Error-Robust Coding and Transmission of Compressed Layered Hybrid Video Streams for Packet-Switched Wireless Networks

Till Halbach

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doktor-Ingeniør

Department of Electronics and Telecommunications
Norwegian University of Science and Technology (NTNU)

2004

Abstract

This dissertation considers packet-switched wireless networks for transmission of variable-rate layered hybrid video streams. Target applications are video streaming and broadcasting services. The work can be divided into two main parts.

In the first part, a novel quality-scalable scheme based on coefficient refinement and encoder quality constraints is developed as a possible extension to the video coding standard H.264. After a technical introduction to the coding tools of H.264 with the main focus on error resilience features, various quality scalability schemes in previous research are reviewed. Based on this discussion, an encoder decoder framework is designed for an arbitrary number of quality layers, hereby also enabling region-of-interest coding. After that, the performance of the new system is exhaustively tested, showing that the bit rate increase typically encountered with scalable hybrid coding schemes is, for certain coding parameters, only small to moderate. The double- and triple-layer constellations of the framework are shown to perform superior to other systems.

The second part considers layered code streams as generated by the scheme of the first part. Various error propagation issues in hybrid streams are discussed, which leads to the definition of a decoder quality constraint and a segmentation of the code stream to transmit. A packetization scheme based on successive source rate consumption is drafted, followed by the formulation of the channel code rate optimization problem for an optimum assignment of available codes to the channel packets. Proper MSE-based error metrics are derived, incorporating the properties of the source signal, a terminate-on-error decoding strategy, error concealment, inter-packet dependencies, and the channel conditions. The Viterbi algorithm is presented as a low-complexity solution to the optimization problem, showing a great adaptivity of the joint source channel coding scheme to the channel conditions. An almost constant image qual-

ity is achieved, also in mismatch situations, while the overall channel code rate decreases only as little as necessary as the channel quality deteriorates. It is further shown that the variance of code distributions is only small, and that the codes are assigned irregularly to all channel packets. A double-layer constellation of the framework clearly outperforms other schemes with a substantial margin.

Keywords — Digital lossy video compression, visual communication, variable bit rate (VBR), SNR scalability, layered image processing, quality layer, hybrid code stream, predictive coding, progressive bit stream, joint source channel coding, fidelity constraint, channel error robustness, resilience, concealment, packet-switched, mobile and wireless ATM, noisy transmission, packet loss, binary symmetric channel, streaming, broadcasting, satellite and radio links, H.264, MPEG-4 AVC, Viterbi, trellis, unequal error protection

Preface

This dissertation is submitted in partial fulfillment of the requirements for the degree of *doktor-ingeniør* (corresponds to Ph.D.) at the Norwegian University of Science and Technology (NTNU).

The research work has taken place between March 1999 and May 2004. It includes compulsory Ph.D. courses of a duration which corresponds to two semesters of full-time studies, and teaching assistantship and other duties which lasted over a time period equivalent to nineteen months.

Involved institutions during the period were the Department of Electronics and Telecommunications (the former Department of Telecommunications), NTNU, where most of the research was carried out, and the Chair and Institute of Communications Engineering, Rheinisch-Westfälische Technische Hochschule (RWTH) Aachen, Germany, where I was guest scientist three times for the duration of two to three months. As a visiting scholar, I spent further six months at the School of Electrical Engineering and Computer Science at Washington State University (WSU) in Pullman (WA, USA).

The teaching assistant work was funded by the Department of Electronics and Telecommunications, and by scholarships from SINTEF under the project number B39999.0 and NTNU with the project numbers 700265 and RSO-811552.

Professor Tor Audun Ramstad at NTNU has been advisor. Temporary local advisors were Professor emeritus Hans Dieter Lüke and Professor Jens-Rainer Ohm at RWTH, as well as Dr. Thomas R. Fischer at WSU.

Except for the matrix manipulation program `Matlab`, this Ph.D. thesis has been developed using free software. The basic system was based on a `Redhat Linux` distribution. Frequently used software packages were `GNU Emacs`, `xfig`, `Gnuplot`, `Octave`, `xpdf`, `gcc`, `ddd`, `TML/JM`, and many other more or less minor utility programs. The typesetting framework is a LaTeX

distribution in combination with the `memoir` package and `pdflatex`. All images are displayed unscaled with a resolution of 72 dpi.

The thesis may contain names of registered trademarks without explicitely being marked as such.

Till Halbach
Trondheim, Norway
May 2004

Acknowledgements

It all began with a diploma thesis abroad. In summer 1997, I was in my final months of study at RWTH, and I had not yet been exchange student, something which has been one of my greatest wishes since I began studying in Aachen in autumn 1992. So the diploma thesis was my last chance. At that time, I worked as a student assistant at the IENT, and *Hans-Dieter Lüke* as the institute's chair happened to know somebody in Norway, more specifically in Trondheim. A contact was quickly established, and hence Hans-Dieter Lüke had a key position in the initiation of my studies at NTNU, for which I am very grateful.

After the diploma thesis was finished, my advisor at NTNU, *Tor A. Ramstad*, asked me if I could imagine to continue as a Ph.D. student. I did, and thus Tor A. Ramstad became my advisor for the next five years. The thesis would not have seen the light of the day in this form without his outstanding advice.

The bonding to Aachen was still quite strong, and in the summers 1999, 2000, and 2001, I had the chance for being visiting scientist at the IENT, for which I am deeply indepted to Hans-Dieter Lüke who enabled the first two visits, and to *Jens-Rainer Ohm* for welcoming me at the RWTH for another stay in 2001. During these periods, I received many helpful advices which were invaluable for later research. I also enjoyed the inspiring working athmosphere at IENT and many fruitful discussions with my colleagues. In particular, I wish to mention *Mathias Wien* who at that time was very active in the standardization development of H.264.

Early in 2002, rumors had it that the research at the WSU in Pullman in my field of studies was of pretty high quality. So later that year, I have had the pleasure of working with *Thomas R. Fischer* at WSU for the duration of six months. I am deeply grateful for all time we spent together, and for his insight and the excellent suggestions he shared with me. Many thanks also for introducing me to life in the United States.

During all the years as a Ph.D. student, I have enjoyed many interesting discussions with the people around me, and I have experienced very friendly support. To avoid the situation that I would have forgotten someone, I just wish to thank generally all my colleages at the Department of Electronics and Telecommunications at NTNU. I would, however, always make an exception to the above rule by mentioning *Kirsten Ekseth*. She has had answers to virtually all of my questions, made the start of my life in Norway much easier, and was in general of invaluable help in many academic and non-academic matters. Special thanks also to *Fredrik Hekland* for proof-reading of the dissertation and *Ole Morten Strand* for reading-improving suggestions.

Another thing I really appreciated was the existance of friendly Internet forums like `comp.text.tex`, the work of all the thousands of software developers who design excellent programs which are freely available under the GNU public license, and the efforts of all engineers and researchers who have contributed within the standardization bodies VCEG and MPEG to H.264's reference software.

Last but not least, the importance of many closely related people must not be underestimated for this work. I am especially thankful for the warm support my parents always have given me during my studies. Moreover, many friends have shared their time for leisure, which has been a refugium and a necessity for quiteness and digression to gather new inspiration and energies. Finally, *Elin Røssvoll* has accompanied my studies with much understanding, and her love has been the major source of my energy towards the finalization of the thesis.

Contents

List of Figures

List of Tables

List of Abbreviations

Mathematical acronyms are listed in the List of Symbols and Notation.

1-D	one-dimensional
2-D	two-dimensional
3-D	three-dimensional
3GPP	3rd-Generation Wireless Protocol
4-D	four-dimensional
AB	anti-blocking
AC	alternating current; or: arithmetic coding
ARQ	automatic repeat request
ASIC	application-specific integrated circuit
ASO	arbitrary slice order
ATM	asynchronous transfer mode
AVC	ATM Video Coding Experts Group; or: Advanced Video Coding
B-ISDN	Broadband ISDN
BLO	BL optimization
BL	base layer
bps	bits per second
BSC	binary symmetric channel
BS	bit stream
CABAC	context-adaptive binary arithmetic coding
CAVLC	context-adaptive variable-length coding
CC	channel coder/coding
CDROM	Read-Only Memory Compact Disk
CD	channel decoder/decoding
chroma	chrominance

CIF	common intermediate format
codec	encoder decoder (pair)
CP	channel packet
CSC	competing scheme
CSI	channel state information
CW	code word
dB	decibel
DC	direct current
DPB	decoded picture buffer
DSP	digital signal processing
DVD	Digital Versatile Disc
EC	entropy coding; or: error concealment
ED	entropy decoder/decoding
EEP	equal error protection
EFC	error-free case
EL	enhancement layer
ESC	EEP scheme
EXPG	exponential Golomb code
FEC	forward error correction
FGS	fine granular scalability
FMO	flexible MB ordering
fps	frames per second
FRC	fixed-rate coding
GOB	group of blocks
GOP	group of pictures
HDTV	high-definition TV
HMM	hidden Markov model
HQLO	HQL optimization
HQL	high-quality layer
HVS	human visual system
IDR	instantaneous decoder refresh
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
III	video coding mode exclusively with succeeding I pictures
IPP	video coding mode with a single I pictures followed by P pictures

IP	Internet Protocol
ISDN	Integrated Services Digital Network
ISO	International Organization for Standardization
IS	International Standard
ITU-T	ITU — Telecommunication Sector
ITU	International Telecommunications Union
I	intra-frame (coding)
JFCD	Joint Final Committee Draft
JM	Joint Model
JPEG	Joint Photographic Experts Group
JSCC	joint source channel coding
JVT	Joint Video Team
kbps	kbits per second
LAN	local area network
luma	luminance
MB	macroblock
MC	motion compensation
MDA	mobile digital assistant
ME	motion estimation
MMS	multimedia message service
MP4FF	MPEG-4 Visual file format
MPEG	Moving Pictures Experts Group
MQLO	MQL optimization
MQL	medium-quality layer
MV	motion vector
Mbps	Mbits per second
NALU	NAL unit
NAL	Network Abstraction Layer
NSC	new scheme
NTSC	National Television Systems Committee
OSI	Open Systems Interconnection
PAL	phase alternating line
PDA	personal digital assistant
pel	pixel
pixel	picture element
PLC	packet loss channel

PMF	probability mass function
PPCRC	punctured parallel concatenated recursive convolutional
PSN	packet-switched network
P	temporally predicted, inter-frame (coding)
QCIF	quarter CIF
QoE	quality of experience
QoS	quality of service
RA	rate allocation
RC	rate control
RGB	red-green-blue
RLC	run length coding
ROI	region-of-interest
RPB	reference picture buffer
RS	Reed-Salomon
RTP	Real-Time Transport Protocol
SC	source coder/coding
SD	source decoder/decoding
SECAM	Sequential couleur avec mêmoire
SNR	signal-to-noise ratio
SPIHT	set partitioning in hierarchical trees
SP	source packet
SSH	SNR-scalable hybrid (codec)
TCP	Transmission Control Protocol
TML	TestTest Model Long-Term
TV	television
UDP	User Datagram Protocol
UEP	unequal error protection
USC	UEP scheme
UVLC	universal variable-length code
v	version
VA	Viterbi algorithm
VBR	variable bit rate
VBV	video buffering verifier
VCEG	Video Coding Experts Group
VCL	Video Coding Layer
VLCD	H.263 Annex-D VLC

VLC	VL coding
VL	variable-length
VoD	video on demand
WATM	wireless ATM
WCDMA	wideband code division multiple access
WLAN	wireless LAN
WN	wireless network

List of Symbols and Notation

All signals are assumed being real-valued.

$\|\| \cdot \|\|$	number of elements in a set
B	channel bandwidth (in *bps*)
BER	bit error rate, also denoted as ϵ
$\mathcal{C}$	channel code set
$C(\cdot)$	number of channel packet redundancy bits (in *bits*)
C	channel capacity (in *source bits per channel bits*)
C_{n}	channel noise
Cb	one of the two chroma components of a 3-D $YCbCr$ signal
$CBER$	channel bit error rate, also denoted as ϵ
CBP	coded block pattern
C_{BSC}	channel capacity of a BSC (in *src. bits per ch. bits*)
Γ_i	channel code option i of a trellis stage
$\mathbf{\Gamma}$	vector of channel codes
$\Gamma(\cdot)$	channel code of a channel packet
$\mathbf{\Gamma}^*$	optimum channel code distribution
Cr	one of the two chroma components of a 3-D $YCbCr$ signal
CRC	cyclic redundancy check
$CS(\cdot)$	channel code specifier
CSF	contrast sensitivity function
$\tilde{D}$	expected distortion
$D(\cdot)$	channel packet distortion
D	distortion
d	common denominator of channel code rates
$d(\cdot,\cdot)$	distance metric

$\bar{D}$	average distortion
$\bar{D}_{\mathrm{c}}(\cdot)$	average channel and concealment distortion
$\bar{D}_{\mathrm{c,f}}(\cdot)$	average GOP noise distortion by layer concealment
$\bar{D}_{\mathrm{c,l}}(\cdot)$	average GOP noise distortion by layer concealment
$\bar{D}_{\mathrm{c,m}}(\cdot)$	average GOP noise distortion by mean concealment
$\bar{D}_{\mathrm{c,t}}(\cdot)$	average GOP noise distortion by temporal concealment
$D_{\mathrm{CP}}(\cdot)$	channel packet distortion
DCT	discrete Fourier transform
$\bar{D}_{\mathrm{e}}(f,l)$	average slice loss GOP distortion
$\bar{D}_{\mathrm{e}}(i,i_{\mathrm{c}})$	average GOP distortion
$\tilde{D}_{\mathrm{GOP}}(\cdot)$	expected GOP distortion
$\tilde{D}_{\mathrm{GOP}}$	expected GOP distortion
$\bar{D}_{\mathrm{jc}}(i)$	average joint GOP concealment distortion
$\bar{D}_{\mathrm{jq,f}}(i)$	average joint GOP quantization distortion by layer concealment
$\bar{D}_{\mathrm{jq,l}}$	average joint GOP quantization GOP distortion by layer concealment
$\tilde{D}^{*}$	minimum expected GOP distortion
$\bar{D}_{\mathrm{q}}(i)$	average GOP quantization distortion
$\bar{D}_{\mathrm{q}}(i)$	average quantization distortion
$\bar{D}_{\mathrm{q,f}}(\cdot)$	average GOP quantization distortion by layer concealment
$\bar{D}_{\mathrm{q,l}}(i)$	average GOP quantization distortion by layer concealment
$\bar{D}_{\mathrm{q,m}}(\cdot)$	average GOP quantization distortion by mean concealment
$\bar{D}_{\mathrm{q,t}}(\cdot)$	average GOP quantization distortion by temporal concealment
ϵ	channel bit error rate
E	prediction error
E_{BL}	base layer prediction
ΔE_{pred}	prediction error difference
E_{HQL}	high-quality layer prediction
$\epsilon_{\mathrm{layered}}$	efficiency of layered relative to non-layered coding
$\epsilon_{\mathrm{layered}}^{(\mathrm{BLO})}$	efficiency of layered coding with base layer optimization

$\epsilon_{\text{layered}}^{(\text{HQLO})}$	efficiency of layered coding with high-quality layer optimization
$\Delta\epsilon$	mismatch channel error rate difference
ϵ_{true}	true bit error rate
$E\{\cdot\}$	expectation of a random variable
$f(\cdot)$	frame index of a video segment
F	frame buffer
F_{AB}	anti-blocking filter function
$F_{\text{AB}}^{(\text{HQL})}$	high-quality layer anti-blocking filter
F_{org}	frame rate (in *fps*)
F_{src}	encoded frame rate (in *fps*)
$g(\cdot)$	frame index counter
$H(\cdot)$	entropy of a discrete variable (in *bits per sample*)
i	counting index
$i_{\text{s}}(\cdot,\cdot,\cdot,\cdot)$	source packet index
i_{c}	channel packet index
$\mathcal{I}_{\text{e}}(\cdot)$	set of source packet indices
$\mathcal{I}_{\text{e,BL}}(\cdot)$	source packets indices set of base layer slices
$\mathcal{I}_{\text{e,BL,1}}(\cdot)$	set of source packet indices
$\mathcal{I}_{\text{e,BL,2}}(\cdot)$	set of source packet indices
IQ	inverse quantization
$i_{\text{s,BL}}$	index of the first slice in the next frame's base layer
$i_{\text{s,f}}$	index of the first source packet in a channel packet
$i_{\text{s,l}}$	index of the last source packet in a channel packet
$i_{\text{s,s}}(\cdot)$	base layer slice index
IT	inverse transform
$\mathcal{J}(\cdot)$	set of source packet indices
J	Lagrangian functional
j	counting index
k	index counter
k_i	number of source bits per channel redundancy bits
λ	Lagrangian multiplier
$l(\cdot)$	layer index of a video segment
L_i	signal length
$\bar{m}_X$	signal mean

$\mapsto$	mapping operator
$\mathcal{M}$	mapping
M	number of samples vertically
m	channel mismatch factor
$m(\cdot)$	layer index counter
MOS	mean opinion score
MSE	mean squared error
$\widetilde{MSE}_{\text{GOP}}$	expected GOP MSE
$m_{\tilde{X}}$	mean of the signal $\tilde{X}$
$\mathbb{N}$	space of natural numbers
N	number of samples horizontally
N_{CP}	number of channel packets
N_{f}	number of frames per GOP
N_{l}	number of quality layers of a video
$N_{\text{MB}}(f)$	total number of MBs in a frame
$N_{\text{MB}}(f, y, x)$	number of MBs of a particular frame slice
N_{par}	number of rate distortion points
N_{s}	number of slices per video layer
N_{SP}	number of source packets
N_{t}	number of transmitted channel packets
N_{x}	number of slices in one row of a single video image
N_{y}	number of slices in one column of a single video image
$P(\cdot)$	consecutive error probability
$P_{\text{e}}(\cdot)$	channel packet loss probability
$P_{\text{e}}(\cdot, \cdot, \cdot)$	channel packet loss probability
P_{I}	spatial prediction filter function
PLR	packet loss ratio
PMF	probability mass function
$P_{\text{ne}}(\cdot, \cdot)$	no-error probability
P_{ne}	(with various superscripts) layer no-error probability
$P_{\text{ne}}^{(\text{a})}(\cdot)$	accumulated no-error probability
P_{P}	temporal prediction filter function
$\widetilde{PSNR}_{\text{GOP}}$	expected GOP $PSNR$ (in dB)
$PSNR$	peak-signal-to-noise ratio
$PSNR_{\text{BL}}$	base layer $PSNR$

$PSNR_{\mathrm{HQL}}$	high-quality layer $PSNR$
Q	quantization
QP_{BL}	base layer QP
QP_{HQL}	high-quality QP
ΔQP	difference of two QPs
QP	quantization parameter
QP_{MQL}	medium-quality layer QP
ΔR	rate difference
$\lfloor \cdot \rfloor$	rounding to the nearest integer towards minus infinity
$\mathcal{R}(\cdot)$	set of channel packet indices
R	(with various subscripts) signal rate
r	channel packet index
$RBER$	residual BER
$R_{\mathrm{c}}(\cdot)$	payload rate (in *bits*) of a channel packet
$R_{\mathrm{c}}^{(\mathrm{a})}(\cdot)$	accumulated channel payload rate (in *bits*)
r_{CC}	channel code rate (in *source bits per channel bits*)
$r_{\mathrm{cc},i}$	channel code rate (in *source bits per channel bits*)
R_{ch}	channel rate (in *source bits per second*)
R_{CP}	length (in *bits*) of a channel packet
RDP	rate distortion product
RDP_{layered}	layered RDP
$RDP_{\mathrm{non\text{-}layered}}$	non-layered RDP
R_{org}	information rate (in *information bits per second*)
$R_{\mathrm{s}}(f)$	layer source rate per video frame (in *bits*)
$R_{\mathrm{s}}(i)$	source rate of a channel packet (in *bits*)
$R_{\mathrm{s}}^{(\mathrm{a})}(\cdot)$	accumulated source rate (in *bits*)
r_{SC}	compression ratio (in *information bits per source bits*)
R_{src}	source rate (in *source bits per second*)
R_{tr}	transmission rate (in *channel bits per second*)
$\mathcal{S}(\cdot)$	set of source packet indices
s	source packet index
$SATD$	sum of absolute transformed differences
s_{d}	decoding state
SG	slice group (identifier)
S_{i}	transform coefficient set

SNR	signal-to-noise ratio
SSD	sum of squared differences
σ_X^2	signal variance
$\sigma_{\tilde{X}}^2$	variance of the signal $\tilde{X}$
T	transform
t	source packet index
T_4	primary 4×4 transform matrix in H.264
$T_H\{\cdot\}$	2-D Hadarmard transform
U	one of the two chroma components of a 3-D YUV signal
V	one of the two chroma components of a 3-D YUV signal
$w_{\text{ss}}(\cdot)$	slice size weighting factor
ΔX_{rec}	reconstruction error
$\tilde{X}_{\text{BL}}$	reconstructed (encoded and decoded) 2-D base layer signal
$x(\cdot)$	horizontal index of a video segment
X	arbitrary 2-D random variable
ξ_i	sub-vector
$X_{\max}$	maximum signal value
$X(\cdot,\cdot,\cdot)$	original video slice
$\hat{X}$	predicted signal
$\tilde{X}(\cdot,\cdot,\cdot,\cdot)$	reconstructed video segment/slice
W	arbitrary 1-D signal
$y(\cdot)$	vertical index of a video segment
Y	2-D coefficient matrix, or 2-D arbitrary random signal
Y	luminance component of a 3-D $YCbCr$ or YUV signal
Z	arbitrary 1-D signal

Chapter 1

Introduction

Mobile communications and the demand for multimedia content have experienced unequaled rapid growth in the last decade. Naturally, the great — albeit prevailingly separate — commercial successes in these areas fuel the old vision of ubiquitous mobile multimedia communication: being able to communicate at any time from anywhere any type of data. At the time of writing, the convergence of both areas is underway.

There are several prerequisites for this development. Most important, low-power general-purpose processors and application-specific integrated circuits (ASICs) are becoming faster and more integrated than before, while their limited power consumption is maintained at the same time. Additionally, the integration of memory chips has been intensified. The prices for both components, processor and memory, have dropped continuously, such that the industry nowadays, with the advent of new powerful signal processing/compression algorithms, is able and willing to integrate new components into their products in order to be able to offer cheap and advanced communication devices on the consumer markets. Also the infrastructure of networks has been significantly improved in recent years. This means higher reliability, increased bandwidth, faster transmission, and a higher degree of link-ups.

Considering the content/source and in particular visual information, image quality has been enhanced. An example for this are bigger TV screens and images of very high resolution, both of which enable a higher quality of experience (QoE). Finally, new features like region-of-interest (ROI) coding and spatial scalability in image coding have opened new markets. When it comes to the storage of multimedia information in form of digital data, optical storage devices like CDROM and its succes-

sor DVD have conquered the consumer markets for e.g. movie distribution and home cinema. Yet, access to multimedia databases becomes a more and more common phenomenon, where, upon request, a multimedia stream is delivered to the recipient over different channel types.

The tendency of the explosive increase of use of cellular phones appears unbroken, partly motivated by the all-in-one approach of personal 'communicators' which offer additional features like radio, audio player, speech recording, etc., and partly based on advances in battery technology. The triumph of mobile telephones suggests hereby that, in the future, many of these devices are capable of high-capacity data transfers; they may for example be able to receive content from a broadcast service. The cellular phones offering services like multimedia message service (MMS) are most likely only the first vanguard of the next generation of mobile devices. Taking all these aspects into account, wireless multimedia services are likely to find widespread acceptance within the next decade. As visual information, i.e. still images and videos, dominates in multimedia content, advanced image processing is playing a major role in multimedia applications and will continue to do so.

Even though there is the trend today that more and more bandwidth becomes available for communication, several factors counteract this development. As already mentioned, the consumer expectation towards visual information in form of e.g. larger image sizes and better image quality has grown simultaneously, and so has the number of content providers. An example for this is the very high number of TV channels among which the consumer can choose nowadays. Compression of visual content is therefore still highly desirable and often necessary.

This work deals with robust digital communication, i.e. compression, of visual information for noisy channels. First, considered channel types are derived from the list of target applications. The focus then turns to a generic compression system which connects the sink with the source. The definition of terms in robust compression is followed by a brief introduction to the area of joint source channel coding. The chapter closes by naming the work's contributions to science and the industry sector, and with a brief outline.

1.1 Target applications

There is a — probably uncountable — multitude of possible applications which employ visual compression, and each application has requirements

which differ from each other, regarding complexity, latency, feedback, etc. A set of applications is therefore specified in the following to limit the scope of this work.

Four main aspects dominate the requirements and assumptions made in the subsequent chapters.

1. As the main interest is on mobile devices, the system must not be too complex to implement, and its power consumption should be low. Also, the communication link to the end user will be wireless.

2. The frequency spectrum, i.e. bandwidth, for radio signals is usually very limited, which requires the frequency range allowed for a particular service be exploited as much as possible to maximize the amount of data exchange.

3. Many mobile end terminals allow for inter-activity and inter-operatorability, which basically means that both encoder and decoder of a compression system are integrated into the same device. Usually, there is either a symmetric two-way channel or an asymmetric feedback; however, this does not apply to services like broadcast and streaming where a multitude of feed-back channels is not feasible. As a consequence, the well known automatic repeat request (ARQ) schemes cannot be deployed, and it is difficult to guaranty error-free transmission while simultaneously maintaining a high image quality. Thus, either residual errors may have to be expected in the code stream, or robust techniques with the property of graceful performance degradation in the presence of errors must be utilized.

4. Communication denotes often the use of conversational services, which leads to the requirement that the round-trip delay, i.e. the time — in a packet-wise transmission system — between the transmission of the first channel symbol of a data packet and receiving the last channel symbol of the transmission report packet, be limited to 250 ms. Systems which fulfill these requirements are called real-time or delay-sensitive applications, the examples being video phone and conferencing.

   Applications with moderate latency requirements are entertainment video applications. Here, the delay bounds are between 0.5 and 2 s [WSBL03]. Yet, streaming services may allow delays of 2 s and more. The focus of this work is on non-conversational applications.

In recent years, there has been a tremendous interest in the asynchronous transfer of moving pictures, also referred to as packet video. In fact, any type of data can be grouped into strings of bits and conveyed over generic packet-switched networks (PSNs). A popular representative of this kind of channels are asynchronous transfer mode (ATM) networks. The principles of ATM transmission are, among many other applications, utilized in the widely spread Broadband ISDN (B-ISDN). Lately, with the advent of mobile devices, such as laptops, PDAs (or more specifically MDAs), and cell phones, also wireless ATM (WATM) transmission schemes have come into the focus, employed for example in wireless LANs (WLANs). This includes WCDMA schemes for wireless networks (WNs) like 3GPP, and WLAN systems based on IEEE 802.11, also known as WiFi.

Video communication and ATM are exceptionally well suited for each other. A video encoder which is required to guaranty a certain quality of service (QoS), i.e. image quality, will typically produce a variable bit rate (VBR) output stream. This is also referred to as unconstrained VBR [LOR98]. In case of much motion in a scene, the rate consumption of an encoder with quality constraint increases. An ATM network is capable of coping with many such VBR sources by statistical multiplexing [RB99]. Typically, the network has a quite heterogeneous structure; there exist a variety of switching nodes, wired and wireless links, central servers, and end user terminals. A typical constellation is shown in Fig. 1.1, where data is distributed from a media server connected to a packet-switched network. The packets are eventually routed via a base station to a mobile terminal.

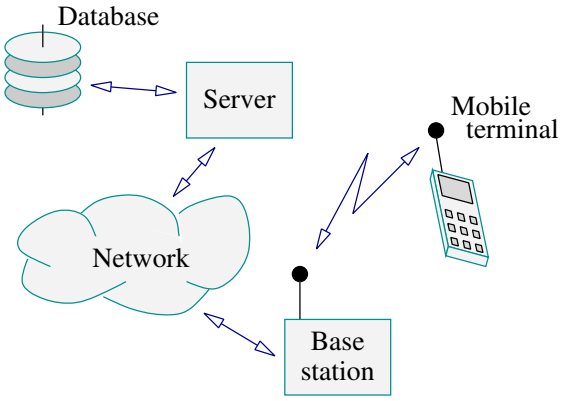


Figure 1.1 — Database access over a wireless hop and a packet network link

The focus of this work is on the final distance to and from the mobile terminal, which implies lossy transmission. Considering time-discrete amplitude-discrete channels and in particular binary channels due to their simplicity, random bit and burst errors are likely to occur under the transmission. Burst errors can, by means of sufficiently large pseudo-random interleavers, be turned into random bit errors, which is thus the kind of errors considered in the sequel.

There is a multitude of possible applications for the solutions presented in this work. The primary target are entertainment video applications like digital TV broadcasting and direct-to-home satellite distribution. Other important application fields are media streaming services including video on demand (VoD), 3GPP MMS, and video mail, video surveillance, as well as multicast applications.

1.2 Compression

A generic system that communicates a source over a channel to a sink consists of an encoder and a decoder, jointly called codec. Such a system is depicted in Fig. 1.2.

