369–372, May 2005, doi: [10.1109/IWVDVT.2005.1504627](https://doi.org/10.1109/IWVDVT.2005.1504627).
- [80] C. Stiller, J. Konrad, and R. Bosch, “**Estimating Motion in Image Sequences - A tutorial on modeling and computation of 2D motion**,” *IEEE Signal Processing Magazine*, vol. 16, pp. 70–91, 1999.
- [81] F. Dufaux and J. Konrad, “**Efficient, robust, and fast global motion estimation for video coding**,” *Image Processing, IEEE Transactions on*, vol. 9, no. 3, pp. 497–501, 2000, doi: [10.1109/83.826785](https://doi.org/10.1109/83.826785).
<http://dx.doi.org/10.1109/83.826785>
- [82] G. Rath and A. Makur, “**Iterative Least Squares and Compression Based Estimations for a Four-Parameter Linear Global Motion Model and Global Motion Compensation**,” vol. 9, no. 7, p. 1075, October 1999.
- [83] A. Dumitras and B. G. Haskell, “**A look-ahead method for pan and zoom detection in video sequences using block-based motion vectors in polar coordinates**,” in *ISCAS (3)*, pp. 853–856, 2004.
- [84] R. Jin, Y. Qi, and E. Hauptmann, “**A Probabilistic Model for Camera Zoom Detection**,” in *Proceedings of ICPR 2002*, pp. 859–862, 2002.
- [85] F. E. Grubbs, “**Procedures for Detecting Outlying Observations in Samples**,” *Technometrics*, vol. 11, no. 1, pp. 1–21, February 1969.
- [86] L.-U. Choi, M. Ivrlac, E. Steinbach, and J. Nosssek, “**Bottom-up approach to cross-layer design for video transmission over wireless channels**,” in *Vehicular Technology Conference, 2005. VTC 2005-Spring. 2005 IEEE 61st*, vol. 5, pp. 3019–3023 Vol. 5, May-1 June 2005, doi: [10.1109/VETECS.2005.1543901](https://doi.org/10.1109/VETECS.2005.1543901).
- [87] M. Ivrlac and J. Nosssek, “**Cross layer optimization - an equivalence class approach**,” in *Smart Antennas, 2004. ITG Workshop on*, pp. 223–230, March 2004, doi: [10.1109/WSA.2004.1407672](https://doi.org/10.1109/WSA.2004.1407672).
- [88] E. Setton and B. Girod, “**Rate-distortion analysis and streaming of SP and SI frames**,” *IEEE Trans. Circuits Syst. Video Techn.*, vol. 16, no. 6, pp. 733–743, 2006.
- [89] C. Berrou, A. Glavieux, and P. Thitimajshima, “**Near Shannon limit error-correcting coding and decoding: Turbo-codes. 1**,” vol. 2, pp. 1064–1070 vol.2, 1993, doi: [10.1109/ICC.1993.397441](https://doi.org/10.1109/ICC.1993.397441).
<http://dx.doi.org/10.1109/ICC.1993.397441>
- [90] E. Setton, P. Ramanathan, and B. Girod, “**Rate-distortion analysis of SP and SI frames**,” in *IEEE International Conference on Image Processing*, Genova, September 2005.
- [91] J. Postel, “**Internet Protocol**,” RFC 791 (Standard), IETF, Tech. Rep. 791, Sept. 1981, updated by RFC 1349.
<http://www.ietf.org/rfc/rfc791.txt>
- [92] K. Kilki, *Differentiated Services for the Internet*. Macmillan Technical Publishing, 1999.
- [93] K. Nichols, S. Blake, F. Baker, and D. Black, “**Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers**,” RFC 2474 (Proposed Standard), Internet Engineering Task Force, Dec. 1998, updated by RFCs 3168, 3260.