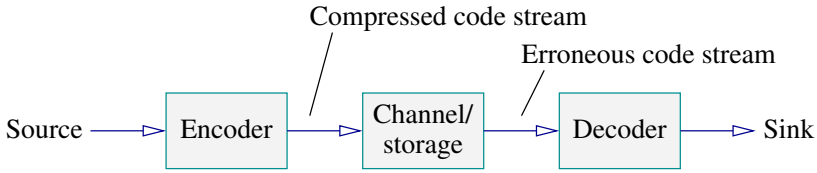


Figure 1.2 — A generic communication system

Compression is necessary due to the amount of data to convey and bandwidth constraints of the channel, and it is possible because of the high redundancy inherent to raw visual data. By removing statistical redundancy, the statistics of the source are accounted for. By the removal of psycho-visual redundancy, so-called irrelevancy, the limitations of the human visual system (HVS) are exploited; see also App. B. A scheme based on these two principles is called a lossy compression system. The theoretical bounds of lossy coding are described by distortion rate theory [Sha59]. Given a signal which has to be transmitted with a fixed transmission rate R, at least the distortion $D(R)$ is introduced by the compression process. Or vice versa, given a signal which by coding is compressed with the distortion D, at least the transmission rate $R(D)$ is

necessary to transmit the compressed information over the channel. Ideally, the signal reconstructed at the decoder is as close as possible to the original. In case of lossy transmission, D ideally includes the contribution of channel errors.

1.3 Robustness and joint source channel coding

As explained in the previous sections, transmission/channel errors are inevitable on many channels. A (channel) *error* is defined as one or several channel symbols altered during transmission. Error *robustness* or *resistance* denotes the property of both encoder and decoder to be able to handle channel errors such that the visual impact on the decoded video be as small as possible. The less error-robust a system, the more vulnerable it is to errors.

On the encoder side, the ability of generating a robust code stream is called error *resilience*. Resilience tries to prevent errors from spreading spatially or temporally. Robustness on the decoder side is achieved by error *concealment*, i.e. the hiding of errors in a visually pleasing manner. Concealment techniques, also denoted as signal *recovery/reconstruction/ restoration*, assume a successful error detection.

Residual errors are symbol errors which remain in the code stream passed to the source decoder after attempted error correction by channel codes. Errors may not be corrected either because they have not been detected as such before, or channel conditions are beyond the correction capabilities of the channel codes employed. In any case, error resilience measures support channel coding, i.e. error robust code streams allow the use of channel codes which — for the same channel conditions — may be weaker than those employed together with code streams which are very vulnerable to errors.

There are many opinions about what kind of error types to pay attention to at the decoder. As an example, the recently published video coding standard H.264 [ITU03] ignores random bit errors, arguing that these can always be converted into packet erasures at a physical- or link layer level [Wen02]. However, other research [KG98] asserts that residual errors must always be accounted for, and that they cannot be totally avoided, especially for real-time or near-real-time transmission in mobile environments [GF98]. This work acknowledges both positions.

Traditionally, most communication systems treat source and channel coding separately. This is motivated by the source channel separation

principle which says that the process of coding of a signal can be split into two generic blocks without loss in optimality [Sha48]. However, the assertion is based on the assumption of a stationary and memoryless channel with infinite delay and complexity. Given that most practical channels are non-stationary, and that infinite system delay and complexity are not feasible, it is expected to gain both increased coding efficiency and higher error robustness with a joint treatment of source and channel coding, called joint source channel coding (JSCC).

Many papers have already been published on this topic, and [ADR96] and [KIK02] may serve as starting points for literature research. The scheme developed later in this dissertation can also be classified as a JSCC technique.

1.4 Outline of the dissertation

The upcoming chapters and sections are organized as follows.

I) **Hybrid scalable coding**
This is the former of the two major parts of the thesis. It includes

1) a motivation for layered coding,
2) an introduction to hybrid coding,
3) a review of various scalability schemes,
4) the development of a new SNR-scalable technique,
5) an exhaustive description of all performance evaluation experiments carried out, and
6) a brief summary at the end as well as conclusions and an outlook regarding this first part.

II) **Robust transmission of code streams**
This is the latter of the two major parts of the thesis. It contains

1) a review of previous research concerning embedded and hybrid code streams,
2) an analysis of error propagation in hybrid code streams,
3) the design of a packetization scheme for channel packets,
4) the theoretical basis for a new channel rate allocation method, including the derivations of a proper error metric,

5) the description of a practical low-complexity implementation featuring the Viterbi algorithm,

6) a section with detailed experimental results reflecting the new system's performance, and

7) a verification of the theoretical results by a software simulation.

III) **Summary**
Here, content and results of the second part of the thesis are summarized, and conclusions are drawn.

IV) **Appendices**
The appendices give

1) a description of the original source signals employed,

2) a brief general discussion of the influence of properties of the human visual system on communication systems, and

3) a listing of various error metrics utilized throughout this work.

1.5 Contributions of the dissertation

Corresponding to the outline of the dissertation, its major contributions are two-fold.

Regarding scalable coding, the dissertation provides

1. a technical discussion of the video coding standard H.264 and its error resilience features,

2. a review of quality scalability schemes in international standards, and a survey of approaches during the standardization of H.26L,

3. the development of a new technique for SNR scalability and ROI coding as a possible extension of H.264, including a software implementation, a detailed performance evaluation, and comparisons to other approaches, and

4. recommendations for parameter settings for a high coding efficiency of the new SNR-scalable scheme.

With regard to robust transmission of compressed layered hybrid code streams for packet-switched wireless networks, the thesis contributes

1. an overview of JSCC schemes for transport of embedded and hybrid code streams,
2. an analysis of error propagation in H.264,
3. an extension of the current standard to a JSCC framework, the focus of which being channel rate allocation, considering both encoder and decoder side, and the channel,
4. a proposal for segmentation of a layered hybrid code stream for packet-wise transmission,
5. the description of a suitable channel packet architecture and packetization scheme based on successive source consumption,
6. the derivation of proper error metrics describing the system performance,
7. a new definition of the expected group of pictures distortion tailored for scalable hybrid code streams,
8. the proposal for a source rate constraint to achieve constant quality output, and
9. a software implementation featuring the Viterbi algorithm as a low-complexity solution to the optimum channel code rate allocation problem, including an in-depth performance evaluation of the developed scheme and comparisons to other research.

Chapter 2

Hybrid scalable coding

This chapter gives the rationale for scalable coding within the video compression scheme commonly denoted as a hybrid video codec. Without loss in generalization, block-based hybrid coding is exemplified in the following by the coding tools specified in the H.264 standard.

2.1 Scalability in hybrid coding

This section discusses the necessity of scalability for the aforementioned applications.

Scalability in video communication is desirable for many reasons. *SNR scalability* denotes the ability of a communication system to display imagery with several quality levels. Applications which allow for *spatial scalability* can offer different image sizes or, in other words, different spatial resolutions. The issue of various temporal resolutions in terms of frame rate is addressed by *temporal scalability*. Other scalability features exist like complexity scalability, memory scalability, and latency scalability. The main motivation for scalability is the availability of different resources like computational power at encoder and/or decoder, or channel bandwidth or image display size.

Scalability is usually associated with hierarchically constructed service layers, each of which offers a different degree of service in terms of image quality, coding complexity, etc. The idea of layered coding, and more specifically quality scalability, was presented in the literature for the first time in the context of ATM networks [DV86, Ver86]. Without loss in generality, SNR scalability is considered subsequently. Possible appli-

cations which utilize quality scalability are multipoint video conferencing and video communications on ATM networks.

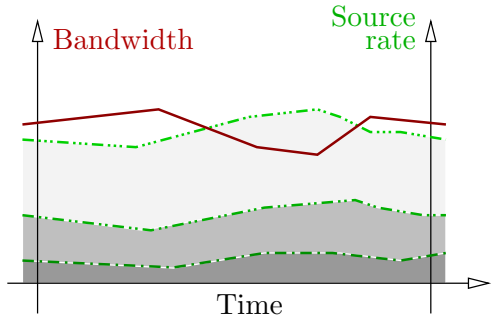


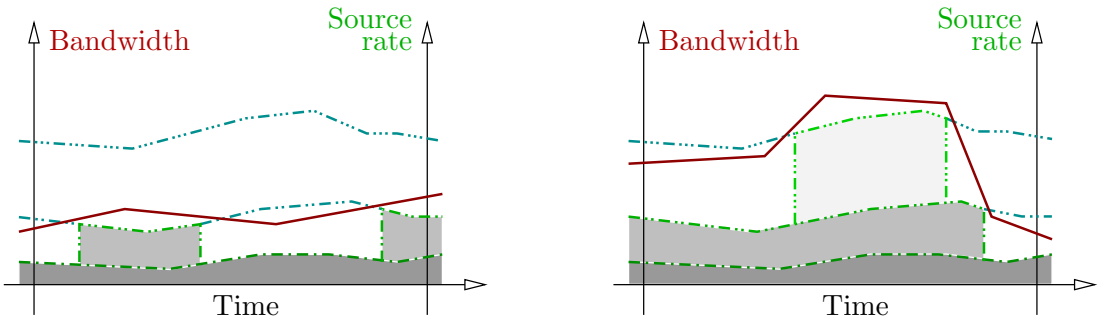
Figure 2.1 — Example progression of channel bandwidth and rates of three video layers over time. Here, the unicast application transmits the streams regardless the bandwidth

In today's heterogeneous networks, channel in-stationarity is often encountered, which leads to bandwidth fluctuations as sketched in Fig. 2.1. A pre-produced code stream or a fixed-rate coding (FRC) scheme, however, cannot react to varying communication rate requirements. The inability to adjust the rate causes under-utilization of channel resources in case the communication rate is below the available bandwidth, and network traffic congestion when the rate is higher [Rhe98]. Multiple pre-produced non-scalable streams as in *simulcast* applications or simultaneous encoding at different rates, as well as video transcoding can offer variable-rate communication; however, this comes at the cost of an increased usage of storage resources, computational power, or latency [tT00].

A major drawback of block-based hybrid systems which offer SNR scalability is the significant increase in bit rate as compared to a single-layer representation which represents the data at the highest quality of a multiple-layer framework. Hence, generally speaking, any framework which offers multiple-layer representations trades off rate and quantization distortion.

There exists a variety of SNR-scalable schemes which are based on both block-based hybrid coding of the low-quality layer, as well as an additional transform combined with subsequent bit plane coding, or a subband decomposition for higher-quality layers, so-called *fine granular scalability* schemes (e.g. [RC99, ISO00]). However, as discussed in Sec. 2.3, these schemes suffer from a drift problem and, in case of subband coding,

have the disadvantage of a considerably higher complexity than hybrid coding alone. Therefore, this work considers exclusively systems which produce hybrid code streams for all fidelity layers.



(a) Receiver 1 (b) Receiver 2

Figure 2.2 — A multicast application which allows for transmission of all streams for which the total rate is below the channel bandwidth

In contrast to stream replication, a scalable coding scheme can be utilized by *multicast* (one sender, several receivers) and *unicast* (one sender, one receiver) applications, such that the available bandwidth be shared. SNR scalability offers several quality layers at different communication rates. As the spatial frequencies vary among the layers, it is also referred to as *frequency scalability*.

The layer of lowest quality is called *base layer*. Higher-quality representations of the video are accomplished by combining the data of a low-quality layer with data of one or several *enhancement/refinement* layers. Rate control can efficiently be achieved by dynamically changing the number of layers for each receiver, as illustrated in Fig. 2.2.

2.2 H.264 and MPEG-4 AVC

This section introduces H.264 and gives an overview of its Baseline[1] coding tools. Parts thereof were presented in [HW02] and [Hal03a].

ITU-T Recommendation H.264 [ITU03] is identical with ISO/IEC International Standard 14496-10 [ISO03], informally known as MPEG-4 Part 10/AVC, due to joint standardization efforts of the so-called Joint

[1] In this work, only progressive-scan, i.e. non-interlaced image material is considered. One *picture* is hence identical with one video frame.

Video Team (JVT), formed by ITU-T and ISO/IEC [Hal02e]. The standard is the first third-generation video coding scheme after the first generation with H.120, H.261 and MPEG-1 Video, and the second generation which consists of the standards H.263, MPEG-2 Video, and MPEG-4 Visual [Hal03b].

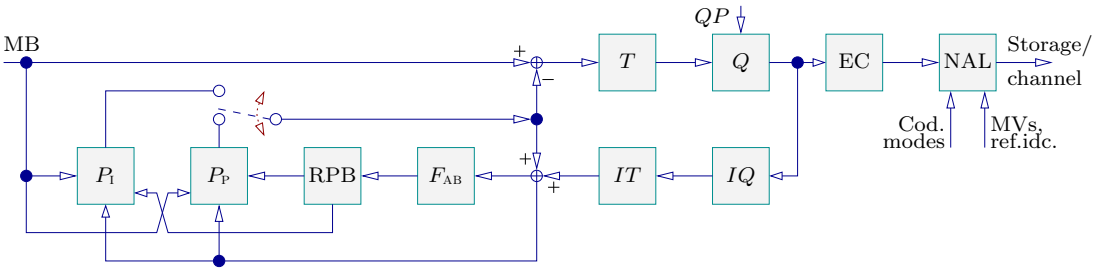


Figure 2.3 — Flow diagram of an H.264 Baseline-compliant encoder. The reference picture buffer (RPB) is identical with the decoded picture buffer (DPB) from the specifications. P_I denotes INTRA and P_P INTER prediction. Anti-blocking filter, transform, inverse transform, quantization, and inverse quantization are represented by F_{AB}, T, IT, Q, IQ, respectively. The EC block performs entropy coding and passes the resulting code stream to the Network Abstraction Layer (NAL). There, the encoded data are multiplexed with side information like coding mode, motion vectors, and reference indices, and finally transmitted. All blocks except for the NAL amount to the Video Coding Layer (VCL)

For a better illustration of the block-based hybrid coding scheme of H.264, the block diagram of an H.264 Baseline-compliant encoder is derived from the specifications, see Fig. 2.3. A closed-loop backward spatial or temporal prediction of the current signal is computed and subtracted from the original. This prediction error is then transformed, quantized and entropy-encoded. The standard requires a binary input signal representation and generates in turn binary channel symbols, i.e. a bit stream. The corresponding decoder diagram is illustrated in Fig. 2.4.

To reduce complexity requirements, H.264 works on so-called MBs which consist of one 16×16-pixel luminance (luma) block and — assuming a $YCbCr$ color space and 4:2:0 chrominance (chroma) subsampling of the input signal — two blocks of 8×8 pixels for the color components. The blocking artifacts caused by the MB approach are reduced by an adaptive in-loop anti-blocking filter which applies non-linear filtering to all block edges.

H.264 consists like its predecessors of several sets of algorithms, also

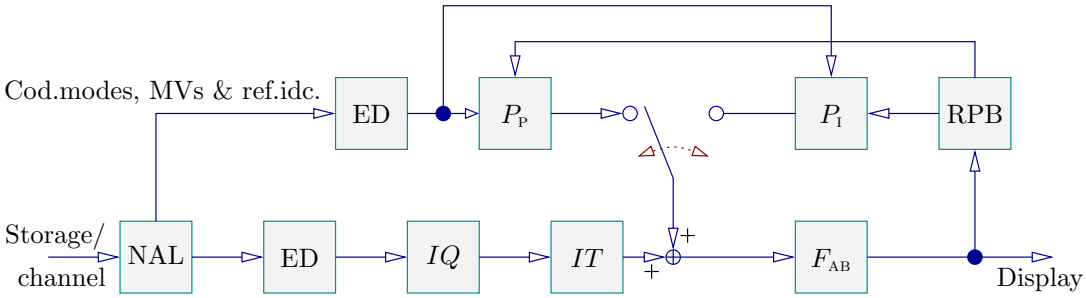


Figure 2.4 — Flow diagram of an H.264 Baseline-compliant decoder. The Network Abstraction Layer (NAL) splits the incoming data into side information and coefficients which are entropy-decoded (ED) separately. The functionalities of other blocks are explained in Fig. 2.3

called profiles, of which one has to be chosen for a specific application. Each algorithm incorporated by a particular set contributes to a small extend to the overall impressive savings in favor of H.264 as compared to previous standards. The specification of a particular profile is complemented by a so-called level which defines limitations on parameter values like picture size and processing rate.

So far, there are three profiles. Baseline is a low-complexity low-latency profile which provides basic functionality. Typical applications are interactive ones like mobile video telephony and video conferencing. The Main profile targets studio, broadcast, and other high-quality applications like HDTV and DVD. It is a high-complexity high-latency profile without any error-resilient features. Finally, Extended is highly error-robust and of low complexity, and may be used in e.g. streaming applications. All profiles have Baseline compatibility, which is also assumed used[2] throughout this work if not mentioned otherwise.

Spatial prediction, also known as INTRA or I coding, is based on the samples of previously encoded blocks of the current picture to predict the sample values of the current block, hereby exploiting spatial redundancy. There are various directional modes, of which one is illustrated in Fig. 2.5. Depending on the samples available for reference, H.264 supports nine regular modes with 4×4-pel luminance blocks, four modes with 16×16-pel luminance blocks to improve coding of large uniform luminance

[2]The SNR scalability feature was originally developed for low-delay applications, which turned out not to be feasible in Chap. 3. Hence, the Baseline profile is primarily used in this work.

areas, and four modes with 8 × 8-pixel chrominance blocks. The 4 × 4-pel prediction modes are differentially coded and then, together with the other modes, passed to the Network Abstraction Layer (NAL). The spatial prediction process operates quite efficiently; it leads to an I frame coding performance comparable to that of JPEG2000 [Hal02c].

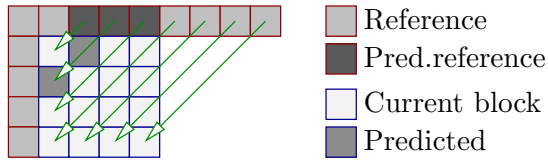


Figure 2.5 — Spatial prediction of luminance samples with designation 'INTRA 4 × 4 diagonal down left'

The signal's temporal prediction, also referred to as INTER or P coding, is made from samples belonging to a previously decoded picture. The two processes motion estimation (ME) and motion compensation (MC) operate on blocks of variable size to adapt precisely to the motion within an image sequence. MBs and 8 × 8-pixel blocks can be divided into subblocks with one or both dimensions cut into halves as depicted in Fig. 2.6. This leads to a hierarchical tree-structured motion segmentation with possible block sizes ($x \times y$) 16×16, 16×8, 8×16, 8×8, 8×4, 4×8, and 4×4 pixels. All subblocks within one MB must be of the same type (I/P).

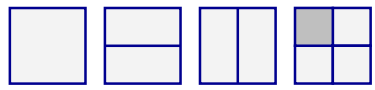


Figure 2.6 — Possible block splits of 16 × 16- and 8 × 8-pixel blocks. The gray-shaded block is thus either of size 8 × 8 or 4 × 4 pixels

The quarter-pel accuracy of the ME/MC process is achieved by the combination of a six-tap filter with the coefficients (1, -5, 20, 20, -5, 1) and a two-tap filter defined by (1, 1). Along image boundaries, the samples are extrapolated. The temporal prediction of the chrominance signal employs bilinear interpolation and a re-use of the luminance motion vectors. All motion vectors are differentially encoded by means of a median estimate or, in the case of 8 × 16- and 16 × 8-pixel blocks, a direct estimate.

The reference samples can be taken from several frames, which is also named the concept of multiple reference frames. This concept necessitates the conveyance of the index to the reference frame for each block down

to the size 8×8 pels. It is stressed that the maximum number of motion vectors per MB can be 16.

The main transform in H.264 is a separable DCT-approximating 4×4-pel integer transform, which has the advantage that problems like coefficient drifting and encoder/decoder mismatch are avoided. Associated with the transform is an appropriate scaling later in the quantization stage. The transform matrix T_4 is given by

$$T_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix} . \tag{2.1}$$

It is — independently of the block mode — applied to the prediction error first horizontally and then vertically. There is an additional 2×2-pel integer transform available for the chrominance components and — as a second step — an additional 4×4-pel integer transform applied to the first-step DC coefficients in large uniform luminance areas, i.e. 16×16-pixel blocks. All transforms are low-complexity transforms which can be realized without multiplication and using 16-bit arithmetic.

The transformed coefficients in the encoder are quantized by means of one quantizer out of the set of 52 uniform scalar quantizers with deadzone. The quantization sets for luminance and chrominance components differ. The quantizer's step size is controlled by a quantization parameter QP. The QPs are defined such that the quantization factor doubles with an increase of the parameter by the value six, and offer a wide range of possible image qualities. There is no weighted quantization as in MPEG-2 Video because no gain could be shown for this technique so far due to the small transform size. For simplicity reasons, this work considers only the instance where the QP is held constant over one entire frame, even though it can be altered on a MB basis.

Side information and header data are encoded by a single variable-length code table, an exponential Golomb code with parameter zero. There is no need for table storage as the code is regular with a variable-length prefix of the form $00\ldots01$ and a fixed-length suffix. The first code word entries of the table are listed in Tab. 3.1. Some syntax elements have in advance to be mapped to the indices of code words due to their probability distribution, such that often occurring symbols be assigned small code word indices, which in turns results in short code words. The transform coefficients are treated differently, as explained in Sec. 2.3.2.

2.2.1 Error resilience features in H.264

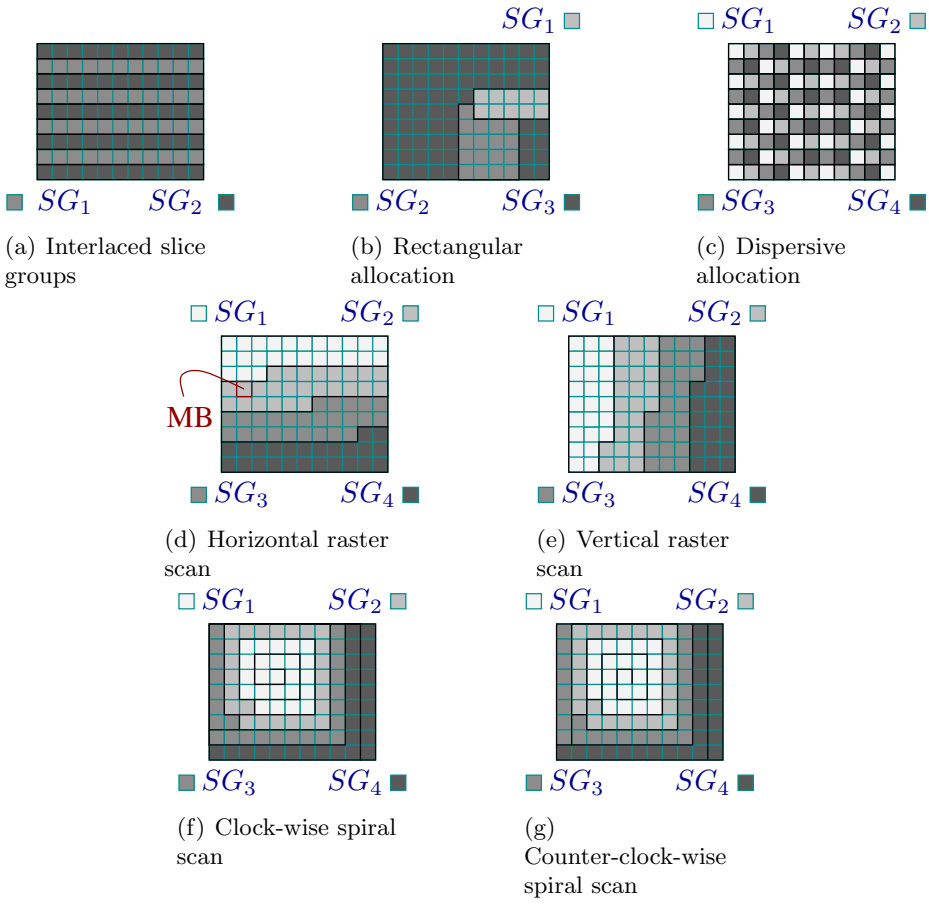
Parts of this section were presented in [HO04].

The main tool in the Baseline profile to resist transmission errors is the definition of slices and slice groups. One or several MBs are allocated to a slice group by means of an allocation map. A slice group is in turn composed of one or several slices. It is hereby possible to limit the size of slices according to the packet length requirements of a given network. Each MB belongs to exactly one slice which in turn may only consist of MBs of one picture. A slice is the smallest independently decodable unit in an encoded video. As such, any prediction referring to a spatial location beyond slice boundaries is not allowed, and e.g. the respective samples and motion vectors are marked as not available. The use of slices is aimed at stopping spatial error propagation, but simultaneously the coding efficiency is reduced as the prediction gain decreases.

H.264 knows effectively six different slice group (SG) allocation types, also known as flexible MB ordering (FMO). First, there is either a horizontal or vertical raster scan of MBs (Fig. 2.7(d) and Fig. 2.7(e), respectively), i.e. slice groups are filled row-/column-wise. This may be done in a forward or backward manner. A combination of both are rectangular slices which consist of contiguous areas of MBs, see Fig. 2.7(b). If carefully designed, rectangular slices do not reduce the coding efficiency so much but, nevertheless, bound the error impact to only a limited spatial area. Next, dispersive/scattered slices (Fig. 2.7(c)) are tailored for heavily interference-prone channels. Concentrated transmission errors are spread over the whole spatial plane and may efficiently be concealed by the decoder. However, this scheme reduces prediction gains significantly. The concept of interleaved slices follows the group of blocks (GOB) structure in H.263, see Fig. 2.7(a). There is also the possibility of a clock-wise or a counter-clock-wise scan as depicted in Fig. 2.7(f) and Fig. 2.7(g), respectively. If one of these concepts should not suffice, MBs can be explicitly allocated to a slice group, one by one.

According to the I and P MB principle, there are I and P slices. An I slice may contain only I MBs, and a P slice may be compound of both I and P MBs. Optionally, a *constrained* INTRA coding mechanism can be signaled to the decoder by means of a flag. If the flag is set, an I MB must not refer to samples that are generated by INTER coding, but may only rely on I-coded pixels.

The concept of multiple reference frames is an implicit error resilience feature inherent in H.264. The more reference frames there are, the faster



(a) Interlaced slice groups

(b) Rectangular allocation

(c) Dispersive allocation

(d) Horizontal raster scan

(e) Vertical raster scan

(f) Clock-wise spiral scan

(g) Counter-clock-wise spiral scan

Figure 2.7 — Allocation types of MBs to *SG*s. For simplicity of illustration, the number of slices per slice group is set equal to one

the decoder will recover after error occurrence. There are three other error resilience techniques specified in the Baseline profile. The concept of redundant slices is explained later in Sec. 2.3. In order to account for the packet arrival characteristics of e.g. IP-type networks, the decoder allows the slices to arrive in arbitrary order (ASO). All slices of one frame must, however, have arrived before the decoder can start decoding of the next frame. Finally, messages to the decoder interleaved in the code stream containing supplemental enhancement information may contain further information about the bit stream, which can be utilized e.g. by an error concealment scheme.

H.264 in its current form is designed primarily to handle losses of data

rather than corruption of data. The rationale is that any realistic scenario would involve a system layer (with respect to ISO's Open Systems Interconnection (OSI) model) which carries the video content, and along with it information about where errors and losses are likely to have occurred. Also, residual errors are often more disastrous due to coding techniques like VL coding (VLC) and prediction which both lead to serious error propagation, as discussed in Chap. 3.2.

Another and more important reason is that there is always a trade-off between error resilience and coding efficiency, and the majority in the JVT did not want to sacrifice the codec performance in the error-free case. As a result, bit error resilience is not addressed in H.264, at least so far, even though there were some attempts to do so, see e.g. [Hal01, Hal02a], and [Hal02d]. Instead, packet losses are of concern.

The NAL is the interface between the Video Coding Layer (VCL) and the underlying network. It groups the data from the VCL in so-called NAL units (NALUs) of variable length, hereby accounting for a variety of transmission and storage protocols like H.320, H.323, H.324/M, RTP/UDP/IP, MPEG-2 Systems, MP4FF, etc. In the Baseline profile, one NAL unit transports one slice. The units may be conveyed bit-wise, but mainly packet-switched networks are of concern. Examples for these are ATM networks, transmission by means of ITU-T Recommendation H.323, and IP-type networks in combination with the transmission protocols UDP and RTP. A single NAL unit is either encapsulated in one RTP packet (simple packetization), it is fragmented and sent in multiple RTP packets (NALU fragmentation), or one RTP packet may contain multiple NALUs (NALU aggregation) [Wen03]. Subsequently, one NAL unit is identical with one channel packet.

All mentioned error resilience features are tailored to the coding tools of H.264 and their vulnerability to transmission errors. Many other schemes have been proposed in the literature, and several overviews and reviews cover the most important proposals. They are [WWWK00], [WZ98], [VZW99], [BB98], and [KIK+98a].

2.3 SNR scalability schemes

The most common video compression standards nowadays are MPEG-2 Video, MPEG-4 Visual, and H.263. All of those have extensions for quality scalability, which are briefly reviewed in this section, complemented by an overview of the progressive coding mode of the JPEG standard.

MPEG-2 Video [ISO94b] provides the SNR Scalability profile. The base layer's quantization error is quantized itself with high accuracy and conveyed as enhancement layer data. The high-quality quantization error may in turn be fed back into the encoder's feed-back loop before the inverse transform. With a high-quality data feed-back, a drift problem occurs for decoding based on only base layer information because the reference for motion compensation at the encoder is a reconstructed high-quality layer, whereas at the decoder, it is the reconstructed base layer. The drift problem is reported to be acceptable up to a frame skip of three and 15 frames per group of pictures (GOP) [RH96]; however, the quality degradation is perceptible. The standard mandates the decoding loop to be based upon the high-quality layer. The advantage of such a scheme is that motion estimation and compensation is only invoked once, and that the requirements to frame memory remain unchanged; however, as a consequence, the system's coding efficiency will be suboptimum.

In an amendment, i.e. extension, of MPEG-4 Visual [ISO99], a layered coding technique called fine granular scalability (FGS) is specified. Two quality layers are supported; the base layer uses the conventional non-scalable coding technique as in basic MPEG-4 Visual, whereas the enhancement layer is computed by coding the residual between the original and the reconstructed base layer frames using bit plane coding of the *DCT* coefficients, without any form of motion compensation [Li99]. The enhancement layer stream is therefore embedded and may be truncated after any number of bits. This feature allows precise rate control and an optimum exploitation of channel resources. However, the lack of exploitation of high-quality references leads to a limited prediction gain. Another problem , which is encountered in case the enhancement layer bit stream is truncated, is that then also drift occurs since the reference frames are not entirely reconstructed.

During the standardization of version 2, H.263 was extended by Annex O[3] [ITU98] which provides, among other features, SNR scalability. This is achieved by coding the residue, computed as the difference between original and reconstructed base layer signal, formed in the spatial domain. The reference for the current high-quality layer can be a previously encoded (and decoded) high-quality layer, a lower-quality layer of the current frame, or a combination of both. The coding mode decision for all MBs of one frame is carried out on all layers [GK99], such that this method can be indicated as quite expensive due to parallel runs of the complex ME/MC processes. Furthermore, the overhead in terms of data

[3] H.263v2 is informally known as H.263+.

representing coding mode, motion vectors (MVs), etc., produced hereby is not negligible.

Finally, also the SNR scalability functionality of the still-image JPEG standard [ISO94a] should be mentioned here for later comparison to H.264's INTRA coding ability. The so-called progressive coding mode in the JPEG standard is done by either spectral selection of *DCT* coefficients or by their successive approximation (bit plane coding), or a combination of both. In a first coding pass/scan, coding of the original image sequence at a low or moderate compression ratio leads to base layer data. These data are then reconstructed and subtracted from the original signal, hereby forming the coding error of the first pass. The error is in turn coded and sent to the receiver as refinement information. Theoretically, there can be an arbitrary (but of course finite) number of layers [PM92]. The JPEG progressive coding technique has been ported to video coding in [RKK98].

Many other scalability schemes exist in the literature, [VU92] and also [GFH97] may serve as starting points for an overview of previous and current research. At the time of writing, SNR scalability in terms of several quality layers is not supported in the H.264 standard, despite the often mentioned 'network friendliness' as e.g. in [WSBL03]. Of course, the concept of *redundant slices* allows the insertion of primary and secondary slices in the bit stream (BS). If a primary slice is affected by errors, it can be replaced by an error-free redundant one, otherwise the redundant slices are discarded. Though this feature is useful in a simulcast environment, where the primary slices are coded with a high and the redundant slices with a low bit rate, multicast is not possible.

A new technique for SNR scalability is hence developed in the following as a desirable extension of H.264.

2.3.1 Previous research

Apart from the methods listed above, various SNR scalability techniques have been proposed for inclusion in H.264 during the standardization process[4]. All proposals except for [SMW03] are based on two layers.

In [HBM+00], the authors describe the application of plain FGS within the H.26L[5] framework. However, this contribution comes with-

[4] It is noted that the work on the SNR-scalable scheme developed in this dissertation was finalized in January 2003.

[5] H.264 is also known as H.26L.

out any technical details or results and is mainly meant as a starting point for further research.

In [BHM00] and [MBH00b], the authors propose a variant of FGS as considered in MPEG-4 Visual. The modification is that the residue is taken in the frequency domain for the whole frame and, prior to bit plane coding, decomposed into different subbands of similar statistics. However, this method suffers from a loss in efficiency relative to a single-layer scheme. The contribution was later complemented in [IB00, MBH00a, MBH00a, MW00], where different authors present results for a comparison between the FGS scheme and simulcast. The comparison is in favor of FGS which outperforms simulcast in terms of quality at higher bit rates for a fixed channel bandwidth and for almost all rates when the bandwidth is variable.

A different approach is followed in [HWG00], which bases on conventional SNR scalability. The authors form the residue of base and high-quality layer in the spatial domain and, under encoding, utilize both the low- and high-quality reconstructed frames for temporal prediction purposes. There are separate loops for MC in the encoder and the decoder, but ME is executed only once, i.e. coding mode information and motion vectors of the base layer are re-used on higher-quality layers. This technique accepts a suboptimal coding performance when all layers can be conveyed error-free, as utilizing more high-quality reference frames would increase the prediction gain. In case of errors, a drift problem, here called leaky prediction, occurs because the high-quality layer is not available for reference at the decoder.

Another extension of MPEG-4 Visual's FGS scheme is proposed for use in H.26L in [HYWL01]. The authors add a second motion compensation loop for the high-quality layer and base the temporal prediction of that layer on both layers. The coding performance of this method is increased with regard to MPEG-4 Visual FGS, but the drift problem persists.

A new trial to include FGS in H.26L was undertaken in [PLC+02], this time combined with ROI coding. The residue is formed in the spatial domain as usual, but then the authors scan the coefficients in a spiral scan manner before they are passed to the bit plane coding engine.

A very interesting new perspective to the relationship of hybrid coding and 3-D coding is given in [SMW03]. The authors present a filter bank, of which the analysis step corresponds to a conventional encoder (H.264 Baseline-compliant) and the synthesis step to the decoder. The filter bank

is realized with an open-loop motion compensation and implemented by means of lifting structures. Among temporal and spatial scalability, this approach offers the functionality of SNR scalability. The scheme's coding efficiency depends strongly on the input source but is at most almost comparable to that of H.264 as presented in this thesis. Moreover, a single-layer version has been shown to provide a performance superior to H.264's compression ability [SMW04].

The SNR-scalable scheme proposed later in Sec. 2.4 has not been contributed to the JVT as the standardization process of H.264 had already evolved too far towards a finalization.

2.3.2 H.264's entropy coding

To extend H.264 by the feature SNR scalability, a look on the standard's entropy coding technique is required.

In the Baseline profile, low-complexity entropy coding of the quantized transform coefficients is specified, so-called context-adaptive variable-length coding (CAVLC). CAVLC exploits the coefficients' statistical correlation by first scanning them in a zig-zag manner into a one-dimensional array. Every non-zero coefficient is then associated with a variable *run* which counts the number of zero coefficients to the previous non-zero coefficient, also denoted as *levels*. H.264 follows hereby the strategy of other standards like MPEG-2 Video or H.263, but in contrast to the 2-D and 3-D VLC techniques employed in these standards, level and run are treated separately in H.264.

It can be observed that there are very often 1's with either sign among the highest-frequency coefficients. These are recorded in number (up to three) and, together with the total number of non-zero coefficients, coded with one out of a set of code tables. The decision which table to use is made with regard to the number of non-zero coefficients in neighboring already encoded blocks. Additionally, the sign of the 1's is conveyed to the decoder. The values of the remaining coefficients is then coded using adaptive Rice codes, where the adaptivity is given by a varying suffix size to adapt to the coefficients' frequency range. That is, several code tables are used, and the choice among the tables is made according to the value of the previously encoded coefficient. After that, the sum of run's is computed and encoded with one out of 15 tables depending upon the number of non-zero coefficients in that block. Now, the only thing that remains is to code the individual run values with one out of seven code tables, depending upon the remaining sum of run's. All code tables used

by CAVLC have been generated empirically.

Consider as an example the 1-D array of coefficients (12, -7, 0, 0, 5, 1, -1, 0, -1, 1, 0, ..., 0) as derived from the 2-D coefficient matrix Y before zig-zag scan,

$$Y = \begin{bmatrix} 12 & -7 & 1 & -1 \\ 0 & 5 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}, \tag{2.2}$$

from which the sequence of non-zero coefficients (12, -7, 5, 1, -1, -1, 1) and the associated run's (0, 0, 2, 0, 0, 1, 0) are extracted. The number of non-zero coefficients is 7, and the number of trailing 1's is 3, leading to the 2-tuple (7,3) which is variable-length-encoded. The signs of these coefficients is (+, -, -), each signaled with one bit to the decoder. The sum of run's is $2+1=3$. Finally, the remaining coefficients (1, 5, -7, 12) and the respective run's (0, 1, 0, 0, 2, 0, 0) are encoded in a backward manner. The decoder determines the position of the highest-frequency coefficient by adding the number of non-zero coefficient and the sum of runs, assuming that array indexing starts with one. The trailing 1's and other coefficients can hereafter be placed according to the successively decoded individual run's.

CAVLC is a quite efficient method to code the quantized and scanned luminance and chrominance transform coefficients. During H.264'a standardization, it was proposed in a series of technical reports [Bjø02, BL02, Au02, Lil02, AKS+02] to replace the former UVLC/EXPG codes (see Tab. 3.1) used to code symbol pairs of levels and run's, so-called 2-D VLC. The use of CAVLC yields a performance which is approximately 33% better than that of entropy coding by plain UVLC or EXPG [BL02].

As it is seen below, the improved performance in entropy coding is also the key to efficient coding of quality layers in H.264.

2.4 SNR scalability extension of H.264

In this section, a new scheme which is suitable for SNR scalability in H.264 is developed. To keep both the system simple and the amount of computation for simulations reasonable, the focus is on the instance of one base layer (BL) and one high-quality layer (HQL). Parts of the section have been presented in [HF03b].

Due to the energy compaction property of most signal transforms in video coding, one would expect that schemes which form the coding error

in the frequency domain be more efficient than schemes which compute the error in the spatial domain. Therefore, the difference signal is computed in the frequency domain *and* — in contrast to e.g. MPEG-2 Video — with respect to the quantized transform coefficients because of their small dynamic range. In other words, the transform coefficients of the second coding pass are predicted by the coefficients of the base layer, and the difference is then lossless entropy-encoded. A similar layered coding algorithm based on partitioning of *DCT* coefficients was devised in [KK01], but with a rate constraint in contrast to the quality constraint for the computation of coefficients as used here. Apart from this, no higher-quality layer motion compensation is performed, which impairs the coding gain, especially for high-motion videos. The work can, however, show that the concept of refinement can more efficient than a framework based on data partitioning as in the JPEG standard.

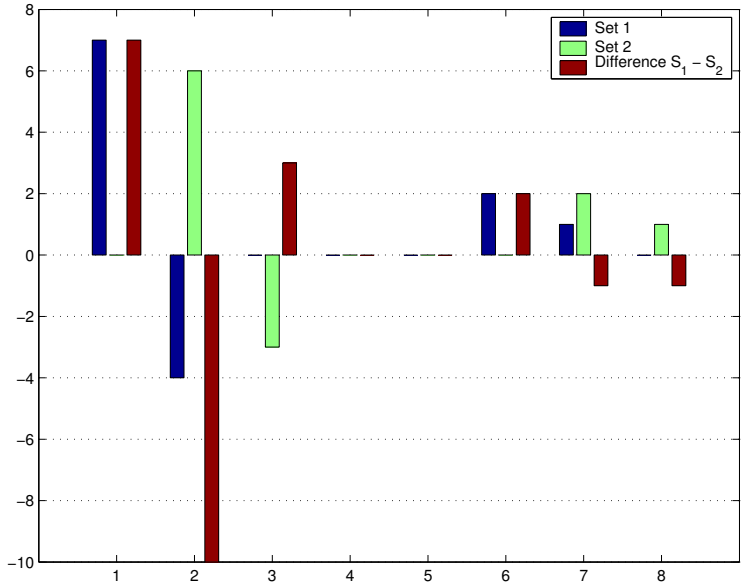


Figure 2.8 — Level over the first eight scan positions of two coefficient sets S_1 and S_2, including the difference set. The remaining eight coefficients are assumed zero

The process is illustrated with an example in Fig. 2.8. The first set S_1 of scanned coefficients consists of the levels (7,-4,2,1) and run's (0,0,3,0), and the levels and run's of the second set S_2 are (6,-3,2,1) and (1,0,3,0), respectively. The difference set consists of the level sequence (7,-10,3,2,-1,-1) and the corresponding run's (0,0,0,2,0,0). The approach of coding

the transformed and quantized prediction error difference

$$\Delta E_{\text{pred}} = Q\{T\{E_{\text{HQL}}\}\} - Q\{T\{E_{\text{BL}}\}\}\,, \tag{2.3}$$

where E with different subscripts denotes the prediction error in the respective layers, is more efficient than coding the reconstruction error

$$\Delta X_{\text{rec}} = X - \tilde{X}_{\text{BL}}\,, \tag{2.4}$$

with regard to the original signal X at the encoder input and the reconstructed signal $\tilde{X}$ at the decoder output, because of the low correlation between $Q\{T\{E_{\text{HQL}}\}\}$ and $Q\{T\{E_{\text{BL}}\}\}$, especially for large

$$\Delta QP = QP_{\text{BL}} - QP_{\text{HQL}} \qquad (QP_{\text{BL}} > QP_{\text{HQL}})\,. \tag{2.5}$$

On the one hand, correlated sets S_i lead to long sequences of small difference levels with their associated run's, both of which are not very efficiently entropy-encoded. This is because H.264 conveys a variable CBP in the bit stream, which as a flag signals to the decoder for all 8×8-pel subblocks of each MB which of the six 4×4-pel blocks (four for the luma component and two for the chroma components) contain non-zero coefficients. If CBP flags non-zero coefficients, the entropy encoder as explained in Sec. 2.3.2 is invoked for that particular 4×4-pel block, otherwise it is leaved out. The CBP improves hereby greatly the coding gain. On the other hand, uncorrelated coefficient sets are expected to be encoded well due to the excellent performance of the CAVLC. The efficacy of the approach is hence dependent on the performance of the entropy-encoding module and the statistics of the scan of coefficient differences.

H.264 yields an excellent coding gain [Hal03a] by exploitation of the source signal's spatial and temporal redundancy as much as possible. That is, it aims at minimization of the energy of the prediction error, i.e. the transform coefficients. The cost of this strategy is the large amount of bits, which data other than the coefficients consume. These data compound of header data, data containing coding mode, motion vector and reference indices, CBP, and other, and are in the following denoted as side information. As an example, the QCIF-size video `Foreman`, coded with IPP GOP structure at 30 fps, results in a bit stream where the average amount of side information per frame, normalized by the total number of bits spent per frame on the average, is 7% for an I frame, and 63% for an P frame.

Thus, unlike other approaches like [GK99], the strategy followed here is to re-use side information of one layer in all other layers. As a consequence, the coding mode has to be consistent throughout all layers as well. This leads to a quality degradation for the layers for which the mode decision has not been optimized; however, the decrease is typically moderate (less than 1 dB. To continue the above example, `Foreman` is coded with two layers and $QP_{\mathrm{BL}} = 35$ and $QP_{\mathrm{HQL}} = 30$, first where the coding mode decision is carried out for the base layer, denoted as BL optimization (BLO), and then for the high-quality layer, denoted as HQLO. The results are given in Tab. 2.1. All values vary a little, which means that mode decision changes for some MBs, but far from all. The base layer's values vary less than those of the high-quality layer due to the coarser quantization which determines the quantizer's reconstruction levels regardless small coefficient variations caused by different MB coding modes. The exact $PSNR$ difference depends on the source statistics, coding parameters like number of reference pictures, and the coding mode evaluation scheme.

The encoder should make the layer choice for coding mode decision dependent on the application in mind. For video on demand, the highest-quality layer should be chosen, which gives the best coding performance. If, however, the HQL is not always available at the decoder as after error-prone transmission, temporal and spatial prediction should be based on the base layer.

Layer	BLO	HQLO
BL	32.19	32.03
HQL	34.93	35.51

Table 2.1 — Layer $PSNR$ (in dB) with BL- and HQL-optimized coding mode decision

The encoder will preferably select the prediction mode according to a Lagrange rate distortion optimization criterion, which in turn is based on the true encoded rate and the true distortion of each block and each prediction mode [SW98]. This mode selection achieves the optimum rate-distortion performance of the codec but is highly complex. A simplified method employed here is therefore to choose the one prediction mode for which the best trade-off between the coding costs source bit rate and distortion in terms of for instance the sum of absolute transformed difference ($SATD$) is accomplished. This can e.g. be done by Lagrange functional minimization [Eve63], where the coding costs are linearly combined and

jointly minimized,

$$J = SATD + \lambda R\,, \tag{2.6}$$

with a rate constraint R and the Lagrangian multiplier λ as optimization parameter. Relative to rate-distortion optimization, a loss in coding efficiency has hereby to be accepted. The $SATD$ for a 4×4-pel block is defined as

$$SATD = \frac{1}{2} \sum_{i,j=0}^{3} \left| T_H\{E(i,j)\} \right| . \tag{2.7}$$

$T_H\{\cdot\}$ is the 2-D Hadarmard transform, and the definition of the prediction error is

$$E(i,j) = X(i,j) - \hat{X}(i,j)\,, \tag{2.8}$$

with the original and predicted samples X and $\hat{X}$, respectively, at pixel position i in line j. The minimization is done for all intra- and inter-frame MB coding modes, the latter one implying all reference frames.

To avoid drift as in MPEG-2 Video, the enhancement layer requires its own reference picture buffer and that prediction, transform, and quantization be performed independently of the base layer. Another option would be to base the prediction on the base layer's buffer, but experiments with various image material showed a rate increase of up to 80% due to the low quality of the reconstructed frames in this buffer; thus this idea was discarded. It is mandatory for both layers to operate with the same coding mode, i.e. MB type, since the two sets of transform coefficients are subtracted from each other. Differing modes like INTRA 16×16 and INTRA 8×8 would result in different number of coefficients sets due to the DC transform as explained in Sec. 2.2.

Consistent MB types means in particular that the time-consuming motion estimation be done for only one layer. As already mentioned, for wired streaming and videoconferencing systems, the encoder should optimize the mode decision with regard to the high-quality layer because it expects the application to receive ideally both code streams. For broadcasting applications and error-prone environments, mode decision should be chosen to be optimized with respect to the base layer to be able to guaranty a minimum image quality (i.e. QoS) under erroneous transmission. In the sequel, the high-quality layer is chosen as basis for mode decision, if not mentioned otherwise. Fig. 2.9 shows the simplified diagram of the SNR-scalable encoder. It is stressed that, with the proposed two-layer technique and HQLO, the high-quality layer reconstructed by QP_{HQL} is identical with a single layer which has been reconstructed setting $QP = QP_{\text{HQL}}$.

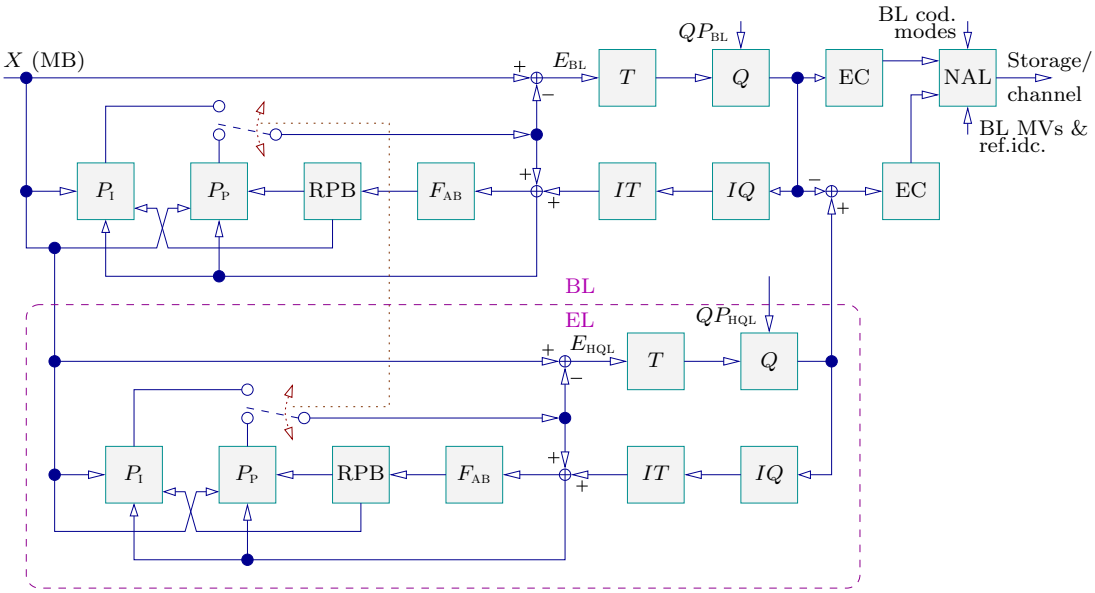


Figure 2.9 — The extended encoder. The base layer is completely standard-compliant (Baseline profile), i.e. channel packets with base layer data form a valid bit stream. In fact, the same applies to the enhancement layer coding unit. All acronyms are explained in Fig. 2.3

Due to the MB approach of H.264, the enhancement layer does not necessarily have to cover the whole spatial plane but can be computed for one or several specific areas within, hereby allowing for ROI coding. These areas have to begin and end at MB boundaries because of the largest prediction mode which is of size 16×16 pixels both with INTRA and INTER coding.

The use of a 1-bit ROI array/mask is suitable for specification of the region of interest, consisting of one's to signal ROI MBs and of zeros elsewhere, resulting in 99 data bits per frame for QCIF images, which have to be conveyed to the decoder. The ROI mask can, but does not have to, be updated for every video frame. This concept is very flexible as it allows for transmission of several areas of differently increased quality. However, a QP or ΔQP has to be signaled for each ROI. As the increase of overhead means a decrease in coding efficiency, the implementation is in the following limited to one ROI, i.e. the whole frame, with one fixed QP_{HQL}. An example for a high-quality layer which does not cover the whole picture is shown in Fig. 2.10.

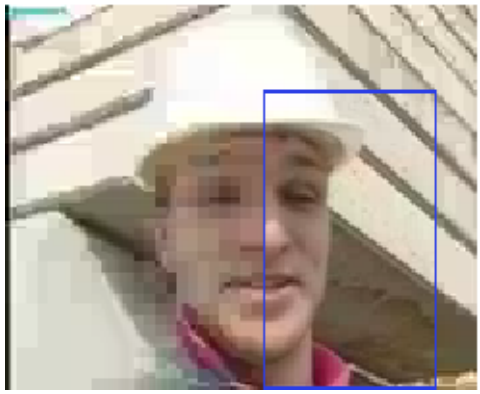


Figure 2.10 — ROI coding is possible despite the fact that coefficient prediction is done in the frequency domain due to the MB approach in block-based hybrid coding. The foreground marked by the rectangle is coded by $QP_{\mathrm{HQL}} = 20$, the background is coded with $QP_{\mathrm{BL}} = 40$

The encoder outputs, in addition to the base layer bit stream, a separate enhancement layer stream which consists of a CBP for each MB and the entropy-encoded transform coefficient differences for all color components, and both DC and AC coefficients. As H.264 works on a slice basis, also the MBs of the enhancement layer are conveyed slice-wise and hence preceded with a unique slice start code, hereby adding three bytes to every slice. On this way, the enhancement layer can be transmitted packet-wise with one slice per packet, and the system is therefore appropriate for use in error-prone environments and packet-switched networks.

If available, the refinement information is added to the transform coefficients on the decoder side for high-quality image reconstruction. All lower layers can be decoded independently of higher-quality layers such that a clear structure of dependency relations among slices can be established as done later in Sec. 3.4. The diagram of the SNR-scalable decoder is shown in Fig. 2.11. As already mentioned, the scheme presented is not bound to two layers but can easily be extended to any desired number of layers, depending on the application and with respect to complexity, memory, and latency constraints.

A last issue is the anti-blocking filter and its operation along MB and slice edges. The SNR-scalable codec must reflect the fact that the filtering process is — among other parameters — subject to the QPs of two adjacent macroblocks, as explained in [LJL+03]. That is, an additional

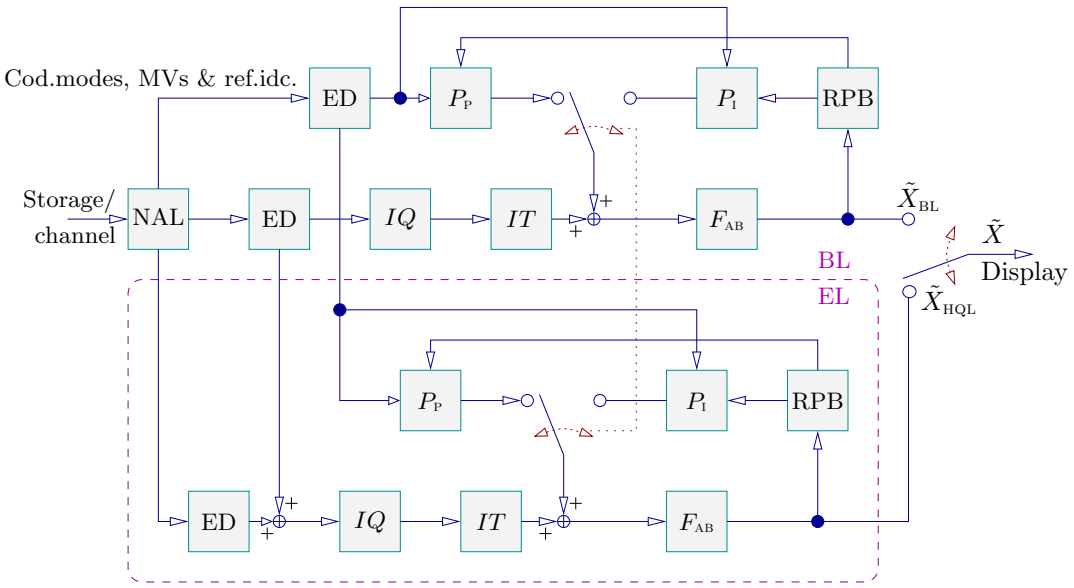


Figure 2.11 — The extended decoder. All acronyms are explained in Fig. 2.4 and Fig. 2.3

high-quality filtering is required. Considering ROI, $F_{AB}^{(HQL)}$ should operate with respect to both QP_{HQL} and QP_{BL}, otherwise only QP_{HQL}. However, due to reasons given in Sec. 3.6 (error propagation issues), it was chosen not to use F_{AB} throughout this work. Hence, all results from the following sections and chapters imply a disabled anti-blocking filter, hereby also enhancing the tractability of results. As a consequence, the results presented here have the potential for further improvements in terms of rate decrease of 5–10% which are estimated with respect to the bit rate savings reported for the anti-blocking filter in [WSBL03]. It is stressed that a disabled anti-blocking filter is still 100% standard compliant since filtering can be turned on and off on a sample level.

Out from what was discussed above, the efficiency of the proposed scalability technique is expected to outperform previous schemes in a rate distortion sense. In how far this turns out to be true is investigated in the following section.

2.5 Experiments, results, and discussion

This section provides a detailed performance evaluation of the H.264 codec extended by SNR scalability. Parts of the section have been presented in [HF03b].

The scheme developed in the previous sections — called SSH4 subsequently — is implemented in both encoder and decoder of the JVT's reference software JM-4.0d[6]. The corresponding preliminary standard document is the Joint Final Committee Draft (JFCD) [Wie02]. The software is operated in the Baseline profile due to its low complexity, low latency, and the set of supported error resilience features. The system performance is here measured in terms of rate increase which is defined as negated rate savings, as specified Sec. C.1.6. The image material as listed in App. A is used for all experiments at 10 fps if not mentioned otherwise, i.e. the frame skip is two. This implies that all results given subsequently are averaged over the whole image sequence; that is, over all non-skipped frames. If not noted otherwise, the $PSNR$ values are with respect to the luminance components only, whereas rate values also include the color components. Values regarding to INTER-frame coding are averaged over additionally the initial INTRA frame.

Layer	QP_{BL}				
	17	20	25	35	45
BL	676.61	540.61	361.49	150.02	84.35
EL	387.48	549.74	657.14	713.12	720.92
Total	1064.09	1090.35	1018.63	863.14	805.27
$PSNR_{\mathrm{BL}}$	45.31	42.79	38.93	31.53	25.21

Table 2.2 — Rate (in *kbps*) distribution of the double-layer system with image sequence `Foreman` (QCIF-size), III coding mode, and HQLO. Also given is the quantization distortion of the base layer, $PSNR_{\mathrm{BL}}$ (in *dB*). $QP_{\mathrm{HQL}} = 15$, and $PSNR_{\mathrm{HQL}} = 46.98\,\mathrm{dB}$

The number of reference frames are set to one for all experiments, and the size of one slice equals one frame. The ΔQPs considered here are $\{2, 5, 10, 20, 30\}$. The QP increment of five has been chosen because the

[6]The changes introduced in versions later than JM-4.0d are of no importance for the results in this work. The rate distortion comparisons in [WSBL03] show that JM-4.0d can claim to have all coding tools relevant for the optimum coding gain implemented. The latest version of JM is v8.1 (May 2004).

bit stream sizes of base layer and high-quality layer are approximately equal for $\Delta QP = 5$. The results are limited to the cases of two and three quality layers to keep simulation durations low and, more importantly, because these are the most probable settings in real-life applications. The MB coding mode is decided on with respect to the high-quality layer, denoted as HQLO.