- <http://www.ietf.org/rfc/rfc2474.txt>
- [94] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, “**An Architecture for Differentiated Service**,” RFC 2475 (Informational), Internet Engineering Task Force, Dec. 1998, updated by RFC 3260. <http://www.ietf.org/rfc/rfc2475.txt>
- [95] J. Babiarz, K. Chan, and F. Baker, “**Configuration Guidelines for DiffServ Service Classes**,” RFC 4594 (Informational), Internet Engineering Task Force, Aug. 2006. <http://www.ietf.org/rfc/rfc4594.txt>
- [96] B. Braden, D. Clark, J. Crowcroft, B. Davie, S. Deering, D. Estrin, S. Floyd, V. Jacobson, G. Minshall, C. Partridge, L. Peterson, K. Ramakrishnan, S. Shenker, J. Wroclawski, and L. Zhang, “**Recommendations on Queue Management and Congestion Avoidance in the Internet**,” RFC 2309 (Informational), Internet Engineering Task Force, Apr. 1998. <http://www.ietf.org/rfc/rfc2309.txt>
- [97] J. Heinanen, F. Baker, W. Weiss, and J. Wroclawski, “**Assured Forwarding PHB Group**,” RFC 2597 (Proposed Standard), Internet Engineering Task Force, June 1999, updated by RFC 3260. <http://www.ietf.org/rfc/rfc2597.txt>
- [98] B. Davie, A. Charny, J. Bennet, K. Benson, J. L. Boudec, W. Courtney, S. Davari, V. Firoiu, and D. Stiliadis, “**An Expedited Forwarding PHB (Per-Hop Behavior)**,” RFC 3246 (Proposed Standard), Internet Engineering Task Force, Mar. 2002. <http://www.ietf.org/rfc/rfc3246.txt>
- [99] 3GPP, “**Quality of Service (QoS) concept and architecture**,” 3rd Generation Partnership Project (3GPP), TS 23.107, Sept. 2007. <http://www.3gpp.org/ftp/Specs/html-info/23107.htm>
- [100] N. Lopes, M. Nicolau, and A. Santos, “**Efficiency of PRI and WRR DiffServ Scheduling Mechanisms for Real-Time Services on UMTS Environment**,” in *New Technologies, Mobility and Security, 2008. NTMS '08.*, pp. 1–5, Nov. 2008, doi: [10.1109/NTMS.2008.ECP.40](https://doi.org/10.1109/NTMS.2008.ECP.40).
- [101] E. Natalizio, S. Marano, and A. Molinaro, “**Packet scheduling Algorithms for providing QoS on UMTS downlink shared Channels**,” in *Vehicular Technology Conference, 2005. VTC-2005-Fall. 2005 IEEE 62nd*, vol. 4, pp. 2597–2601, Sept., 2005, doi: [10.1109/VETECF.2005.1559019](https://doi.org/10.1109/VETECF.2005.1559019).
- [102] L. Saud and R. Lemos, “**Third generation mobile wireless networks quality of service, with a 2.5G case study using Differentiated Services**,” in *Advances in Wired and Wireless Communication, 2004 IEEE/Sarnoff Symposium on*, pp. 71–74, Apr 2004, doi: [10.1109/SARNOF.2004.1302843](https://doi.org/10.1109/SARNOF.2004.1302843).
- [103] R. Ali, S. Pierre, and Y. Lemieux, “**UMTS-to-IP QoS mapping for voice and video telephony services**,” vol. 19, no. 2, pp. 26–32, March-April 2005, doi: [10.1109/MNET.2005.1407695](https://doi.org/10.1109/MNET.2005.1407695).
- [104] IEEE, “**IEEE Std. 802.11e-2005, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications, Amendment 8: Medium Access Control (MAC) Quality of Service Enhancements**,” IEEE Computer Society, Tech. Rep., 2005.
- [105] A. Ksentini, A. Gueroui, and M. Naimi, “**Improving H.264 video Transmission in 802.11e EDCA**,” in *Computer Communications and Networks, 2005. ICCCN 2005. Proceedings. 14th International Conference on*, pp. 381–386, Oct. 2005, doi: [10.1109/ICCCN.2005.1523891](https://doi.org/10.1109/ICCCN.2005.1523891).
- [106] T. Gan, A. Dejonghe, G. Lenoir, K. Denolf, G. Lafruit, and I. Moccagatta, “**Cross-layer optimization for multi-user video streaming over IEEE 802.11E HCCA wireless networks**,” in *Multimedia and Expo, 2008 IEEE International Conference on*, pp. 505–508, 23 2008-April 26 2008, doi: [10.1109/ICME.2008.4607482](https://doi.org/10.1109/ICME.2008.4607482).
- [107] M. Wrulich, “**System-level modeling and optimization of MIMO HSDPA networks**,” Ph.D. dissertation, Institut für Nachrichtentechnik und Hochfrequenztechnik, Vienna University of Technology, 2009.

BIBLIOGRAPHY

- [108] 3GPP, “**Spacial channel model for Multiple Input Multiple Output (MIMO) simulations,**” 3rd Generation Partnership Project (3GPP), TR 25.996, June 2007.
<http://www.3gpp.org/ftp/Specs/html-info/25996.htm>
- [109] D. Chichon and T. Krner, *Propagation Prediction Models*, L. C. E. Damosso, Ed. Brüssel: European Union Publications, 1999.
- [110] 3GPP, “**Physical layer procedures (FDD),**” 3rd Generation Partnership Project (3GPP), TS 25.214, Sept. 2008.
<http://www.3gpp.org/ftp/Specs/html-info/25214.htm>
- [111] M. Narroschke, “**Benefits and costs of scalable video coding for internet streaming,**” *J. Visual Communication and Image Representation*, vol. 16, no. 4-5, pp. 397–411, 2005.
- [112] ITU-T, “**ITU-T Recommendation H.261: Video Codec for Audiovisual Services at p x 64 kbits,**” International Telecommunications Union, March 1993.
<http://www.itu.int/rec/T-REC-H.261-199303-I/en>
- [113] ISO/IEC, “**ISO/IEC 11172-2:1993 Information technology – Coding of moving pictures and associated audio for digital storage media at up to about 1 Mbit/s – Part 2: Video,**” International Standard Organization IEC, 1993.
- [114] ISO/IEC, “**ISO/IEC 13818-2:2000 Information technology – Generic coding of moving pictures and associated audio information: Video,**” International Standard Organization IEC, 2000.
- [115] ITU-T, “**ITU-T Recommendation H.262: Generic coding of moving pictures and associated audio information: Video,**” International Telecommunications Union, November 2002.
<http://www.itu.int/rec/T-REC-H.262-200211-I/en>
- [116] K. Sayood, *Introduction to Data Compression*, S. Verlag, Ed. Springer-Verlag, 2000.
- [117] M. Rahnema, “**Overview of the GSM system and protocol architecture,**” *Communications Magazine, IEEE*, vol. 31, no. 4, pp. 92–100, 1993, doi: 10.1109/35.210402.
<http://dx.doi.org/10.1109/35.210402>
- [118] C. Bettstetter, H.-J. Voegel, and J. Eberspaecher, “**GSM Phase 2+ - General Packet Radio Service GPRS: Architecture, Protocols and Air Interface,**” *IEEE Communications Surveys*, vol. 2, 1999.
- [119] B. Walke, P. Seidenberger, and M. Peter, *UMTS: the Fundamentals*. Wiley, 2003.
- [120] H. Holma and A. Toskala, *WCDMA for UMTS: Radio Access for Third Generation Mobile Communications*. New York, NY, USA: John Wiley & Sons, Inc., 2000.