Layer	QP_{BL}				
	17	20	25	35	45
BL	2541.31	1967.95	1264.37	498.33	323.52
EL	1664.42	2131.76	2486.28	2664.80	2681.40
Total	4205.73	4099.71	3750.65	3163.13	3004.92
$PSNR_{\mathrm{BL}}$	43.33	40.63	36.60	28.75	22.35

Table 2.3 — Rate (in *kbps*) distribution of the double-layer system with image sequence `Mobile&Calendar` (CIF-size), IPP coding mode, and HQLO. Also given is the quantization distortion of the base layer, $PSNR_{\mathrm{BL}}$ (in dB). $QP_{\mathrm{HQL}} = 15$, and $PSNR_{\mathrm{HQL}} = 45.16\,\mathrm{dB}$

2.5.1 Two quality layers

Considering two layers, Tab. 2.2 and Tab. 2.3 show how the total rate is distributed to each layer under a quality constraint. As QP_{BL} increases for a fixed QP_{HQL}, more and more energy is shifted into the high-quality layer, which means a rate increase. The total rate, however, decreases simultaneously. This can be explained by the fact that almost equal QPs produce similar transform coefficient scans and hereby an enhancement layer which consists of many one's and series of zeros within. Both are not efficiently entropy-encodable in a rate distortion sense, as pointed out above. This is consistent for both I and P frames.

In Sec. 2.1, it was discussed how multicast applications utilize layered coding. The aim is to maximize the number of layers received at each decoder. The increase ΔR of the total rate as compared to the reference system, the single-layer codec, is thus of major interest. The comparison with regard to a double-layer system is tabularized in Tab. 2.4 through Tab. 2.9.

The values of ΔR in the tables show some interesting tendencies. First, the bit rate increase varies significantly from quite large for small ΔQPs (here: two) to very small for large ΔQPs (e.g. 30). As above,

QP_{HQL}	Fr. tp.	QP_{BL} 17	20	22	25	27	30	32	35	37	40	42	45
15	I	38.6	41.2		31.9				12.2				3.9
15	P	42.8	27.5		14.1				2.7				0.7
20	I	–	–	36.0	38.6		27.7				9.9		
20	P	–	–	51.8	33.6		15.7				4.3		
25	I	–	–	–	–	35.2	35.0		23.7				8.2
25	P	–	–	–	–	56.4	36.2		19.8				10.7
30	I	–	–	–	–	–	–	34.3	32.7		21.1		
30	P	–	–	–	–	–	–	76.1	54.9		38.9		
35	I	–	–	–	–	–	–	–	–	36.0	32.4		19.4
35	P	–	–	–	–	–	–	–	–	117.4	97.9		82.2
40	I	–	–	–	–	–	–	–	–	–	–	38.2	32.0
40	P	–	–	–	–	–	–	–	–	–	–	185.3	172.4

Table 2.4 — The rate increase (in %) of I and P frame types in layered coding with video `Container` (CIF-size), two layers, and HQLO

QP_{HQL}	Fr. tp.	QP_{BL} 17	20	22	25	27	30	32	35	37	40	42	45
15	I	36.6	38.7		28.7				9.4				2.8
15	P	37.7	27.4		15.4				3.0				0.7
20	I	–	–	34.2	35.4		23.8				7.4		
20	P	–	–	40.2	28.7		14.4				3.5		
25	I	–	–	–	–	33.6	31.2		19.7				6.4
25	P	–	–	–	–	39.9	27.3		14.7				6.5
30	I	–	–	–	–	–	–	33.7	29.4		17.3		
30	P	–	–	–	–	–	–	47.4	34.4		22.0		
35	I	–	–	–	–	–	–	–	–	34.8	28.7		16.6
35	P	–	–	–	–	–	–	–	–	58.5	46.0		38.6
40	I	–	–	–	–	–	–	–	–	–	–	35.3	27.8
40	P	–	–	–	–	–	–	–	–	–	–	68.1	60.2

Table 2.5 — The rate increase (in %) of I and P frame types in layered coding with video `Foreman` (CIF-size), two layers, and HQLO

this is explained by the great correlation of coefficient scans discerned by a small ΔQP, and less coefficient set correlation determined by a large ΔQP. The poorer the quality of the BL, the more of its coefficients are set to zero, and the more the total bit rate converges towards the single-layer rate. The same observation has also been made in [WG97].

Next, there is a rate increase gradient which depends on ΔQP, QP_{HQL}, the frame type, and the image material. For $\Delta QP = 2$ and I frames, ΔR is nearly constant with almost all test videos. The exception is `Mobile& Calendar`, where the great amount of detailedness is sufficiently quantized with a high accuracy of small QPs, such that a further QP reduction does not produce more details, whereas a coarse quantization by two adjacent

QP_{HQL}	Fr. tp.	QP_{BL} 17	20	22	25	27	30	32	35	37	40	42	45
15	I	42.2	48.6		42.0				19.6				6.5
15	P	41.7	38.1		26.3				6.6				1.2
20	I	–	–	37.8	44.4		36.4				14.9		
20	P	–	–	41.5	35.0		20.7				4.5		
25	I	–	–	–	–	35.9	39.7		30.1				11.1
25	P	–	–	–	–	40.9	30.2		15.6				3.7
30	I	–	–	–	–	–	–	32.5	34.8		25.2		
30	P	–	–	–	–	–	–	45.7	30.4		15.3		
35	I	–	–	–	–	–	–	–	–	30.7	31.5		21.0
35	P	–	–	–	–	–	–	–	–	56.2	37.3		22.8
40	I	–	–	–	–	–	–	–	–	–	–	30.9	28.7
40	P	–	–	–	–	–	–	–	–	–	–	74.3	59.9

Table 2.6 — The rate increase (in %) of I and P frame types in layered coding with video `Mobile&Calendar` (CIF-size), two layers, and HQLO

QPs gives two coefficient sets which differ more than in the former case. For P frames, the ΔR gradient at sufficiently small QP_{HQL}s is more distinct than the corresponding gradient of I frames. The explanation here is that the coefficient scans of P frames, coded with quite different QPs, is less correlated than the scans of I frames due to a more accurate inter-frame than intra-frame prediction in H.264.

QP_{HQL}	Fr. tp.	QP_{BL} 17	20	22	25	27	30	32	35	37	40	42	45
15	I	34.6	33.1		23.6				7.9				3.0
15	P	40.5	21.7		10.5				2.4				1.2
20	I	–	–	34.4	32.4		20.9				7.2		
20	P	–	–	48.7	31.9		17.6				8.7		
25	I	–	–	–	–	34.5	29.7		18.4				7.4
25	P	–	–	–	–	60.6	45.7		33.4				26.7
30	I	–	–	–	–	–	–	36.0	30.6		18.8		
30	P	–	–	–	–	–	–	88.4	75.6		63.9		
35	I	–	–	–	–	–	–	–	–	37.0	30.9		18.4
35	P	–	–	–	–	–	–	–	–	119.7	111.7		102.6
40	I	–	–	–	–	–	–	–	–	–	–	36.2	27.1
40	P	–	–	–	–	–	–	–	–	–	–	147.9	148.6

Table 2.7 — The rate increase (in %) of I and P frame types in layered coding with video `Mother&Daughter` (CIF-size), two layers, and HQLO

Very remarkable is further the meaning of the enhancement layer CBP for the efficiency of layered INTRA coding in terms of ΔR: With an increase of QP_{BL}, ΔR increases first somewhat towards a maximum around $\Delta QP = 5$, before its value shrinks monotonically. A small ΔQP

produces quite equal coefficient sets, and if they are identical, the CBP signals an empty enhancement layer without invoking the entropy encoding subsequently, hereby saving rate. The effect vanishes with a rising QP_{HQL}, as then also the base layer CBP contributes to bit rate savings. For INTER coding, there is no such effect, explained by its higher prediction gain as compared to INTRA coding.

QP_{HQL}	Fr. tp.	QP_{BL}											
		17	20	22	25	27	30	32	35	37	40	42	45
15	I	36.8	40.1		30.9				10.9				3.5
15	P	40.0	29.5		17.3				4.0				0.9
20	I	–	–	34.1	36.0		25.6				8.5		
20	P	–	–	41.2	30.0		15.9				4.2		
25	I	–	–	–	–	33.4	31.9		21.1				7.4
25	P	–	–	–	–	40.2	28.2		15.7				6.7
30	I	–	–	–	–	–	–	33.5	30.0		18.6		
30	P	–	–	–	–	–	–	46.9	34.4		22.5		
35	I	–	–	–	–	–	–	–	–	35.2	29.4		18.6
35	P	–	–	–	–	–	–	–	–	60.3	47.7		36.6
40	I	–	–	–	–	–	–	–	–	–	–	36.4	30.0
40	P	–	–	–	–	–	–	–	–	–	–	70.9	61.6

Table 2.8 — The rate increase (in %) of I and P frame types in layered coding with video `Foreman` (QCIF-size), two layers, and HQLO

It is very interesting that, with an extremely large ΔQP, the rate of the double-layer scheme converges towards the rate of the single-layer codec. To show this fact more clearly, the rate distortion curves of both codecs with different QPs are determined, as shown in Fig. 2.12. The five points of the curve labeled $\Delta QP = 0$ correspond — in order of decreasing $PSNR$ — to the single-layer QP ranging from 10 to 30 with an increment of five. It is clearly seen that the rate increase varies strongly depending on ΔQP at high communication rates (> 500 kbits/s) and less at low total rates (< 200 kbits/s). The observation of converging bit rates gets support when ΔR also for very high (> 35) ΔQPs is plotted, which is only possible at high rates due to the fact that the range of scalar quantizers and hereby QP is limited by the interval $[0, \ldots, 51]$. The bit rate increase is almost 0% when the video is reconstructed by means of a base layer coded at $QP = 50$ and an high-quality layer coded with $QP = 10$. The plot is thus consistent with the first row of ΔRs in e.g. Tab. 2.4.

Unfortunately, also a limitation of the proposed layered scheme becomes visible in the results. As the tables given above reveal, coding of the low-motion videos `Container`, `Mother&Daughter`, and `Silent` generates a rate increase of considerably higher than 100% for small ΔQPs,

QP_{HQL}	Fr. tp.	QP_{BL}											
		17	20	22	25	27	30	32	35	37	40	42	45
15	I	35.8	39.6		30.4				10.2				3.0
15	P	42.0	31.7		19.4				6.0				3.0
20	I	–	–	33.4	34.9		24.1				7.6		
20	P	–	–	47.6	36.4		23.1				11.3		
25	I	–	–	–	–	32.8	30.6		19.6				6.6
25	P	–	–	–	–	51.3	40.7		29.0				20.4
30	I	–	–	–	–	–	–	34.0	30.1		18.1		
30	P	–	–	–	–	–	–	69.8	58.9		47.1		
35	I	–	–	–	–	–	–	–	–	35.4	29.9		17.6
35	P	–	–	–	–	–	–	–	–	101.1	91.8		80.1
40	I	–	–	–	–	–	–	–	–	–	–	35.3	27.9
40	P	–	–	–	–	–	–	–	–	–	–	150.8	143.2

Table 2.9 — The rate increase (in %) of I and P frame types in layered coding with video `Silent` (QCIF-size), two layers, and HQLO

which is totally unacceptable. Depending on the image material and the communication rate (and hence QP_{HQL}), ΔQP is therefore recommended to be given values larger than 5–15 to bound ΔR to decent values (below 30%). Other comments on ΔR when compared to other research are given below.

A comparison of `Foreman` of size CIF and QCIF shows that the layered scheme works equally well for either image size. The rate increase of the CIF-size material is with some few exceptions always slightly below that of the QCIF-size video. Generally, a CIF-size video is more detailed than a QCIF-size video, which means that more scan coefficients are present in the high-quality layer at high frequencies. Many of those coefficients are typically quantized to zero in the base layer, and since the zero-level coefficients appear at the end of a scan, the rate increase due to coding of the difference scan relative to the base layer is less than the increase with QCIF-size imagery.

A rate distortion comparison between base layer and high-quality coding mode optimization is given in Fig. 2.13 and Fig. 2.14. The five curve points correspond to $QP_{\text{HQL}} = \{35, 30, 25, 20, 15\}$ for the single-layer scheme and for the high-quality layer of the double-layer codec, and to $QP_{\text{BL}} = QP_{\text{HQL}} + \Delta QP$ with $\Delta QP = 10$ for the base layer. The optimized layer achieves always a higher $PSNR$ and a lower rate R than the non-optimized layer. This effect is more distinct for the high-quality layer than for the base layer.

To be able to compare BLO and HQLO mode decision directly, the measure RDP as defined in Sec. C.1.5 is utilized, as well as the definition

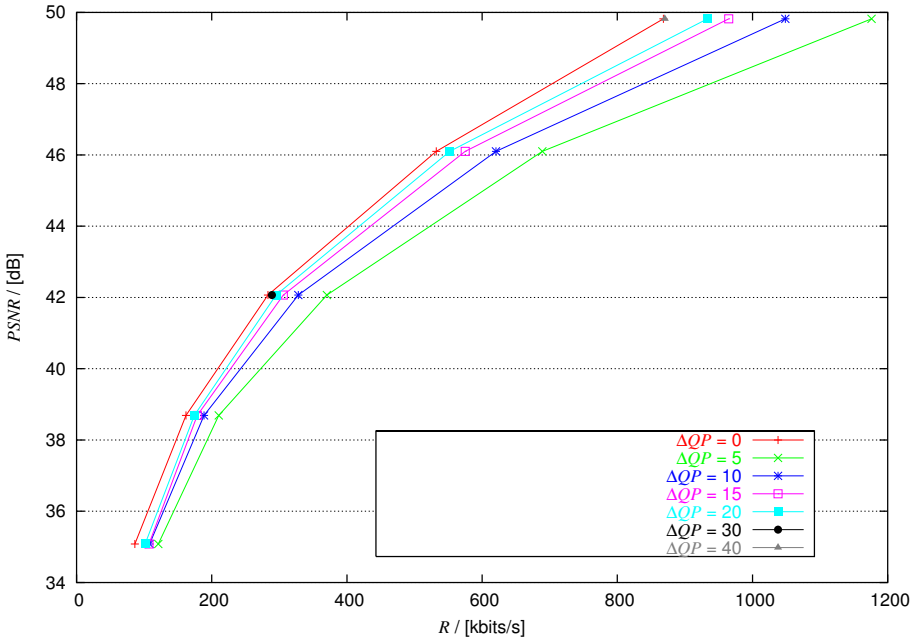


Figure 2.12 — Rate distortion comparison of the double-layer scheme versus the single-layer scheme. $\Delta QP = 0$ is the curve of the single-layer scheme. Video: `Foreman`, size: CIF

of efficiency of layered coding, given by

$$\epsilon_{\text{layered}} = \frac{RDP_{\text{non-layered}}}{RDP_{\text{layered}}} \,. \tag{2.9}$$

Values of $\epsilon_{\text{layered}}$ in non-embedded coding range in the interval from zero to one due to $RDP_{\text{layered}} > RDP_{\text{non-layered}}$ in general, where one is of course most desirable. The use of HQLO turns out to be superior to the use of BLO with all test videos, see Tab. 2.10. With `Foreman`, HQLO coding achieves $\epsilon_{\text{layered}}^{(\text{HQLO})} = 0.82$, whereas BLO coding yields $\epsilon_{\text{layered}}^{(\text{BLO})} = 0.54$. The difference is not that remarkable for other image sequences, e.g. $\epsilon_{\text{layered}}^{(\text{HQLO})} = 0.84$ and $\epsilon_{\text{layered}}^{(\text{BLO})} = 0.69$ with `Mobile&Calendar`— the curves in Fig. 2.14 are closer to each other than those in Fig. 2.13; however, $\epsilon_{\text{layered}}^{(\text{BLO})}$ cannot be better than but only converge to $\epsilon_{\text{layered}}^{(\text{HQLO})}$. For, intuitionally, the high-quality layer provides always better knowledge than the base layer about how the prediction process can exploit the video statistics.

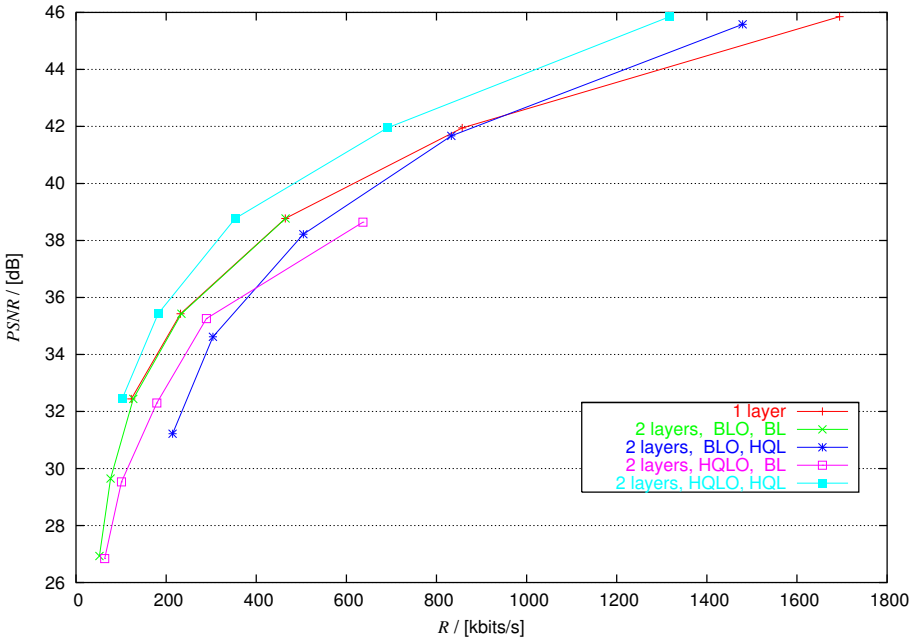


Figure 2.13 — Rate distortion comparison of various layers. Video: `Foreman`, size: CIF, IPP coding mode, $\Delta QP = 10$

Video name	$\epsilon_{\text{layered}}^{(\text{BLO})}$	$\epsilon_{\text{layered}}^{(\text{HQLO})}$
`Container`	0.61	**0.75**
`Foreman` (CIF)	0.54	**0.82**
`Mobile&Calendar`	0.69	**0.84**
`Mother&Daughter`	0.43	**0.70**
`Foreman` (QCIF)	0.51	**0.81**
`Silent`	0.53	**0.68**

Table 2.10 — Efficiency of double-layer coding as defined in Eq. 2.9

2.5.1.1 Comparison to other research

Also in MPEG-2 Video, the major drawback of SNR scalability is the significant rate increase; as much as 15% have been observed with a ΔQP of one, and about 4% with the in MPEG-2 largest possible ΔQP [WG96]. The high number (52) of scalar quantizers in H.264 is a clear advantage for SSH1 in contrast to 32 quantizers used in MPEG-2, as the tables Tab. 2.4 through Tab. 2.9 show. ΔR in IPP coding can be found, with $\Delta QP = 30$,

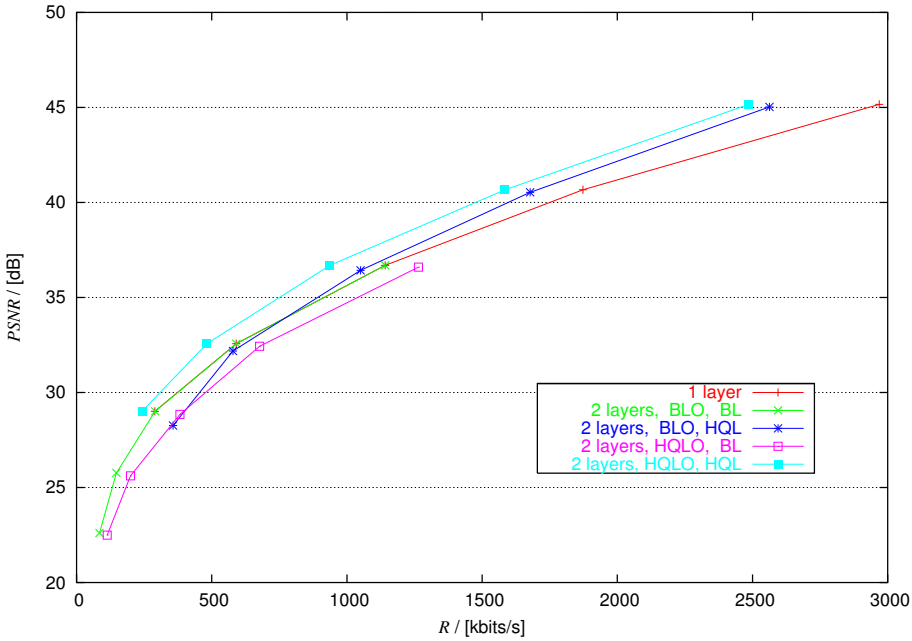


Figure 2.14 — Rate distortion comparison of various layers. Video: `Mobile&Calendar`, size: CIF, IPP coding mode, $\Delta QP = 10$

between 1 and 3%, and partly it is even below 1%. In less 'extreme' situations, SSH1 performs equally well with values of ΔQP around 10–15 and better above this values, as shown by most other tabular entries.

In [Li99], a comparison between the FGS scheme and non-scalable coding in MPEG-4 Visual is given, where a $PSNR$ difference of 3 dB is reported at the upper end of the bit rate range (for equal rates). The layered scheme proposed here performs superior to these results as shown in Fig. 2.12. The drop in $PSNR$ is at most roughly 2 dB with $\Delta QP = 5$ and converges monotonically to zero. As [Li99] further shows the superiority of the FGS scheme relative to simulcast, this means implicitly that also the new method outperforms simulcast, and hence no additional experiments are carried out.

Considering a layered H.263 Annex O video codec, an increase in bit rate of approximately 29% has to be expected with two layers to reach the quality of a single-layer coding scheme [GK99]. Unfortunately, the reference does not mention for which testing parameters and which sources this result is valid. The scheme proposed in this work, however, performs better already for a ΔQP of larger than five in some circumstances, and

$\Delta QP > 10$ in most other cases. A comparison of efficiencies confirms this observation. The efficiency of the double-layer SNR-scalable codec in [GK99] is approximately 0.73 (with QCIF-size `Foreman` at 10 fps and a rate distortion optimization of the coding mode on each layer), whereas SSH1 with the same parameters and HQLO yields an efficiency of 0.81.

The still-image compression standard JPEG operated in the progressive coding mode, i.e. either with spectral selection, successive approximation, or a mixture of both, yields roughly the same bit rate at completion of the decoding process as the JPEG standard baseline coding mode [PM92]. In contrast to that, the rate increase of INTRA coding of the proposed SNR-scalable H.264 scheme is around 10% with a realistic $\Delta QP = 20$. The superiority of the progressive JPEG standard is given by the fact that an embedded bit stream is produced.

2.5.2 Three quality layers

This section considers three layers based on the results of the previous section.

The experiments are limited to two representative image sequences for which the developed SNR scalability scheme worked well with two layers, CIF-size `Mobile&Calendar` and QCIF-size `Foreman`. ΔQP is set equal to eight, and $QP_{\text{BL}} = \{30, 34, 38, 42, 46\}$, the QP of the medium-quality layer is $QP_{\text{MQL}} = QP_{\text{BL}} + \Delta QP$, and $QP_{\text{HQL}} = QP_{\text{MQL}} + \Delta QP$. The videos are coded as sequences of GOPs of length one second, where the first frame in a GOP is an I frame. The MB type decision is based on the medium-quality layer. It is stressed that this triple-layer scheme basically extends the double-layer scheme by a further base layer.

Video name	2 layers			3 layers		
	10 fps	15 fps	30 fps	10 fps	15 fps	30 fps
`Mob.&Cal.`	0.79	0.57	0.32	0.65	0.48	0.28
`Foreman`	0.73	0.56	0.35	0.57	0.45	0.29

Table 2.11 — Efficiency of double-layer and triple-layer coding, as defined in Eq. 2.9, with different frame frequencies

As expected, the coding efficiency decreases further by adding another layer, as shown in Tab. 2.11. The efficiency difference is largest at low frame frequencies, i.e. around 10 fps, namely 0.14 with `Mobile&Calendar` and 0.16 with `Foreman`, and it is smallest at high frame frequencies, e.g.

here 30 fps, namely 0.04 and 0.05, respectively. Another description of this phenomenon is the efficiency of the triple-layer system with regard to the double-layer scheme, which is 0.82, 0.84, and 0.86 for the frame rates 10, 15, and 30 fps, respectively. This behavior can be explained — as above — by the fact that a high frame rate means a good temporal prediction, which in turn leads to very similar coefficient set in the different layers and hereby to a considerable rate increase. The strong degradation in coding efficiency over the increase in frame rate is evidence for this effect. In relation to the double-layer system's high ΔR, the further rate increase of the triple-layer system is only small, and so is its efficiency.

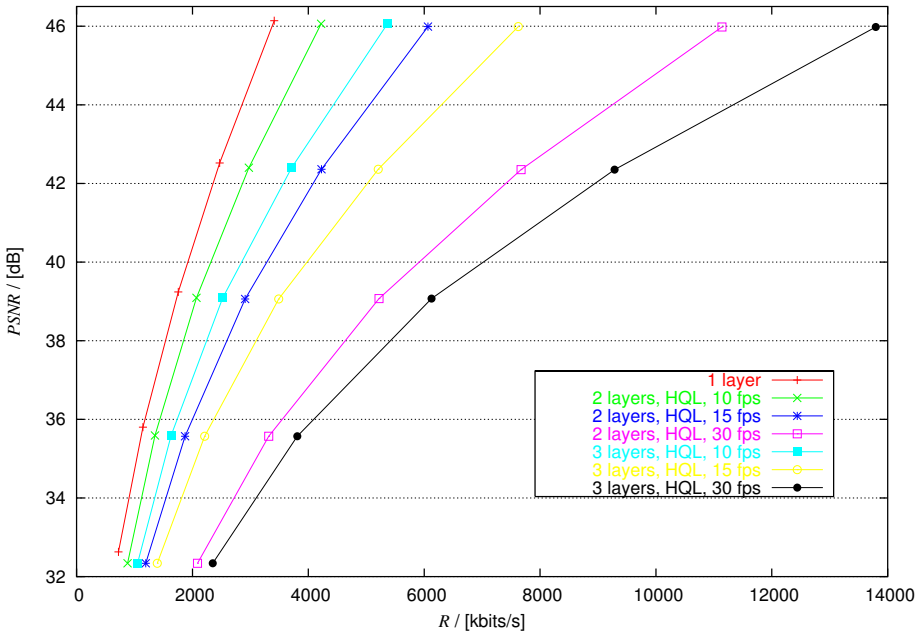


Figure 2.15 — Rate distortion comparison of various layers. Video: `Mobile&Calendar`, size: CIF

Tab. 2.11 compared to Tab. 2.10 reveals a slight drop in performance partly due to different QP values, but mainly because of the fact that MQL optimization (MQLO) is employed. It is further observed that the triple-layer scheme operates close at the efficiency of the double-layer system with BLO enabled. The rate distortion curves of both videos are given in Fig. 2.15 and Fig. 2.16.

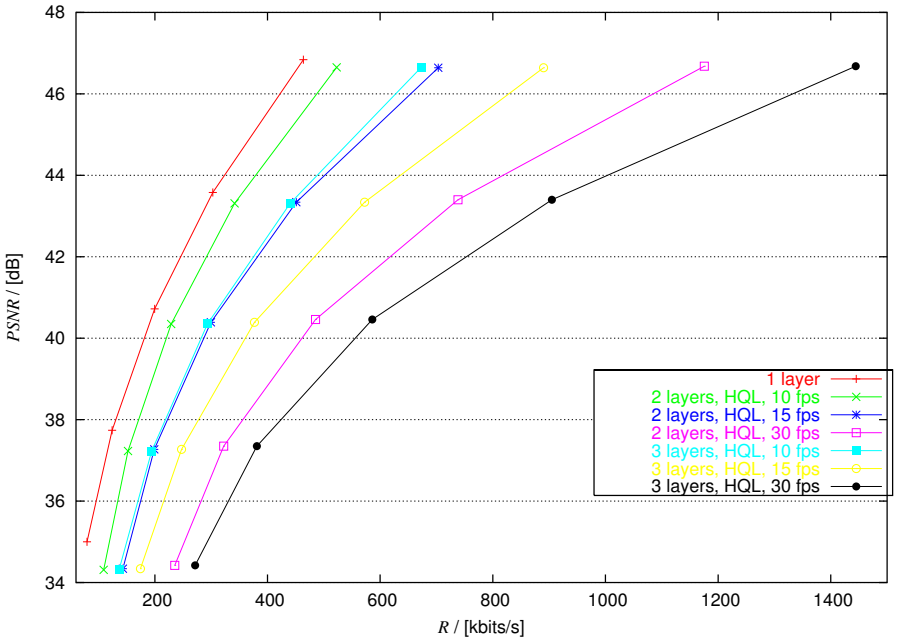


Figure 2.16 — Rate distortion comparison of various layers. Video: `Foreman`, size: QCIF

2.5.2.1 Comparison to other research

In [KIK98b], the performances of two three-layer SNR-scalable H.263-compliant codecs[7] — here called SSH3a and SSHb — are investigated. SSH3a involves a combination of spectral selection and successive approximation of quantized transform coefficients in an H.263-similar codec, whereas SSH3b is an implementation of H.263 Annex O as described in Sec. 2.3. The average distortions yielded hereby are tabularized in Tab. 2.12, in contrast to those achieved by SSH4. The coding parameters including layer QPs and frame skip of SSH4 are adjusted such that the rate constraint is approximately met. More specifically, the true layer rate consumption is roughly equal to the values given in Tab. 2.12, namely 27.41, 26.64, and 71.55 kbps with `Foreman` and 27.60, 28.51, and 73.74 kbps with `Akiyo`. SSH4 clearly outperforms both SSH3a and SSH3b, the differences in $PSNR$ being 1.5–2 dB at the medium-quality layer and 2.5–3.5 dB at the high-quality layer. Thus, it seems that the proposed SNR-scalable scheme does not increase the rate as much as other

[7] Unfortunately, the addressed profile is not mentioned.

systems when adding more layers.

Video name	Bit rate	SSH3a	SSH3b	SSH4
Foreman	28.8	28.72	30.79	30.54
	56	29.73	32.53	34.63
	128	34.63	35.56	38.24
Akiyo	28.8	36.73	38.17	38.35
	56	38.04	39.06	40.68
	128	41.91	41.47	45.32

Table 2.12 — Rate distortion comparison of SSH3a, SSH3b, and SSH4. The layer $PSNR$ is given in dB, and the unit of the bit rate is $kbps$

2.6 Summary, conclusions, and outlook

In this chapter, the rationale for the employment of layered video coding in the target applications was established. After a brief definitions of terms and examples for scalable communication, the video coding standard H.264 was introduced as the latest state-of-the-art representative of the widely known block-based hybrid video coding scheme. H.264's technical specifications including spatial and temporal predictions, and transforms were discussed in detail with regard to both encoder and decoder matters.

After that, the focus was on the standard's error resilience properties, which were discussed exhaustively. This section is furthermore meant to prepare the discussion in the next chapter, namely on error propagation issues in Sec. 3.3 and on an appropriate packet architecture in Sec. 3.5.

Various SNR scalability schemes were introduced, starting with widely spread video coding standards like MPEG-2 Video and MPEG-4 Visual, as well as H.263 (and additionally the still-image coding standard JPEG), and then continuing with schemes which had been proposed during the development of H.264.

Next, a new quality scalability method was derived, based on a detailed study of one of H.264's entropy coding engines called CAVLC, also including a discussion of the new method's impact on the coding mode evaluation of the encoder. The technique can be described as transform coefficient refinement under a fidelity constraint. As shown, too, the scheme enables ROI coding and offers hereby additional functionality compared to many conventional systems.

2.6.1 Summary of results and conclusions

To evaluate the proposal's performance, it was implemented into H.264's reference software, and a series of experiments was carried out for this quality-controlled system. It could be reported that the bit rate increase, as usually encountered with layered hybrid coding schemes when compared to single-layer codecs, is on the average much lower for the new system. Unfortunately, an average value for the overall improvement can not be given because the amount of additional resource consumption depends, as shown, strongly on the source signal and coding parameters like image sequence coding mode (INTRA/INTER), frame skip, and ΔQP.

The new codec outperforms most existing systems in a rate distortion sense when ΔQP is sufficiently large. Considering two quality layers and a set of realistic encoding parameters, the rate increase is typically only between 10% and 20%, in contrast to roughly 30% as reported in the literature. In certain situations, it may become even less than 1% and converges hereby to the performance of a single-layer scheme. The SNR scalability extension profits from the standards excellent coding gain reflected by the fact that H.264 outperforms all previous coding standards like MPEG-4 Visual and H.263++ with a substantial margin [WSJ+03].

It is further shown that the coding ability of a triple-layer system is inferior to that of a two-layer codec. However, compared to two other recently developed triple-layer schemes, it is found that the proposed codec performs better than those in a rate distortion sense.

The concept of SNR scalability by coefficient refinement is limited by memory requirements and restrictions with regard to computational complexity and latency. As explained above, all processing structures of the base layer must be doubled with the exception of motion estimation and coding mode determination. This requires more memory and more processing power. With completely parallel structures, the latency of the layered codec should be the same as that of the single-layer scheme, otherwise it will grow somewhat.

Also limitations of the proposed scheme became visible in the experiments. For certain coding parameters and image test material, the bit rate more than doubles due to an inadequateness of the entropy coding step. This issue is hence picked up in Sec. 2.6.2 once more. Finally, the bit stream produced is not embedded. If this is a disadvantage depends actually on the application, but generally it does not allow for a smooth quality in-/decrease when the channel bandwidth varies like in a ramp function.

In total, it can be concluded that the proposed scheme is an excellent candidate in error-prone environments and for systems where SNR scalability is highly desirable.

2.6.2 Outlook

As always in the scientific world, improvements to existing techniques can be made. Some ideas for future research are presented in the following.

As discussed in Sec. 2.5.1, the rate increase of the new scheme and the drop in $PSNR$ may be higher than desirable, especially with a small ΔQP or a low coding frame rate. This problem is attributable to

- the enhancement layer's quantization dead-zone which thresholds also quantized coefficients,
- many small run lengths, the coding of which consumes much rate,
- the non-linear nature of VLC code tables, as the coding of two coefficients typically produces code words which combined are longer than the code word generated by coding the same coefficient in an equivalent single-layer codec, and
- a different coefficient distribution in the enhancement layer than in a single-layer scheme, i.e. the distribution is no longer matched to the VLC tables suited for 'normal' (single-layer) coefficient sets.

A solution to the problem is to alter the CAVLC coding mechanism to account for the new coefficient distribution. Possible modifications are the deployment of different VLC tables and for instance an increase of the maximum number of codable trailing one's. The usefulness of rate-consuming variables like CBR must be re-considered, as zero changes in the coefficient sets turns out to be rarely the case. Entropy coding with even higher efficiency tailored for layered coding remains hence for future research.

A related topic is the replacement of CAVLC by context-adaptive binary arithmetic coding (CABAC) as in H.264's Main profile [MSW03]. CABAC is more complex than CAVLC but outperforms it by typically 5–15% in terms of bit rate savings. The SNR scalability scheme introduced above would work especially well with CABAC, as the arithmetic encoder adapts automatically to the source statistics of any source, performing close at the entropy bound.

The topic of SNR scalability is related to spatial scalability. Spatial scalability combined with the solution given in this work could be achieved by interpolating the low-resolution picture by appropriate filtering. The transform coefficients of the expanded base layer can be computed thereafter and are then available for prediction of the coefficients of the high-resolution picture.

Finally, two more considerations. If reduced complexity and memory usage are desired, the SNR-scalable scheme could be altered to allow buffer drift, that is leaky prediction, where the high-quality layer utilizes inter-layer temporal prediction. Also, as already mentioned, the inclusion of an anti-blocking filter in the high-quality layer is expected to provide additionally bit rate savings of at least 5%.

Chapter 3

Robust transmission of code streams

As already mentioned, block-based hybrid video codecs — here represented by H.264 — rely mainly on spatial and temporal prediction as well as VLC to obtain high compression. Potential transmission errors are hence likely to propagate through the code stream after their occurrence, as discussed in Chap. 3.2, and this necessitates some form for protection, i.e. channel coding, to make the data to convey more resilient to transmission errors by increasing the redundancy in a controlled manner.

Considering packet-wise data conveyance, the use of forward error correction (FEC) schemes is limited due to temporally very high error rates encountered on e.g. mobile networks. A worst-case system design would lead to a significant amount of overhead. Another approach is the closed-loop error control technique ARQ which has been shown to be more effective than FEC alone [LCM84]. On the Internet, TCP/IP services employ ARQ in combination with check sum codes to detect channel errors. However, such systems can introduce a considerable delay when the error rate is high, as reported in [Hal02b].

In applications like conversational services where either such delays are not acceptable (video conferencing) or where there is no feed-back channel (broadcasting), stand-alone forward channel coding techniques are required due to their efficiency. In order to reduce the amount of side information of a worst-case design, the channel encoding has to be tailored to the particular error conditions on the channel if they are known. Unknown conditions can be estimated, and, combined with the property of graceful performance degradation in mismatch situations, lead to a

robust system. How this can be done in an optimal manner in terms of MSE is investigated in the following.

3.1 Previous work

Research reports regarding robust transmission of video code streams can roughly be divided into those which deal with embedded code streams and those which consider hybrid code streams. In general, binary signal modulation is assumed in the subsequent sections.

3.1.1 Embedded bit streams

Embedded bit streams are code streams which can be truncated at any point without affecting the ability to decode the stream. In other words, even shortened code streams are valid streams, interpreted as being error-free by the decoder. Typically, there is no error concealment associated with an embedded bit stream. Decoding profits from each additional bit in the stream, hereby improving the image quality in a *progressive* manner. That is, first a rough approximation of the signal will be available, which is then more and more refined. Examples of schemes producing embedded bit streams are the still-image compression standard JPEG2000 [ISO98] and the in image compression widely known 3-D SPIHT scheme [KXP00].

Considering the problem of optimum rate allocation between source and channel encoder, a family of algorithms is presented in [CF99]. The approach requires uniform source segment[1]sizes and results in transmitted packets of varying length according to the selected channel code rates. A related solution is given in [Ban02], where the length of the transmitted packets is kept fixed, while the payload length of the packets varies. In contrast to these two works, the authors of [ZA00] propose an approach in which the expected source bit rate is maximized. This is based upon the bit error rate of each channel code and equals the maximization of the expected image quality measured in $PSNR$.

Many more research reports considering embedded code streams exist in the literature; however, as the focus of this thesis is on hybrid bit streams, the scope of this review is limited to the works mentioned.

[1]This term is explained later in Sec. 3.3.

3.1.2 Hybrid bit streams

Hybrid bit streams are code streams produced by the coding concept known as hybrid codec, which is explained in Sec. 2.2. Logical strings of bits, so-called data units, must be available to the decoder to form a valid bit stream, otherwise the stream is assumed to have been affected by channel errors. The decoding process of that particular data unit is then terminated, the pertained region is concealed, and the decoder proceeds with the next error-free unit. The hybrid coding concept is very popular, examples being the H.26x standard series by ITU-T and the MPEG-x Video/Visual series by ISO/IEC.

In [ZBPK03], the authors develop a JSCC system with optimized MB mode selection, several SNR layers, and FEC coding for IP-wise transmission subject to a rate constraint. A similar technique is applied in [GK01], with rate-constrained Internet-type transmission in mind. The coding mode determination for the spatially layered stream is done with regard to the channel conditions (i.e. packet loss rate), the error resilience (in terms of INTRA coding) inserted in the code stream by the source encoder, and the concealment capabilities of the decoder.

Related to the problem of jointly minimizing the expectation of the distortion of the decoded video signal under a rate constraint is the formulation of a transmission power and delay instead of a rate constraint as conducted in [ELP+02]. The work assumes block-based hybrid coding, but no layering. The scheme is extended to account for transmission scheduling of packets in [LEB+03], where scheduling means that, with poor channel conditions, it is advantageous for the transmitter to idle until the channel broadens up. This leads to a modified end-to-end delay constraint.

The authors of [KIK02] consider the allocation of the available transmission rate for layered representation of motion-compensated *DCT*-based SNR-scalable video. The optimum rate 1) for each layer and 2) between source and channel coding is determined by means of universal rate distortion characteristics and conditional distortions.

In [JKL98], the data of a non-layered H.263 bit stream are re-arranged into fixed-length bit strings on a coefficient, block, macroblock, and GOB level to enhance error localization and subsequent concealment. [TN02] re-groups the data bits according to a heuristically derived error sensitivity assessment and applies then UEP to each data unit. While the authors of the two last mentioned works consider transmission over a wireless network, a similar data partitioning approach in [CZZC00] (with the Simple

profile of MPEG-4 Visual) assumes IP-wise transmission. [GLM+03] aims at the FEC optimization by accounting further for the estimated network conditions and a transmission rate constraint. The authors of [HSLG99] study additionally layered coding.

In [YZLF02], the data sensitivity is based on inter-packet dependencies due to prediction, and the distribution of channel codes is optimized — subject to a transmission rate constraint — by the minimization of the expected channel error impact. For this, a novel error metric is defined.

The authors of [ZLE+03] have investigated various coding aspects when transport priority classes are used instead of best-effort networks. The system parameters were optimized jointly. Different packetization schemes for robust IP-wise transmission of hybrid video are discussed in [ZEL+03].

The wireless transmission of a mixed hybrid-embedded bit stream in terms of FGS is analyzed in [WZZZ00] and [WZZZ01], while [WZZ02] accounts in addition to random bit errors for packet erasures. The hybrid base layer is protected such that it is feasible for the systems's error concealment to achieve an acceptable image display quality after erroneous transmission, and the remaining channel rate is assigned to the enhancement layer.

A two-stream method is proposed in [RAG04] for broadcasting, where an additional stream produced by a Wyner Ziv encoder is conveyed to improve the base layer stream, which is generated by a conventional hybrid encoder, at the decoder side.

Network topics for the transmission of VBR video streams over ATM networks are discussed in [LZ97], including constrained and unconstrained VBR, source rate control, and both delivery guaranty and best-effort traffic.

3.2 Error propagation in hybrid code streams

This part deals with error robustness issues of predictive coding in video compression.

Traditional codecs based on block-based hybrid coding schemes rely upon — among other methods — mainly three coding techniques to achieve high compression factors while maintaining a good visual image quality. All mentioned schemes are to some extend prone to channel errors. They are

- run length coding (RLC),
- VL coding (VLC), and
- prediction of e.g. samples in the spatial or temporal domain, or coefficients in the frequency domain.

In hybrid video coding, these techniques are usually combined to achieve an overall good compression ratio, as explained in Sec. 2.2. All methods are examined briefly in the following with regard to their sensitivity to errors.

3.2.1 Run length coding

RLC is coding by convention, where a set of variables describes the behavior of another set of variables. For instance, the number of occurrence of certain symbols, say zero-bits, is counted and transmitted instead of those symbols. The variable 'run' presented in Sec. 2.3.2 works on this basis. Another example is the functionality of a flag which signals the presence or absence of subsequent variables. The CBP as discussed in Sec. 2.4 follows this principle. It is easy to imagine why an error in such variables is disastrous for the decoding process.

3.2.2 Variable-length coding

variable-length (VL) codes are used in lossless coding to reduce the statistical redundancy of the symbols to encode. Most VL codes are known to be very vulnerable to transmission errors. Consider for instance the three code books/tables of which a small part is listed in Tab. 3.1. A table defines the mapping of an index to its corresponding binary code word (CW). Obviously, given a certain source symbol distribution, the coding gain of all three codes is equal. With respect to error resilience, however, the codes behave quite differently.

The exponential Golomb code (EXPG), utilized in the H.264 standard for coding of side information, is likely not to regain synchronization after a single bit error. Tab. 3.2 illustrates this behavior. In the upper example, the error causes the entropy decoder to loose synchronization. The decoder re-establishes synchronization in the second example but is then faced with a code word assignment problem.

The code UVLC is a self-synchronizing code. However, to be able to assign the code words to the correct source symbols, the entropy decoder

CW index	Code name		
	EXPG	UVLC	VLCD
0	1	1	1
1	010	001	000
2	011	011	010
3	00100	00001	00100
4	00101	00011	00110
5	00110	01001	01100
6	00111	01011	01110
⋮	⋮	⋮	⋮

Table 3.1 — Mapping from CW index to CW

Symbol	Sequence
Source symbols	$(3, 0, 2, 1, 2, 1)_{10}$
Code stream	$(0, 0, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 1, 1, 0, 1, 0)_2$
Disturbed code stream	$(0, 0, \underline{0}, 0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 1, 1, 0, 1, 0)_2$
Decoded indices	$(44, 5, 0, ?)_{10}$
Disturbed code stream	$(\underline{1}, 0, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 1, 1, 0, 1, 0)_2$
Decoded indices	$(0, 1, 1, 0, 0, 1, 2, 1)_{10}$

Table 3.2 — Two example transmissions of a source symbol sequence involving the code EXPG. The underlined symbols mark the error location

has to rely on additional information about the error location [Hal02a]. The same applies also to the code named VLCD, a reversibly decodable code.

3.2.3 Prediction

Signal prediction is a lossless coding method which reduces signal correlation and first-order entropy. Prediction is a very versatile tool as it can be used to reduce the dynamic range of a variety of signals like samples, transform coefficients, motion vectors, coding modes, etc. Some applications of prediction in the standard H.264 are explained in detail in Sec. 2.2 and Sec. 2.4.

Unfortunately, prediction makes the code stream very vulnerable to transmission errors. Constructs like slices (see Sec. 2.2.1) aim at limiting

the impact of errors over the spatial plane; however, due to ME/MC the effects of errors can also cross slice boundaries. An example of this spatio-temporal error propagation is given in the next section.

3.2.4 Error propagation in H.264

All of the aforementioned techniques are applied in H.264 as explained in Sec. 2.2, making the standard prone to channel errors. Fig. 3.1 illustrates this fact.

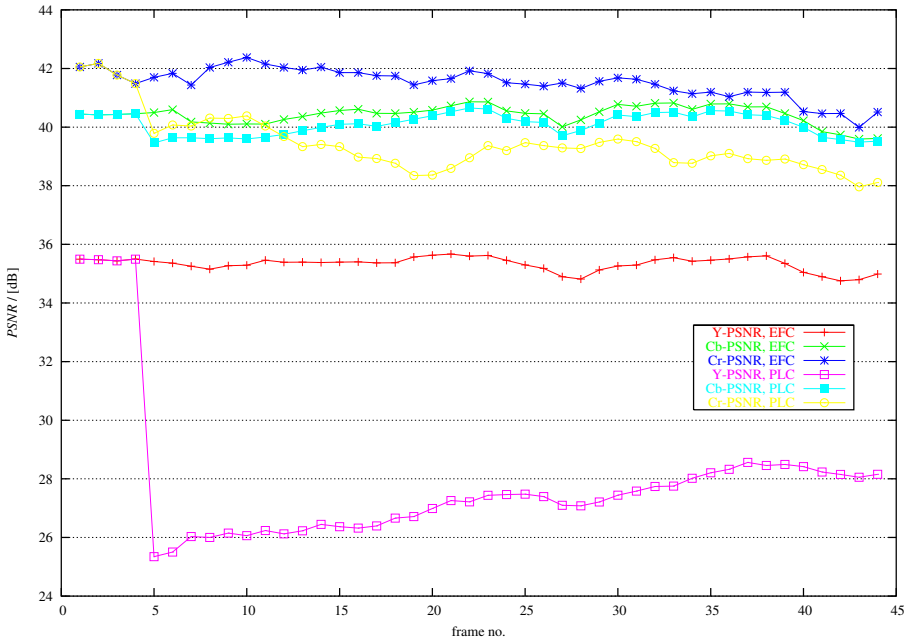


Figure 3.1 — Decoder $PSNR$ of all color components relative to the frame number, and with (PLC) and without (EFC) packet losses

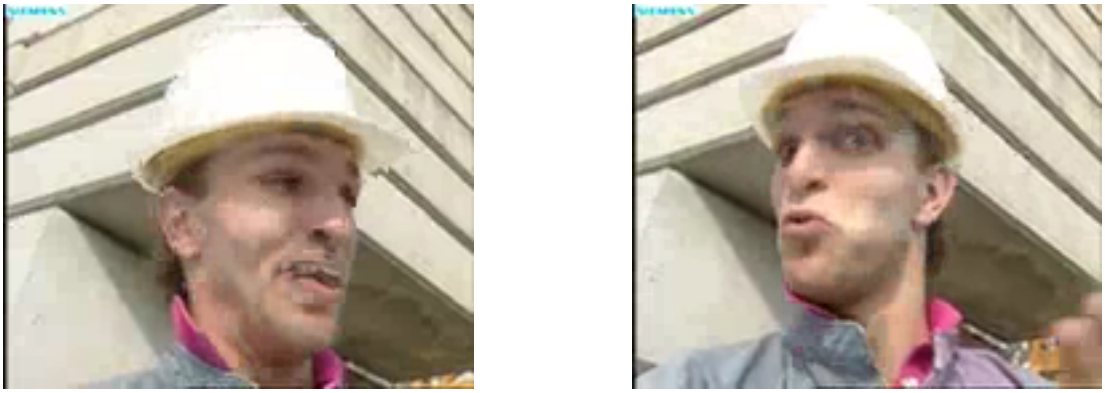
It shows the end-to-end distortion, measured in $PSNR$, of H.264's reference software JM-4.0d in the presence of packet losses. The curves represent the decoder performance for a single realization of memory-less random packet erasures. The QCIF-size `Foreman` video is coded at a QP of 30 and a skip of 1, in the error-free case (EFC) resulting in the $PSNR$ values 35.34 dB, 40.43 dB, 41.51 dB for the color components Y, Cb, and Cr, respectively. During decoding, concealment is applied to the reconstructed images before they are stored for reference purposes. Spatial concealment is done by weighted pixel averaging as proposed in

[SSD98]. For temporal concealment, a most likely motion vector of a lost MB is computed as the motion vector average which gives the lowest pixel difference along the MB edge [LRL93].

In the instance shown here, the number-four frame is completely lost; the $PSNR$ of that frame is not plotted. Clearly, the quality of all remaining pictures is affected by this single packet erasure. The $PSNR$ values of the $YCbCr$ signal are now 28.09 dB, 40.11 dB, and 39.49 dB, averaged over all 45 frames.

In [GF98], the authors derive a model for the distribution of error energy over the spatio-temporal space in the video coding standard H.263. This model is also valid with H.264, as both standards share many technical concepts. The loop filter and the interpolation of the ME/MC process lead to an error energy leakage, and the picture quality recovers over time. However, the error energy decay is not very strong due to excellent spatial and temporal prediction, as shown in Fig. 3.1, and a frequent independent update/refresh of the entire frame or at least parts of it is therefore required to speed up this process if the use of ARQ cannot be considered because of latency issues..

A visual evaluation of the error is shown in Fig. 3.2. In areas with a lot of object motion, the error impact is clearly visible. The visual impression enhances over time seemingly faster than the corresponding error metric, as the error effects have almost visually disappeared in frame 43, while the luminance $PSNR$ is still roughly 7 dB worse than in the EFC. The necessity for FEC schemes becomes clear instantaneously.



(a) Frame no. 5 (b) Frame no. 43

Figure 3.2 — Visual quality of two frames after error occurrence in the fourth frame

The error resilience of H.264 has been studied in various articles. [Wen03] gives an overview of issues regarding IP-wise transmission, it presents all error resilience tools in the recently published standard, and it conducts experiments to evaluate the standard's resilience capabilities. [SHW03] covers the code stream conveyance in wireless applications, starting with an extensive discussion of issues related to transmission in mobile and wireless environments, and touching both decoder and encoder matters, as well as topics considering feed-back channels. Also here, various experiments are executed. A comprehensive complete description of all error resistance tools in H.264 is given in [HO04], mainly focusing on rate distortion matters, the trade-off between robustness and coding gain, and optimum parameter settings of a H.264-compliant codec in erroneous environments. This includes error concealment. [CM03] considers the IEEE 802.11b standard (with feed-back) and random bit and burst errors for evaluation of the robustness of H.264[2], while [OLL+03] investigates in how far the $PSNR$ is a proper parameter to measure the image quality at the output of the decoder in mobile multimedia communication. Finally, [JHJ+03] compares the error robustness performances of MPEG-4 (Simple profile) and H.264 (Baseline) for random bit and burst errors, with IP packet networks as target applications.

3.3 Considerations on limitation of error propagation

Consider an H.264 Baseline-compliant hybrid video encoder which generates an SNR-scalable bit stream, as explained in Sec. 2.1. The stream incorporates the encoded signal and is divided into logical parts or partitions/segments. Each segment is compound of one slice of a particular frame and a particular layer and is referred to as source packet (SP). Encoding is done with a quality constraint, i.e. the specification of a QP controls both the quality of the reconstructed images and the size of the source packets[3]. Hence, the size of one source packet varies according to the size of the respective slice, signal properties like amount of motion and detailedness, and it depends also on the QP the slice is encoded with.

[2]The coding tools are chosen here such that no particular profile is addressed.

[3]In a later stage of the H.264 standardization process, a rate control technique was adopted, which allows the specification of a target bit rate and hereby leads to varying QPs within a slice. However, the poor stability of its implementation in the reference software did not allow to take this technique into account.

Next, given a set $\mathcal{C}$ of channel codes, an algorithm is required which seeks to assign each source packet with packet index i_s a channel code $\Gamma(i_\mathrm{s}) \in \mathcal{C}$ such that the overall expected distortion of the frames to transmit is as small as possible. With other words, the aim is limiting the impact of channel errors, i.e. their propagation through the code stream, to a minimum by our code selection. The number of considered frames is bound by latency requirements, complexity, and error propagation issues.

Error propagation throughout a video can efficiently be terminated by constrained intra-frame coding of arbitrary regions of a video frame, starting from a single MB over a whole slice to a complete frame. However, without feed-back channel, instantaneous decoder refresh (IDR) of regions smaller than a whole frame may not completely stop error propagation, depending on the source signal. The insertion of I frames, where all MBs of a frame are coded in intra-frame coding mode, in the sequence of frames is therefore chosen in the following to be the method of choice to recover from error propagation. The set of frames from one I frame (including) to the next (excluding) is also called GOP, since in non-interlaced coding, one frame is identical with one picture. The whole video is hereby divided into a sequence of GOPs.

3.4 Code stream segmentation

Given a GOP compound of encoded segments as illustrated in Fig. 3.3. Each segment $\tilde{X}(f, l, y, x)$ is indexed by its horizontal position $x = 0, \ldots, N_\mathrm{x} - 1$, the number of slices in one row being N_x, and $y = 0, \ldots, N_\mathrm{y} - 1$, where the number of slices in one column is N_y. The number of slices per layer is $N_\mathrm{s} = N_\mathrm{x} N_\mathrm{y}$. The affiliation of a slice to a certain quality layer is expressed by $l = 0, \ldots, N_\mathrm{l} - 1$, with the number N_l of layers, and the frame index is $f = 0, \ldots, N_\mathrm{f} - 1$, where N_f stands for the number of frames in the current GOP. In contrast to the reconstructed signal, which has been encoded and decoded, it suffices to represent the original slice by the random variable $X(f, y, x)$ with only three subscripts. $\tilde{X}$ is further described by the first moment, i.e. its mean $m_{\tilde{X}}$, and the second moment, i.e. its variance $\sigma^2_{\tilde{X}}$. There are only few large slices.

The data of all slices are written to the source bit stream in raster scan order. This order is described by the mapping $\mathcal{M} : \mathbb{N}^4 \mapsto \mathbb{N}$ of the 4-D image sequence of encoded segments to the 1-D bit stream, $(f, l, y, x) \mapsto i_\mathrm{s}$. The source packet indices are computed from the segment indices by

$$i_\mathrm{s}(f, l, x, y) = N_\mathrm{s} N_\mathrm{l} f + N_\mathrm{s} l + N_\mathrm{x} y + x\,. \tag{3.1}$$

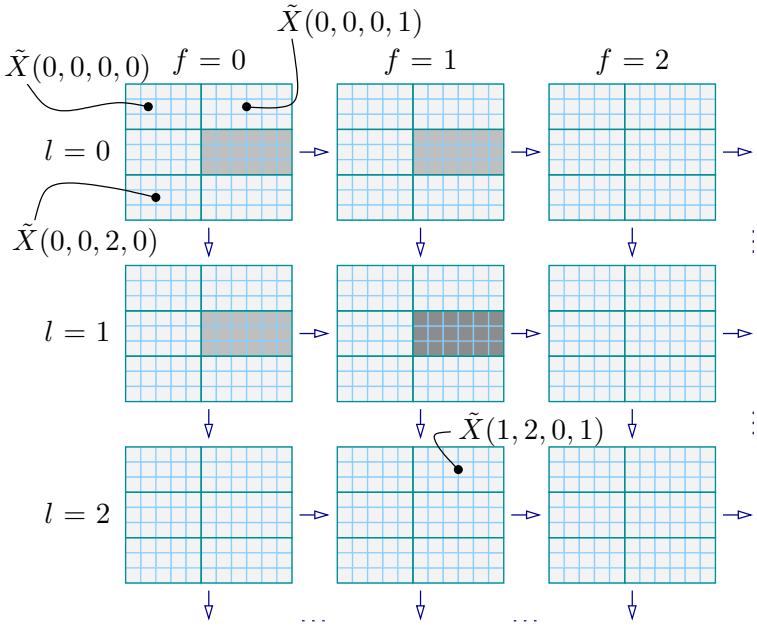


Figure 3.3 — Slices in CIF-size frames with $N_\mathrm{x} = 2$ and $N_\mathrm{y} = 3$. The lightly shaded segments constitute the set of all reference source segments to the current segment of interest, here darkly shaded

The inverse mapping $\mathcal{M}^{-1} : \mathbb{N} \mapsto \mathbb{N}^4$ can then be expressed by

$$f(i_\mathrm{s}) = \left\lfloor \frac{i_\mathrm{s}}{N_\mathrm{s} N_\mathrm{l}} \right\rfloor , \tag{3.2}$$

$$l(i_\mathrm{s}) = \left\lfloor \frac{i_\mathrm{s}}{N_\mathrm{s}} \right\rfloor - f(i_\mathrm{s}) N_\mathrm{l} , \tag{3.3}$$

$$y(i_\mathrm{s}) = \left\lfloor \frac{i_\mathrm{s}}{N_\mathrm{x}} \right\rfloor - l(i_\mathrm{s}) N_\mathrm{y} - f(i_\mathrm{s}) N_\mathrm{l} N_\mathrm{y} , \text{ and} \tag{3.4}$$

$$x(i_\mathrm{s}) = i_\mathrm{s} \bmod N_\mathrm{x} , \tag{3.5}$$

where $\lfloor \cdot \rfloor$ rounds its argument to the nearest integer towards minus infinity.

3.5 Channel packet architecture

The problem of assigning the optimum channel code to each of the packets is a discrete optimization problem and can be solved using a brute-force

search algorithm. However, considering a typical instance with $N_\mathrm{f} = 30$, $N_\mathrm{l} = 3$, $N_\mathrm{y} = 3$, and $N_\mathrm{x} = 2$, the number of source packets $N_\mathrm{SP} = N_\mathrm{f} N_\mathrm{l} N_\mathrm{y} N_\mathrm{x}$ and hereby the computational burden becomes very large — the number of combinations $||\mathcal{C}||^{N_\mathrm{SP}}$ is impossible to trace, even for a very limited number $||\mathcal{C}||$ of channel codes. In order to allow a low-complexity approach, the Viterbi algorithm, the VL source bit stream is conveyed by means of fixed-length channel packets (CPs). How the uniform length of these packets can be incorporated in the algorithm is explained in detail in Sec. 3.10.

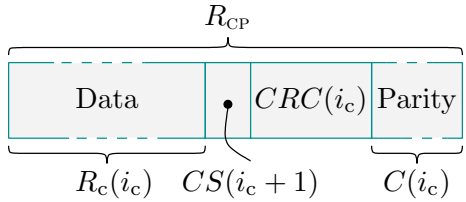


Figure 3.4 — Structure of a channel packet with index i_c

The channel packet architecture illustrated in Fig. 3.4 is adopted from [BBF02]; its basic principle is as follows. The channel encoder adds an 8-bit code specifier $CS(i_\mathrm{c}+1)$ to the payload compound of a part of the source code stream, which specifies the channel code $\Gamma(i_\mathrm{c}+1)$ of the next transmitted packet. The specifier $CS(0)$ of the channel code of the first packet should be conveyed error-free and is hence either signaled to the receiver by external means, e.g. a separate channel, or is set to be fixed by convention. A 16-bit CRC as presented in [CGG90] is computed over both encoded data and code specifier for detection of residual bit errors after channel decoding. Finally, all data are channel-encoded, adding $C(i_\mathrm{c})$ redundancy bits to each packet, and passed to the channel. As already mentioned, the length R_CP of a channel packet is constant, while a packet may correspond — in ATM — to several 53-byte cells with a payload length of up to 48 bytes.

The mapping of indices of source packets to the indices of channel packets is done with respect to the rate consumption of the channel rate allocation. Fig. 3.5 shows the definition of the source rate $R_\mathrm{s}(i_\mathrm{s})$, which is identical with the packet length of the packet with index i_s (in *bits*), and the definition of the payload rate $R_\mathrm{c}(i_\mathrm{c})$ per transported channel packet with channel packet index i_c, also measured (in *bits*). By means of the

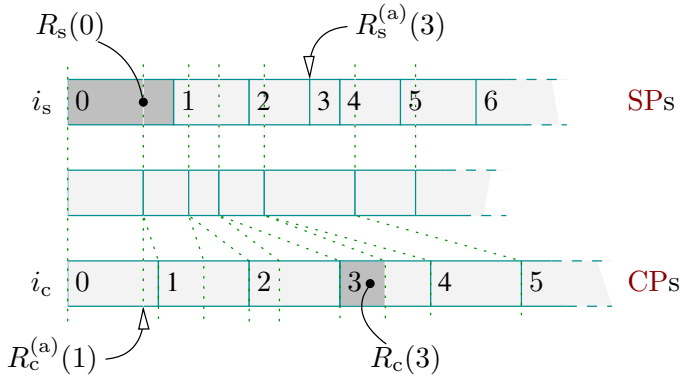


Figure 3.5 — Conversion of source packets to channel packets

accumulated source rate

$$R_s^{(a)}(i_s) = \sum_{i=0}^{i_s-1} R_s(i) \qquad i_s = 1, \ldots, N_{SP} \tag{3.6}$$

(in *bits*), the special case being $R_s^{(a)}(0) = 0$, and the accumulated channel payload rate

$$R_c^{(a)}(i_c) = \sum_{i=0}^{i_c-1} R_c(i) \qquad i_c = 1, \ldots, N_{CP} \tag{3.7}$$

(in *bits*) including $R_c^{(a)}(0) = 0$, the mapping of source packet indices to the channel packet indices can be defined as

$$i_s(i_c) = \max\Big(\big\{0, i_s \mid R_s^{(a)}(i_s+1) \le R_c^{(a)}(i_c)\big\}\Big) \qquad \forall\, i_c\,. \tag{3.8}$$

The number of channel packets which have to be transmitted for one GOP then becomes

$$N_{CP} = \big\{i_c \mid R_s^{(a)}(N_{SP}) \le R_c^{(a)}(i_c)\big\} + 1 \qquad \forall\, i_c\,. \tag{3.9}$$

It is stressed that, for one GOP, N_{CP} varies with different code allocations. Zero padding may have to be applied where appropriate to fill up a channel packet.

3.6 Distortion definitions

As already mentioned, slices are independently decodable, i.e. there are no inter-slice dependencies within one frame. Channel errors can, however, propagate over several frames and hereby also cross slice boundaries.

An analytic description of possible error propagation requires hence constraints on the encoding process. This work considers thus a single reference frame and contiguous rectangular slices, and requires that the outer slice boundaries be treated as frame boundaries, i.e. reference samples outside the corresponding reference slice are extrapolated for MC/ME. Moreover, the anti-blocking filter functionality along slice edges must be switched off. These requirements are not very restrictive, as most applications are bound to a single reference frame due to complexity issues anyway. Furthermore, rectangular slices allow more efficient coding than e.g. horizontal and vertical slices.

Based on the available channel codes, the channel code allocation scheme is designed such that it allows for unequal error protection (UEP), which means that important data are assigned strong channel codes, whereas less important data are less protected. UEP in best-effort networks means transport prioritization. Every channel packet has hence to be associated with a measure of the payload's importance, which is here chosen to be a distortion measure. There are two kinds of distortions that have to be accounted for, one made by quantization and one introduced by channel errors *and* error concealment. The formulation of the latter requires a model for potential error propagation, which corresponds to a description of source packet dependencies, and the formulation of proper decoding and concealment strategies.

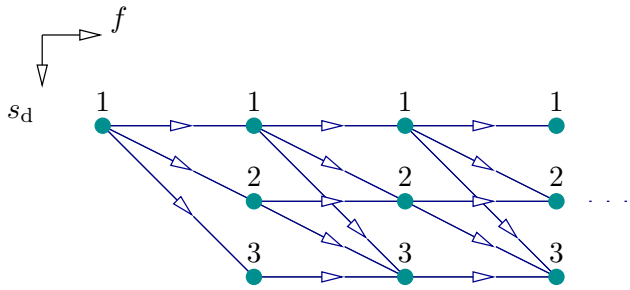


Figure 3.6 — Decoder trellis of one segment for two layers. A node corresponds to a received frame. In state $s_d = 3$, all layers have been lost. In state $s_d = 2$, the high-quality layer has been lost. Both layers have been received error-free in state $s_d = 1$

Transmission errors are often the cause to a serious quality degradation of the decoded images. It is therefore viewed as being advantageous to stop decoding when an error has been detected (by the mentioned *CRC*) and carry out a controlled concealment. This is also referred to

as *terminate on error* and here valid for only one spatial segment due to the above defined slice bound constraint, i.e. the successful decoding of slices at other spatial location is not affected by an error. Decoding of a segment is continued after the next synchronization point, here the beginning of an I frame. Concealment can thus always, in terms of channel error distortion, rely on undistorted data for reference purposes. The decoding trellis is depicted in Fig. 3.6. The generic structure of the decoder is shown in Fig. 3.7.

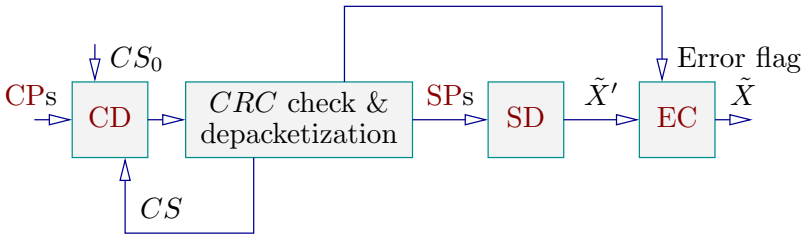


Figure 3.7 — Decoder flow diagram. CD: channel decoding; SD: source decoding; EC: error concealment

3.6.1 Concealment distortion

A suitable measure for the importance of a channel packet is the GOP's average distortion to which the loss of this particular packet would lead. Its formulation involves three simple concealment methods.

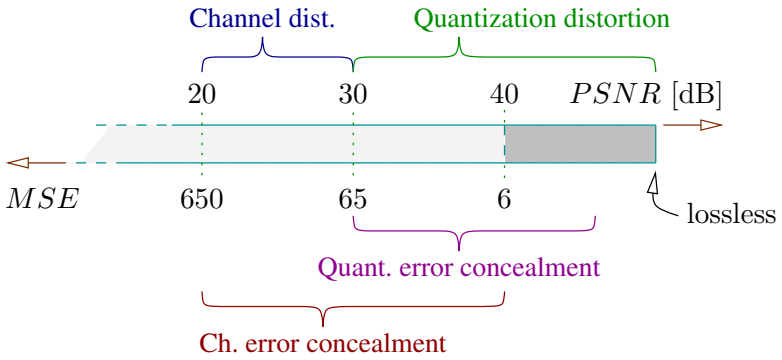


Figure 3.8 — Definitions of error concealment. The MSE scale is logarithmic

If no data are available concealment can refer to, the best estimation of

the lost data is to assume the segment's mean, accumulate the distortion contributions over the whole GOP and normalize by the number of frames,

$$\bar{D}_{\mathrm{c,m}}(i) = \frac{1}{N_\mathrm{f}} \sum_{g=0}^{N_\mathrm{f}-1} d\big(\tilde{X}(g, N_\mathrm{l}-1, y, x), m_{\tilde{X}}\big)\,. \tag{3.10}$$

If not mentioned otherwise, i is the index of the particular SP, and g, f, l, y, and x are functions of i according to Eq. 3.2. The distance metric $d(\cdot,\cdot)$ is here equal to the two-dimensional (2-D) MSE between two variables, as defined Sec. C.1.2. In practice, the mean $m_{\tilde{X}}$ is of course not known to the decoder, and the estimate 128 for the samples of the respective slice will therefore be used (this assumes an 8-bit pixel representation).

If a slice of the base layer is lost, the concealment scheme of choice is to freeze the highest-quality content of the corresponding slice from the previous frame, also referred to as slice copy or CPYSL. For one particular slice, this leads to the average GOP distortion

$$\bar{D}_{\mathrm{c,t}}(i) = \frac{1}{N_\mathrm{f}} \sum_{g=f}^{N_\mathrm{f}-1} d\big(\tilde{X}(g, N_\mathrm{l}-1, y, x), \tilde{X}(f-1, N_\mathrm{l}-1, y, x)\big)\,. \tag{3.11}$$

The last concealment technique exploits the SNR scalability property of the codec. If a lower-quality version of the lost slice is available, all remaining frames which depend on the lost one are replaced. The average slice distortion made hereby is

$$\bar{D}_{\mathrm{c,l}}(i) = \frac{1}{N_\mathrm{f}} \sum_{g=f}^{N_\mathrm{f}-1} d\big(\tilde{X}(g, l, y, x), \tilde{X}(g, l-1, y, x)\big)\,. \tag{3.12}$$

Finally, for the special case of $f = 0$ in Eq. 3.12, $\bar{D}_{\mathrm{c,f}}(i) = \bar{D}_{\mathrm{c,l}}(i)$ is defined. It is important that the slice structuring be consistent through all quality layers and frames to achieve concealment. Summarizing, the average concealment distortion, which also includes the channel distortion, is given by

$$\bar{D}_\mathrm{c}(i) = \begin{cases} \bar{D}_{\mathrm{c,m}}(i) & f = 0 \ \wedge \ l = 0 & \text{(3.13a)} \\ \bar{D}_{\mathrm{c,f}}(i) & f = 0 \ \wedge \ l > 0 & \text{(3.13b)} \\ \bar{D}_{\mathrm{c,t}}(i) & f > 0 \ \wedge \ l = 0 & \text{(3.13c)} \\ \bar{D}_{\mathrm{c,l}}(i) & f > 0 \ \wedge \ l > 0 \quad . & \text{(3.13d)} \end{cases}$$

It is stressed that the distortion reference (the first argument to $d(\cdot,\cdot)$) is — in contrast to the concealment of quantization errors as e.g. discussed

in Sec. 3.6.2 — maximally the highest-quality layer of the encoded slice since, per definition, channel error concealment cannot achieve a better quality than the highest-quality layer, see also Fig. 3.8.

3.6.2 Quantization distortion

According to the simplified codec structure including transmission, given in Fig. 3.9, it is obvious that, in the presence of errors, the total distortion of the transmitted and decoded GOP depends on the quantization error made during encoding, the distortion introduced by channel errors, and the distortion of the error concealment technique employed by the decoder. In fact, it was shown in [TC67] that this assumption is valid when the encoder and decoder reconstruction levels are identical; the conditions is always satisfied in this work.

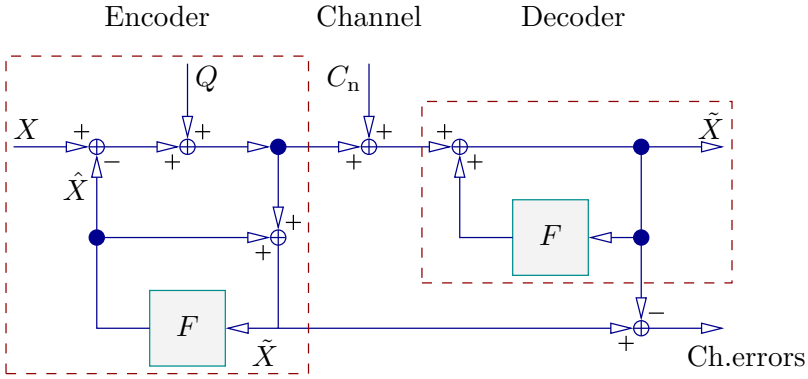


Figure 3.9 — Simplified structure of codec and channel. In the error-free case, the decoder output and the input to the encoder's frame buffer are equal. X is the input, $\hat{X}$ the predicted, and $\tilde{X}$ the reconstructed signal, whereas Q stands for the quantization and C_n for the channel noise. F is the frame buffer

Corresponding to Eq. 3.13, the definition of $\bar{D}_q(i)$ in the event 'error detected' (which includes the invoking of concealment) distinguishes among four cases, where

$$\bar{D}_{q,m}(i) = \frac{1}{N_f} \sum_{g=0}^{N_f-1} d\big(X(g,y,x), \tilde{X}(g, N_l - 1, y, x)\big) \tag{3.14}$$

and

$$\bar{D}_{q,t}(i) = \bar{D}_{q,f}(i) = \bar{D}_{q,m}(i)\,. \tag{3.15}$$

In case of layer concealment and $f > 0$, the average segment quantization distortion becomes

$$\bar{D}_{\mathrm{q,l}}(i) = \frac{1}{N_\mathrm{f}} \sum_{g=0}^{f-1} d\big(X(g,y,x), \tilde{X}(g, N_\mathrm{l}-1, y, x)\big) + \frac{1}{N_\mathrm{f}} \sum_{g=f}^{N_\mathrm{f}-1} d\big(X(g,y,x), \tilde{X}(g,l,y,x)\big)\,. \quad (3.16)$$

Summarizing, the average quantization distortion is given by

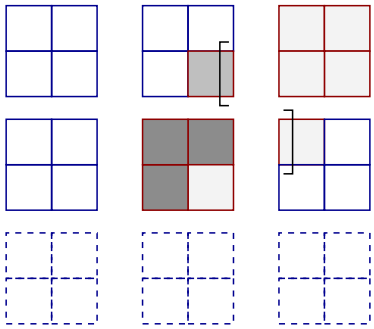
$$\bar{D}_\mathrm{q}(i) = \begin{cases} \bar{D}_{\mathrm{q,m}}(i) & f = 0 \ \wedge \ l = 0 \quad (3.17a) \\ \bar{D}_{\mathrm{q,f}}(i) & f = 0 \ \wedge \ l > 0 \quad (3.17b) \\ \bar{D}_{\mathrm{q,t}}(i) & f > 0 \ \wedge \ l = 0 \quad (3.17c) \\ \bar{D}_{\mathrm{q,l}}(i) & f > 0 \ \wedge \ l > 0 \quad . \quad (3.17d) \end{cases}$$

3.6.3 Joint distortions

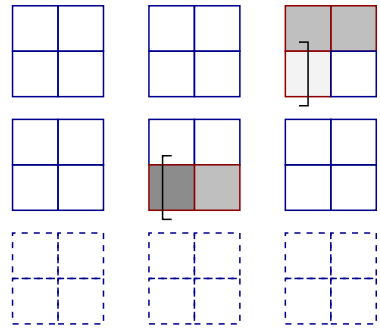
All channel packets which transport a part of a particular source packet must be channel-decoded error-free in order for the source packet to be source-decoded error-free. Hence, the same source packet distortion may be associated with different channel packets. On the other hand, a single channel packet may transport several source packets, the distortion of which influences the expected distortion if and only if it is not included in the distortion formulae of other transported slices of the same spatial location, calling for the formulation of a joint distortion. The on-error distortion is then affected by the first N_s and potentially all base layer slices of the following video frame.

The necessity to differentiate between normal- and joint-error distortion is illustrated in Fig. 3.10. A channel packet conveys all slices between the left and the right bracket in raster scan order, which are drawn with gray shading. The medium-dark shading specifies slices which the normal distortion is accounted for, whereas the joint distortion is regarded with darkly shaded slices. Lightly shaded are slices of which the distortion is ignored for calculation of the channel packet distortion. Only the first N_s slices are accounted for due to the definition of slice distortions.

Let $i_{\mathrm{s,f}} = i_\mathrm{s}(i_\mathrm{c})$ be the index of the first included source packet in the channel packet with index i_c, $i_{\mathrm{s,l}} = i_\mathrm{s}(i_\mathrm{c}+1)$ be the index of the last included source packet in the channel packet, $i_{\mathrm{s,s}}(i) = i_\mathrm{s}\big(f(i_{\mathrm{s,f}}) +$



(a) A channel packet covering a complete high-quality layer and a complete base layer of the next frame



(b) A channel packet covering parts of a high-quality layer and parts of the base layer of the next frame

Figure 3.10 — Different characterizations of channel packet distortion by slice distortions. In both examples, one frame layer consists of four adjacent quadratic slices. Layers correspond to frame layer rows, and frames correspond to frame layer columns. The original slices are represented by dashed lines. For other explanations, see the text

.

$1, 0, y(i), x(i)\big)$ be the corresponding base layer slice at the same spatial location in the next frame, and let $i_{\mathrm{s,BL}} = i_{\mathrm{s}}\big(f\big(i_{\mathrm{s}}(i_{\mathrm{c}})\big) + 1, 0, 0, 0\big)$ be the first slice in the next frame's base layer. Furthermore, two sets are defined, the source packet index set which specifies the at most N_{s} first segments in the next frame (depending on the index of the last source packet sent),

$$\mathcal{I}_{\mathrm{e,BL},1}(i_{\mathrm{c}}) = \big\{i_{\mathrm{s,BL}}, \ldots, \min\big(\{i_{\mathrm{s,l}}, i_{\mathrm{s,BL}} + N_{\mathrm{s}} - 1\}\big)\big\}\,, \tag{3.18}$$

and the source index set which specifies all next-frame segments up to the spatial location of the first sent source packet (also here depending on the index of the last source packet sent),

$$\mathcal{I}_{\mathrm{e,BL},2}(i_{\mathrm{c}}) = \big\{i_{\mathrm{s,BL}}, \ldots, \min\big(\{i_{\mathrm{s,l}}, i_{\mathrm{s,BL}} + N_{\mathrm{x}} y(i_{\mathrm{s,f}}) + x(i_{\mathrm{s,f}})\}\big)\big\}\,. \tag{3.19}$$

The set of source packets indices of these base layer slices can then be written as

$$\mathcal{I}_{\mathrm{e,BL}}(i_{\mathrm{c}}) = \begin{cases} \varnothing & \text{(i)} \quad \text{(3.20a)} \\ \mathcal{I}_{\mathrm{e,BL},1}(i_{\mathrm{c}}) & \text{(ii)} \quad \text{(3.20b)} \\ \mathcal{I}_{\mathrm{e,BL},2}(i_{\mathrm{c}}) & \text{otherwise}\ , \quad \text{(3.20c)} \end{cases}$$

where the two conditions are respectively defined as

$$\text{(i)}: \quad i_{\mathrm{s,l}} < i_{\mathrm{s,BL}} \vee \big(l(i_{\mathrm{s,f}}) = 0 \wedge y(i_{\mathrm{s,f}}) = 0 \wedge x(i_{\mathrm{s,f}}) = 0\big) \tag{3.21}$$

$$\text{(ii)}: \quad i_{\mathrm{s,l}} \geq i_{\mathrm{s,BL}} \wedge l(i_{\mathrm{s,f}}) > 0 \wedge y(i_{\mathrm{s,f}}) = 0 \wedge x(i_{\mathrm{s,f}}) = 0\,. \tag{3.22}$$

The condition $i_{\mathrm{s,l}} \geq i_{\mathrm{s,BL}}$ means $f(i_{\mathrm{s,f}}) \neq f(i_{\mathrm{s,l}})$, or that the channel packet contains several frames. Now, the set of all indices of interest can be formulated as

$$\mathcal{I}_{\mathrm{e}}(i_{\mathrm{c}}) = \big\{i_{\mathrm{s,f}}, \ldots, i_{\mathrm{s,f}} + N_{\mathrm{s}} - 1\big\} \;\cup\; \mathcal{I}_{\mathrm{e,BL}}(i_{\mathrm{c}})\,. \tag{3.23}$$

Next, in case a source packet with index i is within the first N_{s} slices of a channel packet with index i_{c}, the non-joint distortion as defined above is accounted for. If, however, a base layer slice at the same spatial location follows in the channel packet, the highest possible layer is not available for CPYSL but a lower-quality layer. Hence, some average joint distortions have to be defined: $\bar{D}_{\mathrm{jq,l}} = \bar{D}_{\mathrm{q,l}}$, as well as

$$\bar{D}_{\mathrm{jc}}(i) = \frac{1}{N_{\mathrm{f}}} \sum_{g=f}^{N_{\mathrm{f}}-1} d\big(\tilde{X}(g,l,y,x), \tilde{X}(f,l-1,y,x)\big) \tag{3.24}$$

and

$$\bar{D}_{\mathrm{jq,f}}(i) = \frac{1}{N_{\mathrm{f}}} \sum_{g=0}^{N_{\mathrm{f}}-1} d\big(X(g,y,x), \tilde{X}(g,l,y,x)\big)\,. \tag{3.25}$$

3.6.4 Error distortion

Now, the generalized average on-error distortion can be written as

$$\bar{D}_{\mathrm{e}}(i, i_{\mathrm{c}}) = \begin{cases} w_{\mathrm{ss}}(i)\big(\bar{D}_{\mathrm{c}}(i) + \bar{D}_{\mathrm{q}}(i)\big) & \text{(iii)} \quad \text{(3.26a)} \\ w_{\mathrm{ss}}(i)\big(\bar{D}_{\mathrm{jc}}(i) + \bar{D}_{\mathrm{jq,f}}(i)\big) & \text{(iv)} \quad \text{(3.26b)} \\ w_{\mathrm{ss}}(i)\big(\bar{D}_{\mathrm{jc}}(i) + \bar{D}_{\mathrm{jq,l}}(i)\big) & \text{(v)} \quad \text{(3.26c)} \\ 0 & \text{otherwise} \quad , \quad \text{(3.26d)} \end{cases}$$

where the used conditions are respectively defined as

$$\text{(iii)}: \quad i \in \mathcal{I}_{\mathrm{e}}(i_{\mathrm{c}}) \wedge i_{\mathrm{s,s}}(i) \notin \mathcal{I}_{\mathrm{e}}(i_{\mathrm{c}}) \tag{3.27}$$

$$\text{(iv)}: \quad i \in \mathcal{I}_{\mathrm{e}}(i_{\mathrm{c}}) \wedge i_{\mathrm{s,s}}(i) \in \mathcal{I}_{\mathrm{e}}(i_{\mathrm{c}}) \wedge f(i) = 0 \tag{3.28}$$

$$\text{(v)}: \quad i \in \mathcal{I}_{\mathrm{e}}(i_{\mathrm{c}}) \wedge i_{\mathrm{s,s}}(i) \in \mathcal{I}_{\mathrm{e}}(i_{\mathrm{c}}) \wedge f(i) > 0\,. \tag{3.29}$$

The weighting factor

$$w_{\mathrm{ss}}(i) = \frac{N_{\mathrm{MB}}(f,y,x)}{N_{\mathrm{MB}}(f)} \tag{3.30}$$

accounts for potentially unequal slice sizes, where $N_{\mathrm{MB}}(f,y,x)$ is the number of MBs of a particular frame slice and $N_{\mathrm{MB}}(f)$ is the number of MBs in a frame. It is derived as follows.

The MSE is not additive in the spatial domain. In order to calculate the overall distortion of a single frame out from partial distortions, a weighting factor has to be included in all distortion measures like Eq. 3.14 to reflect the influence of segment distortions and potentially different slice sizes. Consider the 1-D signals $W = (W_0, W_1, \ldots, W_4)$, $Z = (Z_0, Z_1, \ldots, Z_4)$ of length $N = 5$, and the difference signal $X = W - Z$. For a split of X into two segments ξ_i of potential unequal lengths L_i, the MSE becomes

$$\begin{aligned} MSE(X) &= \frac{1}{N}\sum_i X_i^2 \\ &= \frac{1}{N}(X_0^2 + X_1^2 + X_2^2) + \frac{1}{N}(X_3^2 + X_4^2) \\ &= \frac{L_1}{N}\frac{1}{L_1}\sum_{i=0}^{L_1-1} X_i^2 + \frac{L_2}{N}\frac{1}{L_2}\sum_{i=L_1}^{L_1+L_2-1} X_i^2 \\ &= \frac{L_1}{N}MSE(\xi_1) + \frac{L_2}{N}MSE(\xi_2) \end{aligned} \tag{3.31}$$

.

Here, $L_1 = 3$ and $L_2 = 2$ are chosen. More generally, the following can be formulated:

$$MSE(X) = \sum_i \frac{L_i}{N} MSE(\xi_i)\,, \tag{3.32}$$

i.e. the overall MSE is a weighted sum of local MSEs according to the size of the area they cover with regard to the size of the total image. The aforementioned example can easily be extended to 2-D matrices and an arbitrary (but finite) number of segments of different slices. Combined with the aforementioned constraint on temporal prediction, the distortion of a GOP can be calculated as the sum of the distortions of sub-GOPs, i.e. single slices over all frames, including all quality layers.

$\bar{D}_{\mathrm{e}}(i, i_{\mathrm{c}})$ is the end-to-end distortion of one channel packet. As such, it takes into account the segmentation of one frame, the number of quality

layers, and the number of frames per GOP. Further influencing factors are quantization distortion (and hereby the source encoding strategy) and the three error concealment methods as explained above. How in addition also the channel conditions can be accounted for is discussed in Sec. 3.8.

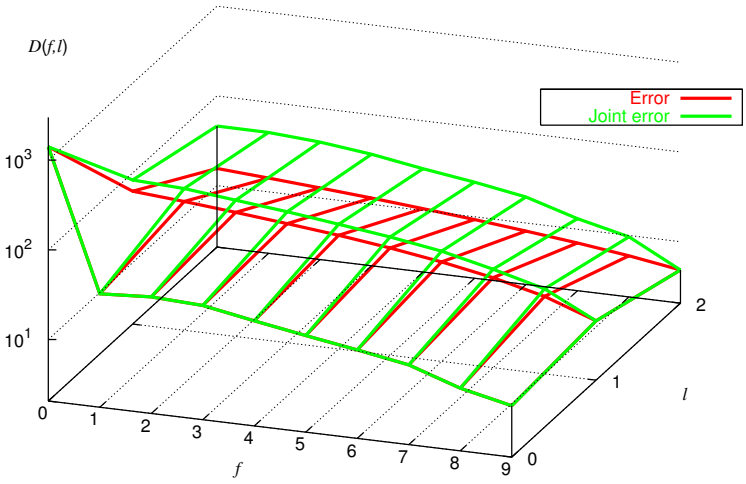


Figure 3.11 — Average error distortion $\bar{D}_{\mathrm{e}}(f,l)$ of one particular slice of CIF-size `Mother&Daughter` with $N_{\mathrm{f}} = 10$ and $N_{\mathrm{l}} = 3$

Fig. 3.11 shows the plot of Eq. 3.26 with $i = i_{\mathrm{c}}$, computed for a single slice of a GOP with $N_{\mathrm{f}} = 10$, $N_{\mathrm{l}} = 3$, $N_{\mathrm{x}} = 3$ and $N_{\mathrm{y}} = 3$ of the CIF-size `Mother&Daughter` video. The quality layers are quantized with $QP_{\{0,1,2\}} = \{40, 35, 30\}$.

It is observed that the distortion decreases monotonically with the frame number for all layers due to the accumulation process over all frames. As expected, high-quality layers have lower distortion values than low-quality layers with the exception of the base layer. Here, temporal concealment leads to an average distortion which is below that made with layer concealment. The result, however, is consistent with the observation made in [KG98], where it is postulated that data which are well recovered by the error concealment process (low-frequency information) are not as important as high-frequency information which is difficult to estimate at the decoder.

The distortion has a peak at $f = 0$ and $l = 0$ because no temporal concealment is applicable here. Instead, the variance of the source is used to measure the average distortion according to the relationship

$$\sigma^2_{\tilde{X}}\big(\tilde{X}(g, N_\mathrm{l} - 1, y, x)\big) = d\big(\tilde{X}(g, N_\mathrm{l} - 1, y, x), \bar{m}_{\tilde{X}}\big)\,. \tag{3.33}$$

This formula assumes the biased definition of the variance, i.e. the sum is normalized by the number of samples.

Yet, the values of joint distortions are always somewhat higher than those of normal distortions except for the last frame of a GOP. This is because, then, the reference for temporal concealment is worse than otherwise; the highest-quality layer of the current frame cannot be used to estimate the data in the next frame, and a layer of lower quality is taken instead. It is stressed that the joint error is actually not defined for $l = 0$ and $f = N_\mathrm{f} - 1$. However, Fig. 3.11 shows those values due to illustration purposes.

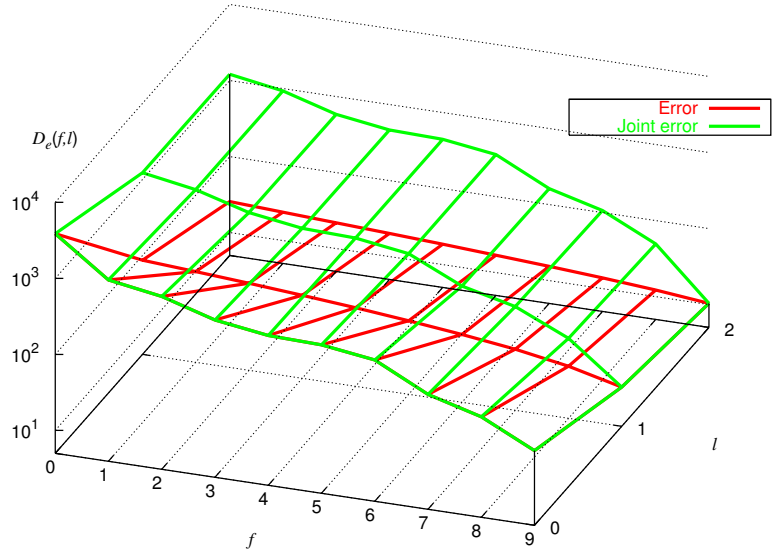


Figure 3.12 — Average error distortion $\bar{D}_\mathrm{e}(f, l)$ with QCIF-size `Foreman`, $N_\mathrm{f} = 10$ and $N_\mathrm{l} = 3$, and $N_\mathrm{s} = 1$

A somewhat different plot is depicted in Fig. 3.12. The joint error is here much higher than the normal error, attributable to the video's high

degree of motion. This results also, considering normal distortion, in a temporal concealment inferior to layer concealment. A logical conclusion would thus be to expect from the channel code allocation algorithm to avoid use of the joint distortion by trying to match the source packet boundaries, even though this is not too easy to accomplish.

3.7 Error probabilities

The channel packets are transmitted over a stationary channel which can be approximated with high accuracy by the model of a binary symmetric channel (BSC), which is schematically drawn in Fig. 3.13. It is assumed that source signal X and channel signal C_n are uncorrelated.

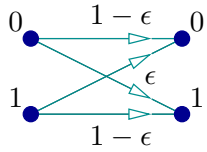


Figure 3.13 — Generic BSC model

The BSC is sufficiently described by the cross-over probability ϵ, the probability for inversion of a 0 into a 1 and vice versa. When a symbol is identical with one bit, the channel capacity is given by

$$\begin{aligned} C_\mathrm{BSC} &= 1 - \epsilon \log \epsilon - (1-\epsilon)\log(1-\epsilon) \\ &= 1 - H(\epsilon) \ \frac{\text{source symbols}}{\text{channel symbols}} , \end{aligned} \tag{3.34}$$

where $H(\epsilon)$ is the entropy of the channel noise, a zero-order, i.e. memory-free, Markovian random variable. The curve of C_BSC is drawn in Fig. 3.14.

According to Sec. 3.5, channel errors can be corrected up to a certain degree by the channel code $\Gamma(i_\mathrm{c})$, and they can be detected. Undetected errors become residual errors in the bit stream, which are detected by the CRC of the respective channel packet. If at least a single bit error in one channel packet is detected as such, the best strategy is, as explained above, to stop decoding and continue from the next available IDR slice on. $P_\mathrm{e}\big(\Gamma(i_\mathrm{c}), \epsilon, R_\mathrm{CP}\big)$ is defined as the probability of at least one bit error in the i_c-th channel packet. In other words, for a fixed R_CP, $P_\mathrm{e}(\epsilon)$ converts random bit errors into packet erasures with $\Gamma(i_\mathrm{c})$ as a parameter.

The expected distortion of a GOP after the transmission of N_t channel packets is the distortion that the particular channel packet causes when

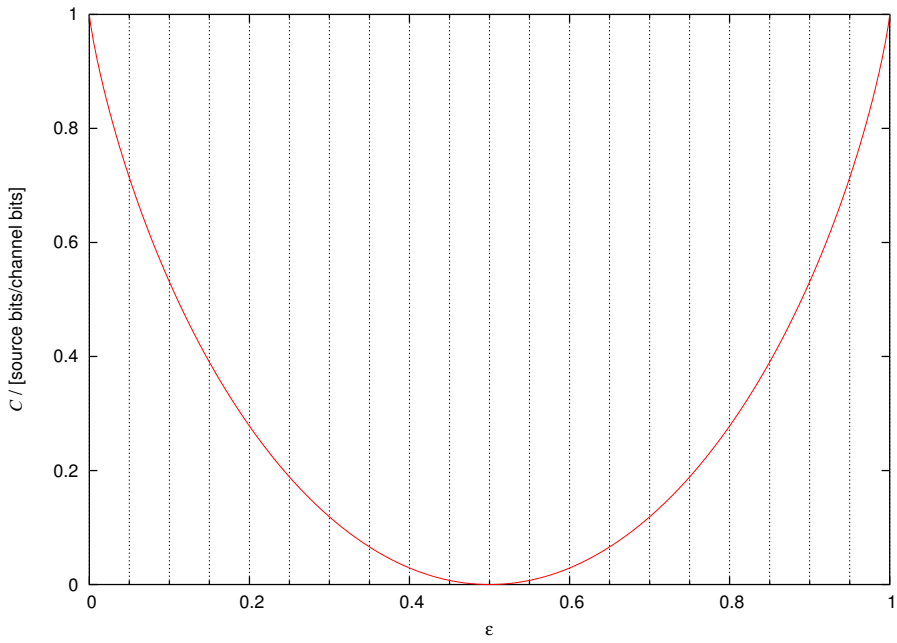


Figure 3.14 — Capacity of a BSC as a function of its cross-over probability

discarded with regard to the GOP, weighted by the probability that an error occurs in that specific packet. However, the probability of the event 'at least one $RBER$ in the channel packet with index i_c but not before'is somewhat involved. First, *reference source packets* with respect to a certain source packet are defined as those packets which the current source packet relies on in terms of coefficient or temporal prediction. In Fig. 3.6, all states which can be reached by tracing back the edges from an arbitrary state represent reference packets to that packet. Next, *reference channel packets* with respect to a certain channel packet are defined as those packets which contain at least one reference source packet of at least one in the channel packet of interest included source packet. It is noted that a source packet can be reference packet of itself due to intra-slice prediction and VLC. In Fig. 3.3, the lightly shaded source packets are temporal and SNR reference packets to the darkly shaded current source packet.

Consider the slice of interest as the segment with the indices f, l, y, and x. Given also a variable s which is taken from the set $\mathcal{S}(i_c)$ of indices of source packets which are transported by the channel packet with index

i_c, $s \in \mathcal{S}(i_\mathrm{c})$, where

$$\mathcal{S}(i_\mathrm{c}) = \left\{ i_{\mathrm{s,f}}, \ldots, i_{\mathrm{s,l}} \right\}. \tag{3.35}$$

Given further two helping variables which, starting at zero, cover the indices over all frames and layers, respectively, up to the indices of the current segment,

$$g(s) \in \left\{ 0, \ldots, f(s) \right\} \tag{3.36}$$

$$m(s) \in \left\{ 0, \ldots, l(s) \right\}. \tag{3.37}$$

Next, let t be source packet indices which are elements of the set of reference source packets of a particular source packet, $t \in \mathcal{J}(i_\mathrm{c})$, where

$$\mathcal{J}(i_\mathrm{c}) = \left\{ i_\mathrm{s}\big(g(s), m(s), y(s), x(s)\big) \right\}. \tag{3.38}$$

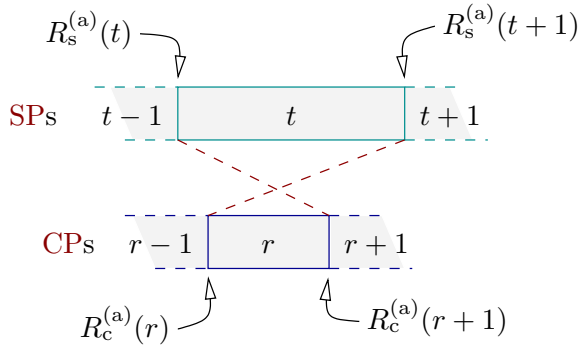


Figure 3.15 — Intersection of a source packet with index t and a channel packet with index r. Check sum code, code specifier, and channel code as depicted in Fig. 3.5 are ignored here

The set $\mathcal{R}(i_\mathrm{c})$ contains the indices of reference channel packets of the current channel packet indexed by i_c. The set is given by

$$\mathcal{R}(i_\mathrm{c}) = \Big\{ r | r \in \{0, \ldots, i_\mathrm{c} - 1\} \ \wedge \\ \left(R_\mathrm{c}^{(\mathrm{a})}(r) < R_\mathrm{s}^{(\mathrm{a})}(t+1) \ \wedge \ R_\mathrm{c}^{(\mathrm{a})}(r+1) > R_\mathrm{s}^{(\mathrm{a})}(t) \right) \Big\}, \tag{3.39}$$

of which the second term, $(\cdot)$, is the definition of the intersection of a source and a channel packet, the former one specified by its index t and the latter one by its index r. The intersection is depicted in Fig. 3.15. The left boundary of the channel packet with index r is given by $R_\mathrm{c}^{(\mathrm{a})}(r)$, and its right boundary by $R_\mathrm{c}^{(\mathrm{a})}(r+1)$. The same applies to a source packet

indexed by t, which is bounded by $R_{\mathrm{s}}^{(\mathrm{a})}(t)$ to the left and $R_{\mathrm{s}}^{(\mathrm{a})}(t+1)$ to the right.

Finally, the probability for the event 'no errors in previously decoded channel packets' can be written as

$$P_{\mathrm{ne}}^{(\mathrm{a})}(i_{\mathrm{c}}) = \prod_{k=0}^{i_{\mathrm{c}}-1} P_{\mathrm{ne}}(k, i_{\mathrm{c}}) \qquad i_{\mathrm{c}} > 0\,. \tag{3.40}$$

The probability $P_{\mathrm{ne}}(k, i_{\mathrm{c}})$ of the event 'no error in the channel packet with index k' is taken into consideration in Eq. 3.40 only when the respective packet is a reference channel packet:

$$P_{\mathrm{ne}}(k, i_{\mathrm{c}}) = \begin{cases} \big(1 - P_{\mathrm{e}}(k)\big) & k \in \mathcal{R}(i_{\mathrm{c}}) \quad (3.41\mathrm{a}) \\ 1 & \text{otherwise} \quad , \quad (3.41\mathrm{b}) \end{cases}$$

where $P_{\mathrm{e}}(i_{\mathrm{c}})$ is the abbreviation for $P_{\mathrm{e}}\big(\Gamma(i_{\mathrm{c}}), \epsilon, R_{\mathrm{CP}}\big)$.

3.8 Average GOP distortion

The channel packets are transmitted over an error-prone channel and hence affected by channel errors. The distortion of one GOP after channel and source decoding is hence not known specifically, and the expectation is required for its description.

The expectation $\tilde{D}$ of a discrete variable $D_{\mathrm{CP}}(i_{\mathrm{c}})$ is defined as the sum over all (N) possible values, weighted by the probability of their occurrence:

$$\tilde{D} = \sum_{i_{\mathrm{c}}=0}^{N-1} D_{\mathrm{CP}}(i_{\mathrm{c}}) P(i_{\mathrm{c}})\,. \tag{3.42}$$

Relating this to the transmission of channel packets, $P(i_{\mathrm{c}})$ can be interpreted as the probability for the event 'loss of the channel packet with index i_{c} and no loss before that',

$$P(i_{\mathrm{c}}) = P_{\mathrm{e}}(i_{\mathrm{c}}) P_{\mathrm{ne}}^{(\mathrm{a})}(i_{\mathrm{c}})\,. \tag{3.43}$$

As already mentioned, a channel packet transmits one or several source packets. Consequently, $D_{\mathrm{CP}}(i_{\mathrm{c}})$ in Eq. 3.42 is replaced by the sum of error distortions as defined in Sec. 3.6.4,

$$D_{\mathrm{CP}}(i_{\mathrm{c}}) = \sum_{i=i_{\mathrm{s}}(i_{\mathrm{c}})}^{i_{\mathrm{s}}(i_{\mathrm{c}}+1)} \bar{D}_{\mathrm{e}}(i, i_{\mathrm{c}})\,, \tag{3.44}$$

to accumulate the distortions of all source packets contained in that specific channel packet. If the sum in Eq. 3.42 is split up to account for the fact that $P_{\mathrm{ne}}^{(\mathrm{a})}(i_{\mathrm{c}})$ is defined only for $i_{\mathrm{c}} > 0$, the expected distortion[4] $E[D_{\mathrm{GOP}}]$ after the transmission of N_{t} ($N_{\mathrm{t}} > 1$) channel packets of a GOP then becomes

$$\begin{aligned}\tilde{D}_{\mathrm{GOP}}(N_{\mathrm{t}}) = &\sum_{i=0}^{i_{\mathrm{s}}(1)} \bar{D}_{\mathrm{e}}(i,0)P_{\mathrm{e}}(0) \\ &+ \sum_{i_{\mathrm{c}}=1}^{N_{\mathrm{t}}-1} \sum_{i=i_{\mathrm{s}}(i_{\mathrm{c}})}^{i_{\mathrm{s}}(i_{\mathrm{c}}+1)} \bar{D}_{\mathrm{e}}(i,i_{\mathrm{c}})P_{\mathrm{e}}(i_{\mathrm{c}})P_{\mathrm{ne}}^{(\mathrm{a})}(i_{\mathrm{c}}) \\ &+ \bar{D}_{\mathrm{GOP}} \prod_{k=0}^{N_{\mathrm{t}}-1} \big(1 - P_{\mathrm{e}}(k)\big)\,, \end{aligned} \tag{3.48}$$

where

$$\bar{D}_{\mathrm{GOP}} = \begin{cases} 0 & R_{\mathrm{s}}^{(\mathrm{a})}(N_{\mathrm{SP}}) > R_{\mathrm{c}}^{(\mathrm{a})}(N_{\mathrm{t}}) \quad (3.49\mathrm{a}) \\ D_{\tilde{X}} & \text{otherwise} \quad (3.49\mathrm{b}) \end{cases}$$

and

$$D_{\tilde{X}} = \frac{1}{N_{\mathrm{f}}} \sum_{g=0}^{N_{\mathrm{f}}-1} \sum_{y=0}^{N_{\mathrm{y}}-1} \sum_{x=0}^{N_{\mathrm{x}}-1} w_{\mathrm{ss}}\big(i(g,y,x)\big) d\big(X(g,y,x), \tilde{X}(g,N_{\mathrm{l}}-1,y,x)\big)\,. \tag{3.50}$$

[4] There is some variation in the literature on the method of how to compute the expected image quality ($\widetilde{PSNR}_{\mathrm{GOP}}$) of a GOP. One approach is to define

$$\widetilde{PSNR}_{\mathrm{GOP}} = \sum_{i_{\mathrm{c}}=0}^{N_{\mathrm{CP}}-1} P(i_{\mathrm{c}})PSNR(i_{\mathrm{c}})\,, \tag{3.45}$$

where $PSNR(i_{\mathrm{c}})$ is the distortion (in *decibels*) associated with a packet with index i_{c}, and $P(i_{\mathrm{c}})$ is the probability determined by the probability mass function for each of the N_{CP} transmitted packets. However, $PSNR(i_{\mathrm{c}})$ is calculated by means of Eq. C.3, and since this is a convex function, $\widetilde{PSNR}_{\mathrm{GOP}}$ will be somewhat higher than the expected distortion defined by

$$\widetilde{PSNR}_{\mathrm{GOP}} = 10\log_{10}\frac{255^2}{\widetilde{MSE}_{\mathrm{GOP}}} \tag{3.46}$$

and

$$\widetilde{MSE}_{\mathrm{GOP}} = \sum_{i_{\mathrm{c}}=0}^{N_{\mathrm{CP}}-1} P(i_{\mathrm{c}})MSE(i_{\mathrm{c}})\,. \tag{3.47}$$

Eq. 3.47 is taken as the correct approach throughout this work if not mentioned otherwise.

Eq. 3.48 accounts for encoding (error resilience) and decoding (error concealment) strategies, as well as channel conditions. It is a recursive formulation of the in the literature often used

$$E[D_{\mathrm{GOP}}] = (1 - Pr)E[D_{\mathrm{ne}}] + PrE[D_{\mathrm{e}}] \tag{3.51}$$

to compute the expected distortion from the expectation $E[D_{\mathrm{ne}}]$ in case of no error, the expected error distortion $E[D_{\mathrm{e}}]$, and the error probability Pr.

It is obvious that the system assumes to have knowledge of ϵ or at least a rough estimate of it to optimize the system performance, i.e. minimize the expected distortion, for a specific operation point. In case of broadcasting or streaming, a limited number of feed-back channels can be installed to allow for decoder-encoder communication. The other alternative is to assume a worst-case scenario, that is, poor channel conditions, accepting an over-protection of the code stream and hence a suboptimum system performance. The channel state information (CSI) can further be gained through detecting a pilot signal or measurements of the received signal in a duplex connection.

Now, the encoding problem from Sec. 3.3 can be reformulated.

3.9 Joint source channel coding problem

In video streaming applications, the signal is often already source-encoded when the request for transmission of the code stream is made. The channel encoder can then adapt the code stream protection to the current channel conditions. As discussed in the previous sections, this is best done by means of the end-to-end distortion on a GOP basis. The allocation procedure has to evaluate the encoder's distortion rate functions $D\big(R_{\mathrm{c}}^{(\mathrm{a})}(i_{\mathrm{c}})\big)$ of each GOP for a set of combinations of channel code allocations, which may be a subset of the set of all possible combinations, as explained in Sec. 3.10. The structure of the framework is illustrated in Fig. 3.16. Compared to the decoder complexity, the codec can be described as asymmetric.

Summarizing, there is a joint source channel coding problem with respect to the optimum channel rate allocation. Let $\mathbf{\Gamma} = \big(\Gamma(0), \ldots, \Gamma(N_{\mathrm{CP}} - 1)\big)$, of which each element $\Gamma(i_{\mathrm{c}})$ is assigned to the channel packet with index i_{c}. The task is to then to find an N_{CP}-tuple $\mathbf{\Gamma}^*$ which is optimum

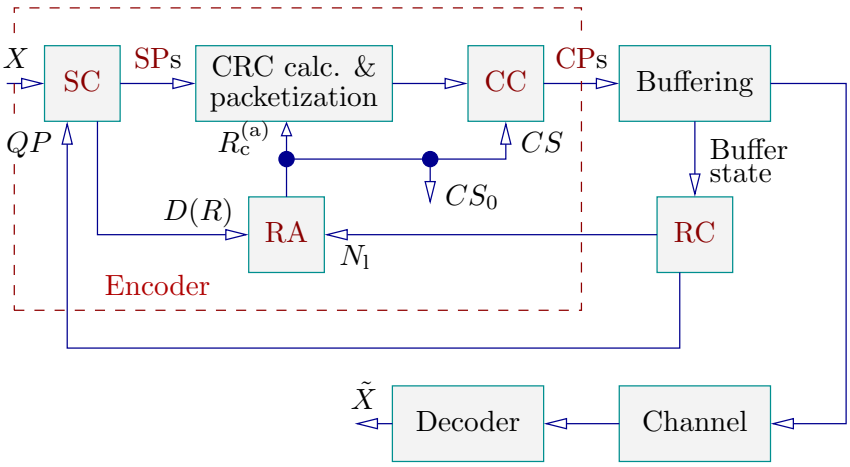


Figure 3.16 — Flow diagram for joint source channel encoding. SC: source coding; CRC: cyclic redundancy check; RA: rate allocation; CC: channel coding; RC: rate control. The channel block is explained in Sec. 3.7, and a detailed decoder flow diagram is drawn in Fig. 3.7

in the sense that it minimizes the GOP distortion,

$$\mathbf{\Gamma}^* = \underset{\mathbf{\Gamma}\in\mathcal{C}^{N_{\mathrm{CP}}}}{\operatorname{argmin}}\, \tilde{D}_{\mathrm{GOP}}\,, \tag{3.52}$$

provided that

$$\tilde{D}_{\mathrm{GOP}} = \tilde{D}_{\mathrm{GOP}}(N_{\mathrm{CP}})\,. \tag{3.53}$$

Eq. 3.52 is subject to the source rate constraint[5]

$$\sum_{i_c=0}^{N_{\mathrm{CP}}-1} R_c(i_c) = \sum_{i_s=0}^{N_{\mathrm{SP}}-1} R_s(i_s)\,, \tag{3.54}$$

which is given by the number N_{SP} of source packets and their lengths which are in turn quality-controlled. I.e., a high QP leads to a large layer MSE, and assuming one layer of one frame to be transported in a single source packet, the source packet lengths will be small. A low QP, on the other side, will result in long source packets as the MSE is accordingly reduced, which consumes much source coding rate. N_{SP} varies according to the number of layers, the number of frames per GOP,

[5]The expression 'source rate constraint' as used here is not to be confused with an explicit constraint put on the source encoder by external means. In fact, it is of implicit nature and a logical consequence of the encoder's fidelity constraint.

and the frame segmentation. N_{CP} depends on N_{SP} and additionally on the rate consumption of the channel code rate allocation. In case strong channel codes are chosen to protect the packet payloads, more channel packets will be generated than if weak codes are used, as the data of all source packets have to be transmitted for a successful decoding of the whole GOP.

This stands in contrast to the transmission of embedded code streams. An embedded stream can always be decoded to a complete GOP with a terminate-on-error decoding strategy. The earlier the error hits, the worse is the quality of all frames of that particular GOP. On the other side, with a hybrid code stream, the quality of the unaffected frames/segments is equal to those of the highest-quality layer and hence as good as possible, and all remaining segments are concealed, which increases the GOP distortion considerably.

Concluding, the rate constraint mentioned above is truly a decoder quality/fidelity constraint, since the code rate allocation algorithm does not terminate before all source bits are used up, and because the severity of channel disturbances influences the number N_{CP} of channel packets. To impose a direct channel rate constraint on the algorithm turns thus out to be impossible.

However, channel transmissions come very often with a rate constraint. To achieve a constant output bit rate, a buffer and associated rate control including a feed-back loop to the source encoding engine is utilized, as shown in Fig. 3.16. Channel packets are buffered before transmission, and a rate control mechanism ensures that a critical situation — buffer under- or overflow — is avoided. This corresponds to the definition of the video buffering verifier (VBV) in MPEG-1 Video or the rate control in H.261 [RH96]. In case the source is generated dynamically, rate control can be accomplished by e.g. adjustment of one or several layer QPs to higher values, by reducing the size of one GOP (dropping of frames), by switching to a lower-resolution video, or by dropping of high-quality layers for at least the current GOP. In case the source is pre-coded, only the latter possibility applies, as long as transcoding is not desired.

Fig. 3.17 shows typical curves of the bit rate consumptions of frames in a GOP. Obviously, R does not only vary with different frame types (I/P), but depends also on e.g. the detailedness and motion in the source signal. Nevertheless, the quality constraint in H.264 leads to a roughly constant $PSNR$.

A generalization of the variable-rate encoding problem posed above

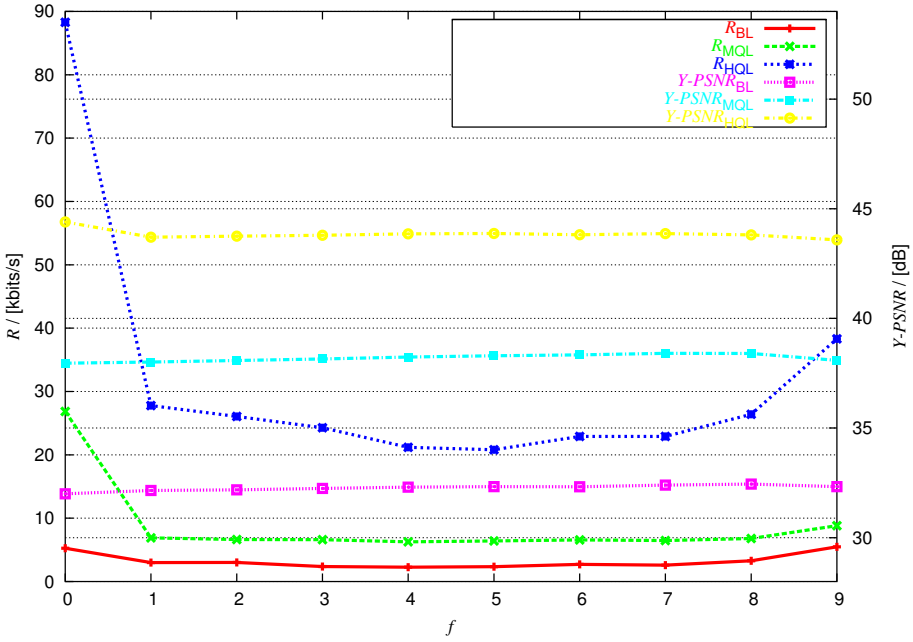


Figure 3.17 — Progression of source bit rate R and luminance $PSNR$, both per frame, for one GOP. The video coding mode is IPP

is that a code stream of finite length is split up into packets of which each is associated with an error metric. For each packet, the encoder has to keep track of the information what the packet's error metric is and which other packets it depends on. The system determines in other words the channel coding costs, i.e. rate, which the optimum protection requires, given an already source-encoded stream, a set of channel codes, and information about the channel state. The optimum protection is defined as the channel code distribution which yields minimum distortion.

A note on Eq. 3.54 may be in place. In [HF03a], a similar algorithm was developed, based on a channel rate constraint instead of a source rate constraint, of the form

$$\sum_{i_c=0}^{N_{CP}-1} \big(R_c(i_c) + 24 + C(i_c)\big) = R_{CP} \cdot N_{CP}\,, \tag{3.55}$$

which is given by the channel packet's payload length $R_c(i_c)$, the length of the code specifier and check sum (in *bits*), and the number of bits consumed by the channel code, $C(i_c)$. The packet length R_{CP} is predefined

by the set of channel codes, and N_{CP} is passed to the algorithm for rate control. This approach yields good results, but has some basic limitation due to the non-embedded nature of the code stream. Channel codes of decreasing protection capabilities are assigned to the layers of the first frame, e.g. the payload lengths in bytes are 255, 255, 341, 255, and 513. The last channel packet which transports high-quality layer information of the first frame transports however also a part of the next frame's base layer. An appropriate error metric is determined for that packet and added to the overall distortion which turns out to be the minimum distortion after termination of the algorithm. That is, the probability of this packet's erasure is equal to one, and as a consequence all subsequent packet distortions are ignored due to the accumulated no-error probability. In other words, the first frame is transmitted correctly, but all remaining frames are concealed and contain therefore no useful information. It is concluded that it is impossible to convey a pre-encoded hybrid code stream with a channel rate constraint.

The implementation of the rate allocation block in Fig. 3.16 is explained in detail in the following section.

3.10 The Viterbi algorithm

The global optimum solution to Eq. 3.52 can be determined by a brute-force attempt, as discussed in Sec. 3.5, but this is limited by N_{SP}. In contrast to that, the application of dynamic programming [Bel57a] on the optimization problem provides a low-complexity approach. The solution is here implemented by the Viterbi algorithm (VA) [For73], which computes the most likely state sequence in a hidden Markov model (HMM), given the observed outputs, by utilizing a trellis structure of paths which represent possible decisions. The optimum solution in a VA sense is the path which, of all paths, comes closest to or is identical with the global optimum.

The trellis is constructed according to the example shown in Fig. 3.18 with three channel codes. All out-going edges of a node correspond to the available channel codes, in dynamic programming called *decisions*. All in-coming branches of a node correspond to the number of already channel-encoded data bits, which hence is the same for those branches. A node is also referred to as trellis *state* s_{t} and defined by $R_{\mathrm{c}}^{(\mathrm{a})}(i_{\mathrm{c}})$. The next state is dependent on only the current state and the current *action*, i.e. decision. The problem of finding the optimum combination of channel

codes is hence a finite Markov decision process [Bel57b].

The payload rate of each packet is $R_{\mathrm{c}}(i_{\mathrm{c}})$. Each trellis edge is associated with an error metric, i.e. $\tilde{D}_{\mathrm{GOP}}(N_{\mathrm{t}})$. Finally, a *stage* is defined as one transmitted channel packet, and the rate constraint expressed by Eq. 3.54 is hence inherent to the trellis. That is, given a fixed set of channel codes, the size of the trellis varies with the source coding parameters as explained above, as well as with the channel conditions. At the last stage, the nodes are also called *leaves*, and the leave with (overall) minimum error has to be traced back along the in-coming branches with the smallest local error metric. That is, if Eq. 3.48 is reformulated as

$$\tilde{D}_{\mathrm{GOP}} = \sum_{i_{\mathrm{c}}=0}^{N_{\mathrm{CP}}-1} D(i_{\mathrm{c}})\,, \tag{3.56}$$

with $N_{\mathrm{t}} = N_{\mathrm{CP}}$, then the overall minimum distortion $\tilde{D}^*$ becomes

$$\begin{aligned}
\tilde{D}^* &= \tilde{D}(\mathbf{\Gamma}^*) \\
&= \min_{\mathbf{\Gamma}\in\mathcal{C}^{N_{\mathrm{CP}}}} \tilde{D}_{\mathrm{GOP}} \\
&= \min_{\mathbf{\Gamma}\in\mathcal{C}^{N_{\mathrm{CP}}}} \sum_{i_{\mathrm{c}}=0}^{N_{\mathrm{CP}}-1} D(i_{\mathrm{c}}) \\
&= \sum_{i_{\mathrm{c}}=0}^{N_{\mathrm{CP}}-1} \min_{\Gamma(i_{\mathrm{c}})\in\mathcal{C}} D(i_{\mathrm{c}}) \\
&= \sum_{i_{\mathrm{c}}=0}^{N_{\mathrm{CP}}-1} \min_{\Gamma(i_{\mathrm{c}})\in\mathcal{C}} \tilde{D}_{\mathrm{GOP}}(i_{\mathrm{c}})\,.
\end{aligned} \tag{3.57}$$

It is noted that the last equality is only valid while back-tracing from the global minimum to ensure that not a local minimum be found instead[6]. The trellis is complete when all leaves are reached. It is stressed that, for more than one channel packet, the leaves are located at different stages.

The complexity of the Viterbi algorithm grows moderately with an upper bound of $O(||\mathcal{C}|| \cdot N_{\mathrm{CP}}^2)$ because the number of states in each stage increases in a linear manner. It is noted that typically the complexity is considerably below the bound. For weak channel codes, leaves are

[6]Considering the $PSNR$ instead of the MSE, a maximization problem persists, which the author would entitle the *climbing-mountaineer problem*. Typically, a climber attempts reaching the highest summits. If a cloud layer covers the sky, which is unfortunately the case sometimes, all he sees are the feet of the mountains, and he cannot decide where to start the climb.

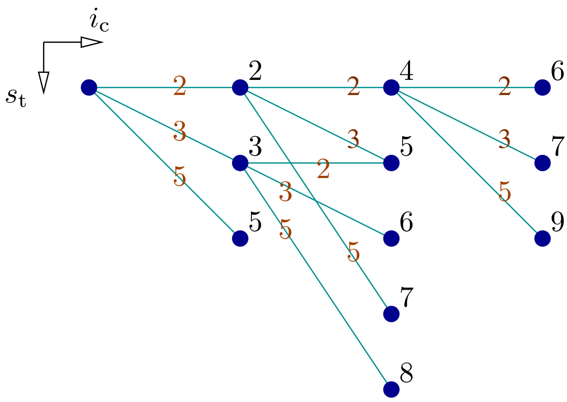


Figure 3.18 — Trellis with initializing stage, payload rates $\{2, 3, 5\}$, and a source bit stream length of five

likely to be reached already at an early stage, and no further edges can be computed for those because the source rate is used up. This is also illustrated in Fig. 3.18.

Consider the channel codes Γ_i, $i = 0, \ldots, ||\mathcal{C}|| - 1$, of rate $r_{\text{cc},i} = k_i/d$, where k_i is the number of source symbols transported by the code i per channel unit of length d. From the initial node, there are $||\mathcal{C}||$ branches linking to the following nodes. The number of branches from one state to the other does, however, not increase with $||\mathcal{C}||$ due to the fact that several paths, e.g. those defined by (Γ_1, Γ_2) and (Γ_2, Γ_1), reach the same state. Out from the in-coming branches of a particular node, all branches except the one with the local minimum error metric can be removed.

The sketched solution for channel rate allocation depends on information about distortion and rate of all frames belonging to a particular GOP to achieve the global minimum distortion. Even though this approach inherits a certain latency of the time needed to build the trellis and calculate the error metrics and other necessary information, the building of the trellis can start as soon as the first source packet is readily coded and continues as source encoding proceeds. Before the first channel packet is sent, however, its channel code must have been determined, and since this is not the case before the end of the GOP is reached, there will be a delay of at least one GOP, here 1 s, which confines the set of possible applications to non-conversational services. This moderate delay is comparable to that of other approaches like [Ban02] and [tT00] and has to be accepted to avoid inferior system performance. Nevertheless, the devised system can be deployed in video streaming, as delay is not a critical

measure there, and near-real-time applications like broadcasting, as well as in off-line encoding and benchmarking.

The suboptimality of the VA has been addressed in [Ban02], where it is shown that the VA algorithm rarely produces a suboptimum solution. This theoretical discussion can be confirmed by results of the VA developed here in a small number of test cases with moderate complexity. In each case, the solution determined by the VA is identical with the global optimum solution as found by an extensive (brute-force) search over the space of possible solutions.

3.11 Experiments, results, and discussion

The properness of the aforementioned derivations and the functionality of the VA is evaluated in the following by means of software implementations and simulations. But first of all, since the definition of parameters which describe the performance of an encoding system varies in the literature, some parameters of interest are defined briefly in the following.

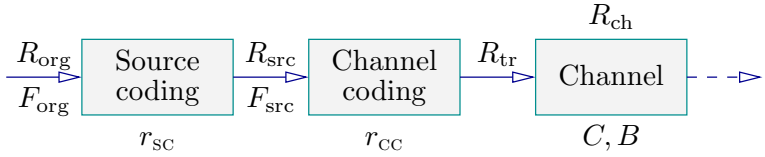


Figure 3.19 — A generic encoder. The modulation of the (digital) signal at the channel input to the (analog) channel signal is included in the channel block

The original signal, at a frame rate F_{org} (in *fps*), is generated with a *rate* of R_{org} (in *information bits per second*) and fed into the source encoder which achieves the *compression ratio* r_{SC} (in *information bits per source bits*). Before channel coding, the (source) rate of the compressed signal is R_{src} (in *source bits per second*), at an encoded frame rate F_{src} (in *fps*). The channel encoder is mainly characterized by the overall *channel coding rate*[7] r_{CC} (in *source bits per channel bits*). Its output stream is conveyed over the channel with the *transmission rate* R_{tr} (in *channel bits per second*). Finally, the channel is characterized by its *capacity*[8] C (in *source bits per channel bits*) and the *channel rate* $R_{\text{ch}} = R_{\text{tr}}C$ (in *source bits per second*) which specifies the maximum source bit amount which

[7] The unit of r_{CC} is not always implicitly mentioned throughout the text.
[8] The unit of C is not always implicitly mentioned throughout the text.

can be reliably transmitted over an error-prone connection. Fig. 3.19 shows a generic encoding system and its associated parameters. R_{ch} is not to be confused with the channel's *bandwidth* B (in *bps*) which is the upper limit of number of bits the channel is able to transmit without congestion.

3.11.1 Basic experiments

This section includes the coding performance evaluation for various imagery with different sizes and source-encoded at various rates, given different channel code sets $\mathcal{C}$ and a fixed channel packet length. Parts of the subsequent sections have been presented in [Hal04b] and [Hal04a].

All test sequences as specified in App. A are considered, further characterized by $R_{\text{org}} = 36.5\,\text{Mbps}$ with CIF-size and $R_{\text{org}} = 9.1\,\text{Mbps}$ with QCIF-size imagery, and $F_{\text{org}} = 30\,\text{fps}$ for both. The video coding mode is IPP, where each GOP is preceded with an INTRA frame. Other basic coding parameters are $N_{\text{f}} = 10$ ($F_{\text{src}} = 10\,\text{fps}$), which means a frame skip of two, $N_{\text{l}} = 3$, and $N_{\text{s}} = 1$, i.e. a slice equals a frame. The values of QP of the respective layers are given in Tab. 3.3. If not mentioned otherwise, the following results are averaged over all GOPs of an image sequence[9]. This method gives valid results in so far as the impact of errors is limited to one GOP only. All experiments assume in-order transmission of channel packets.

As already mentioned, the target application in mind is wireless ATM-wise transmission, the channel model being a BSC as defined in Sec. 3.7. A set of eight codes is employed for channel encoding and error correction, the code rates being $r_{\text{CC}}(i) = k(i)/d$ as given in Tab. 3.4, with the common denominator $d = 12$. For a 517-byte packet size, i.e. $R_{\text{CP}} = 4136\,\text{bits}$, and a packet structure as discussed in Sec. 3.5, this results in the payload lengths $R_{\text{c}}(i)$, as listed in Tab. 3.4. Yet, the corresponding code rates are given in the last row of the table. In the following, the three QP sets {32,24,16}, {40,32,24}, and {48,40,32} are denoted as quality sets A, B, and C, respectively.

Padding is used as necessary such that all packets have the same length. The average overhead, which is defined as the amount of data spent for padding and to code CS and CRC, then becomes 7 bytes per packet. The codes consist of punctured parallel concatenated recursive

[9] Distortions given (in *decibels*) are averaged in non-logarithmic scale first and then converted to the logarithmic scale according to Eq. C.3.

Video name	QPs	$PSNR_{\mathrm{HQL}}$	R_{src}	r_{SC}
`Container`	32, 24, 16	45.05	1599.40	22.8
	40, 32, 24	38.93	595.11	61.3
	48, 40, 32	33.67	230.92	158.0
`Foreman`	32, 24, 16	44.94	2036.28	17.9
	40, 32, 24	38.92	797.74	45.7
	48, 40, 32	33.54	361.47	101.0
`Mobile&Calendar`	32, 24, 16	44.35	4222.03	8.6
	40, 32, 24	37.21	1904.02	19.2
	48, 40, 32	30.79	820.06	44.5
`Mother&Daughter`	32, 24, 16	46.24	905.17	40.3
	40, 32, 24	41.21	354.35	103.0
	48, 40, 32	35.95	188.54	193.6
`Foreman` (QCIF)	32, 24, 16	45.08	505.36	18.1
	40, 32, 24	38.69	218.48	41.8
	48, 40, 32	32.85	109.83	83.1
`Silent` (QCIF)	32, 24, 16	45.39	367.69	24.8
	40, 32, 24	38.60	176.02	51.8
	48, 40, 32	32.68	93.47	97.6

Table 3.3 — Source coding performance with $\Delta QP = 8$. $PSNR_{\mathrm{HQL}}$ is given (in *dB*), and the unit of R_{src} is *kbps*. The image size is CIF if not mentioned otherwise

i	1	2	3	4	5	6	7	8
$k(i)$	4	5	6	8	9	10	11	12
$R_{\mathrm{c}}(i)$	169	212	255	341	384	427	470	513
$r_{\mathrm{CC}}(i)$	0.33	0.41	0.49	0.66	0.74	0.83	0.91	0.99

Table 3.4 — Specification of channel code set A. The units of $k(i)$, $R_{\mathrm{c}}(i)$, and $r_{\mathrm{CC}}(i)$ are *bits*, *bytes*, and *source bits/channel bits*, respectively

convolutional (PPCRC) codes as recommended in [AR99] and [RM00]. Based on $r_{\mathrm{CC}}(i)$, the probabilities P_{e} of a channel packet having at least one bit error after 20 channel decoder iterations have been computed in extensive Monte-Carlo simulations of 10,000 blocks in [BBF02] and can hence be tabulated for use in Eq. 3.48.

Fig. 3.20 gives an overview of the capabilities of the different channel codes. The steep portion of the curves is known as the *waterfall region*, and the tapering-off near the bottom marks a flatter region denoted as

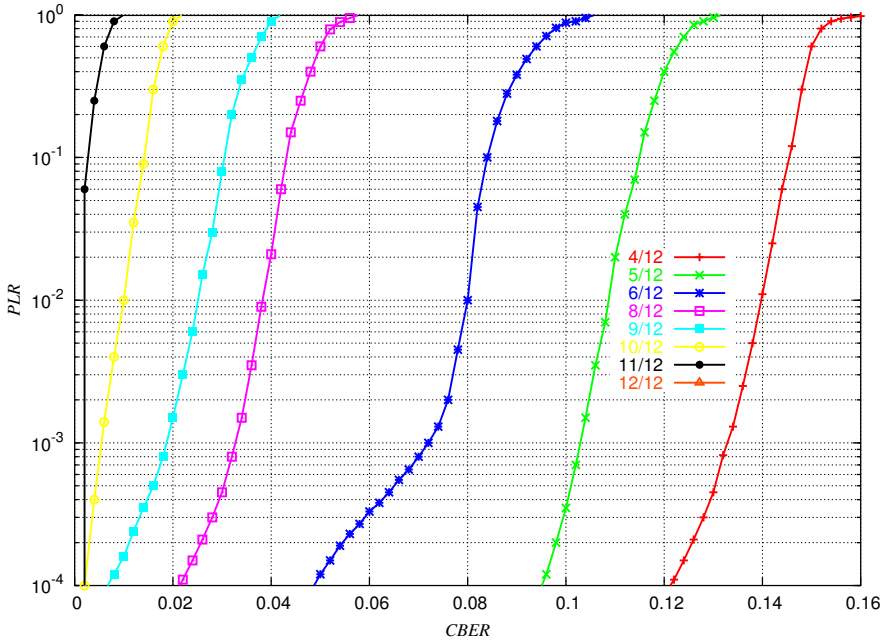


Figure 3.20 — The packet loss rate PLR of each PPCRC code as a function of the channel's bit error rate $CBER$ (or ϵ). The curve of code eight with the rate 12/12 is identical with the upper box boundary of the plot, i.e. $P_e = 1$ with $\epsilon > 0$

error floor. Generally, error-free coding performance for practical codes is considered to occur at a packet loss rate of 10^{-5} [Ban02]. The codes show a 'binary' behavior, i.e. the range of cross-over probabilities ϵ for which the PLR takes on values between the error floor and 1 is quite small.

Subsequently, the performance of the VA is measured in terms of GOP $PSNR$, no-error probability, channel code rate, and transmission rate, whereas the channel conditions are represented by C which can mapped to the channel bit error rate as shown in Fig. 3.14. The results are all consistent, i.e. valid for all image sequences, such that only a subset of the generated plots is shown here. As seen in Fig. 3.21, the VA aims at keeping the $PSNR$ of the decoded GOPs almost constant at the error-free HQL $PSNR$ value (compare to Tab. 3.3) over a wide range of channel bit error rates. This observation also applies in mismatch situations with a $\Delta\epsilon$ (defined below) of up to 15%. One exception is when ϵ becomes too high to ensure error-free transmission; then, the protecting capability of

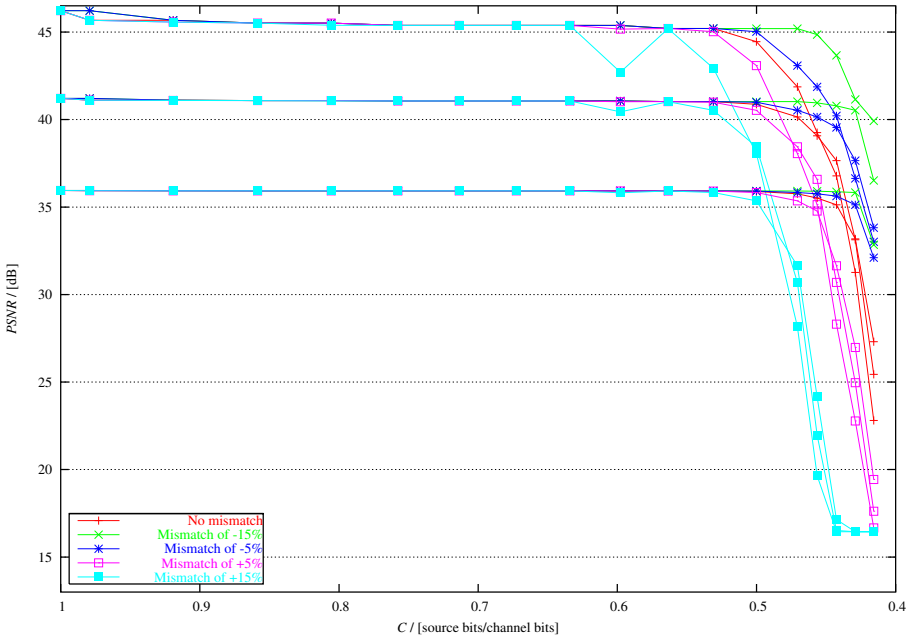


Figure 3.21 — Average GOP $PSNR$ as a function of the channel capacity with the CIF-size `Mother&Daughter` video. The three curve sets correspond, from top to bottom, to the QP sets {32,24,16}, {40,32,24}, and {48,40,32}. Also shown are the $PSNR$ values in case of channel mismatch situations

the strongest channel code is exceeded, here around $\epsilon = 0.11$ or $C = 0.5$. More on this topics later in the discussion of no-error probabilities. The other exception is the $PSNR$ step at the transition from the error-free case ($\epsilon = 0$) to $\epsilon > 0$. The step height is the larger, the higher the source rate, which is due to the accumulation of distortions contributed by each channel packet. This effect is therefore best observed with quality set A.

Four mismatch situations are included in the $PSNR$ figures as well, where ϵ as mentioned above is the estimated channel bit error rate in contrast to the true error rate,

$$\epsilon_{\text{true}} = \epsilon + \Delta\epsilon\,, \tag{3.58}$$

with $\Delta\epsilon = m \cdot \epsilon$ and $m = \{-0.15, -0.05, 0.05, 0.15\}$. It can be observed, as one would expect, that the degradation in $PSNR$ occurs with regard to how bad the true channel conditions are; the $PSNR$ curves drop the earlier, the worse the channel quality really is.

The aforementioned behavior is the same for all videos while coded at different rates, i.e. quality sets. However, the $PSNR$ deviation in mismatch situations is at lower rates smaller than at higher rates. This is in concordance with Eq. 3.48, as more channel packets mean a larger accumulation of the mismatch distortion and hence a lower $PSNR$.

One final comment to the 'saturation effect' of a minimum average $PSNR$ at around 16.4 dB in Fig. 3.21. Here, it is expected that not even a single intact channel packet reach the source decoder, which then carries out mean concealment, which in turn results in this low quality without any useful information. It is further remarkable that the mismatch curves are always grouped together. This is explained by the fact the QP sets are intersecting with each other, and because layer concealment is based on the next lower-quality layer.

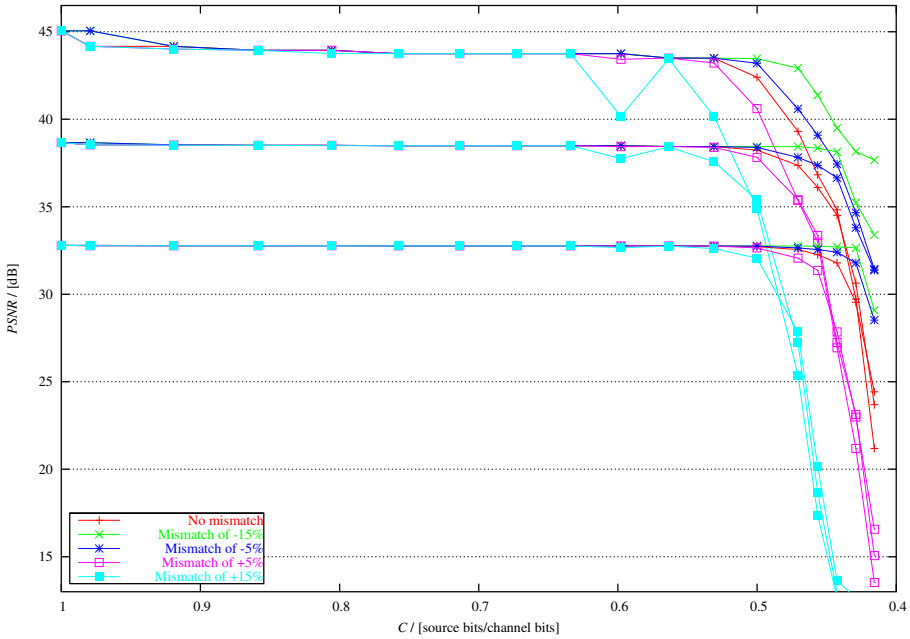


Figure 3.22 — Average GOP $PSNR$ as a function of the channel capacity with the QCIF-size `Foreman` video. The three curve sets correspond, from top to bottom, to the QP sets {32,24,16}, {40,32,24}, and {48,40,32}. Also shown are the $PSNR$ values in case of channel mismatch situations

Fig. 3.22 shows similar curves with the QCIF-size `Foreman` video as a proof for the efficacy of the proposed packetization scheme and rate

allocation algorithm for different visual material and different sizes. In the error-free case ($C = 1$), the average GOP $PSNR$s are identical with the corresponding values in Tab. 3.3. The curve degrades then very slowly from, with QP set A, 44.2 dB at $C = 0.98$ to 43.5 dB at $C = 0.53$, before it drops away dramatically in something in FEC coding often circumscribed as cliff effect. The deviation of curves of a mismatch of 15% from the error-free curves at certain channel capacities is discussed below.

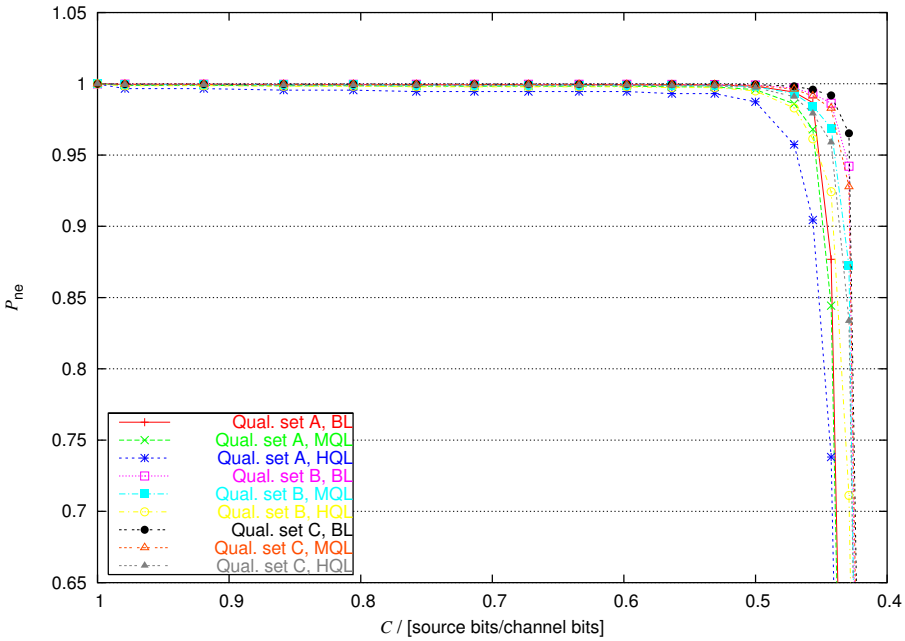


Figure 3.23 — Accumulated no-error probability per GOP as a function of the channel capacity with CIF-size `Mother&Daughter`

The $PSNR$ curves are tightly connected to the progression of the no-error probabilities of each quality layer, which decrease slightly with increasing ϵ. This is depicted in Fig. 3.23 and, in detail, in Fig. 3.24. The plotted values are the probabilities for the event 'no error in any channel packets transporting whole source packets or a part of a source packet containing that respective layer', P_{ne}. It is seen from the figures, as expected, that the lower the layer QP, the higher the source rate of that certain layer, and the lower in turn P_{ne}. A high source rate means many conveyed channel packets with a certain layer, and with a stationary channel, P_{ne} accumulates monotonically to lower values. As a consequence, the higher the source rate, the steeper the cut-off at the

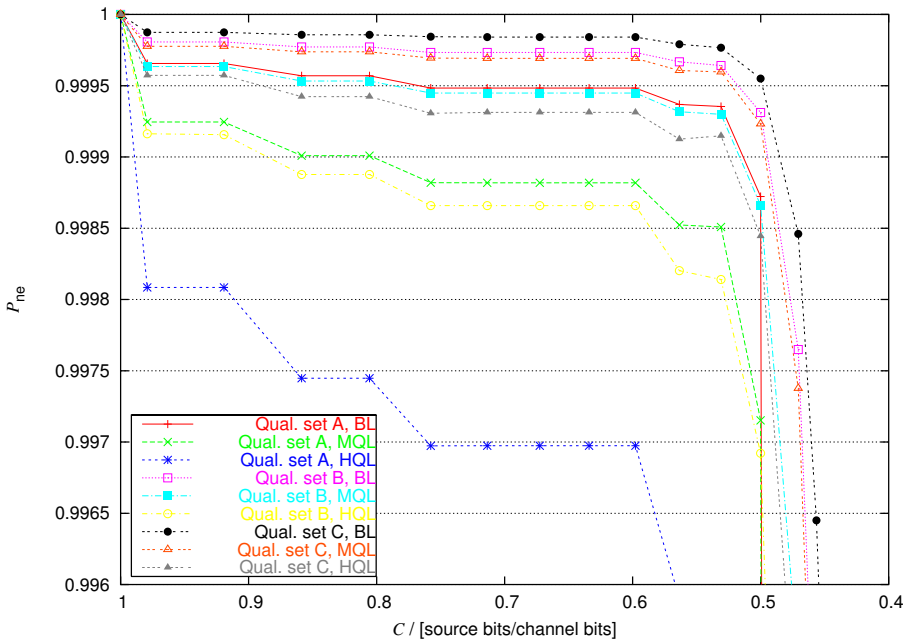


Figure 3.24 — Accumulated no-error probability per GOP as a function of the channel capacity with QCIF-size `Foreman`

cliff region. The accumulation is further more distinctive for the HQL than for the medium-quality layer (MQL) for which it is in turn more significant than for the BL. Also, the following relationships hold true:

$$P_{\mathrm{ne}}^{(\mathrm{BL})} > P_{\mathrm{ne}}^{(\mathrm{MQL})} > P_{\mathrm{ne}}^{(\mathrm{HQL})}\,, \tag{3.59}$$

and

$$P_{\mathrm{ne}}^{(\mathrm{BL_C})} > P_{\mathrm{ne}}^{(\mathrm{MQL_B})} > P_{\mathrm{ne}}^{(\mathrm{HQL_A})}\,. \tag{3.60}$$

Finally, a source rate comparison of the second and the third curve (from bottom to top) in Fig. 3.24 confirms that

$$\sum_{\mathrm{BL_C}+\mathrm{MQL_C}+\mathrm{HQL_C}} R_{\mathrm{s}}(f) > \sum_{\mathrm{BL_A}+\mathrm{MQL_A}} R_{\mathrm{s}}(f) \tag{3.61}$$

for all frames f of a GOP, which is what one would expect intuitionally. In can be concluded that the proposed scheme determines the channel code rate required to be able to decode the video with the highest possible image quality.

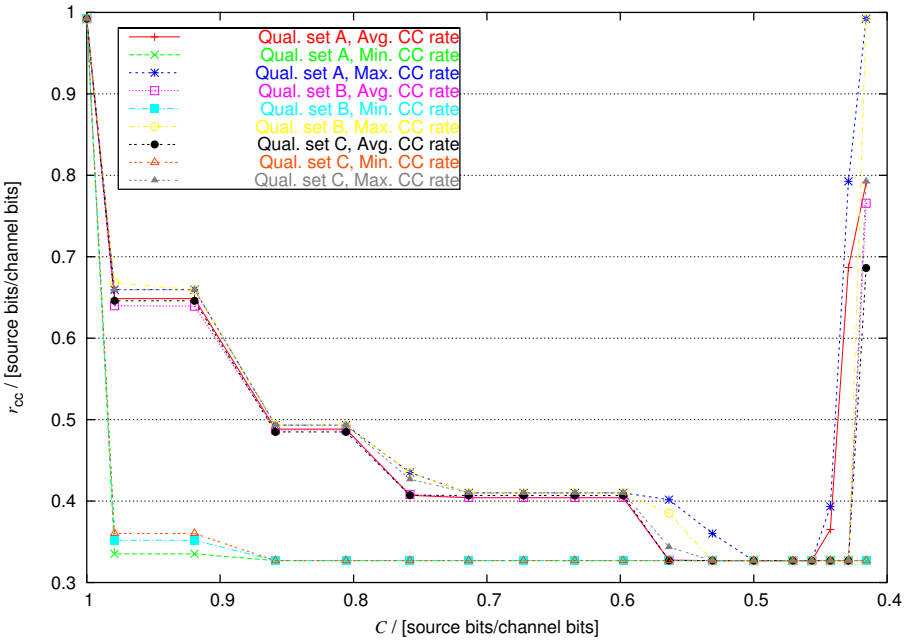


Figure 3.25 — Overall channel code rate per GOP as a function of the channel capacity with CIF-size `Mother&Daughter`

As the GOP image quality is kept nearly constant, the grade of protection of the bit stream to transport varies with the channel conditions. This is depicted in Fig. 3.25. The overall code rate r_{CC} decreases monotonically as the channel conditions become worse, except for the cliff area, where an error-free bit stream after channel decoding can't be guaranteed anymore. Then, the VA suggests that the best strategy be to pack as many payload bits into the channel packets as possible in order to convey a maximum of source information before the first error strikes. Thus, the channel code rate goes up. It is stressed that, as the algorithm aims at the maximization of the channel code rate, plain use of the strongest channel code only would not give maximum GOP $PSNR$ due to an increase of distortion accumulation in Eq. 3.48.

The behavior of all curves of r_{CC} over C is identical for all test material. In Fig. 3.25 and Fig. 3.26, also minimum and maximum channel code rates are plotted in addition to the average code rate. There is not much variance in the distribution of channel codes since the curve of the maximum rate is near the curve of the average rate for all values of C, except for the interval from 0.6 to 0.5. With other words, a single or at

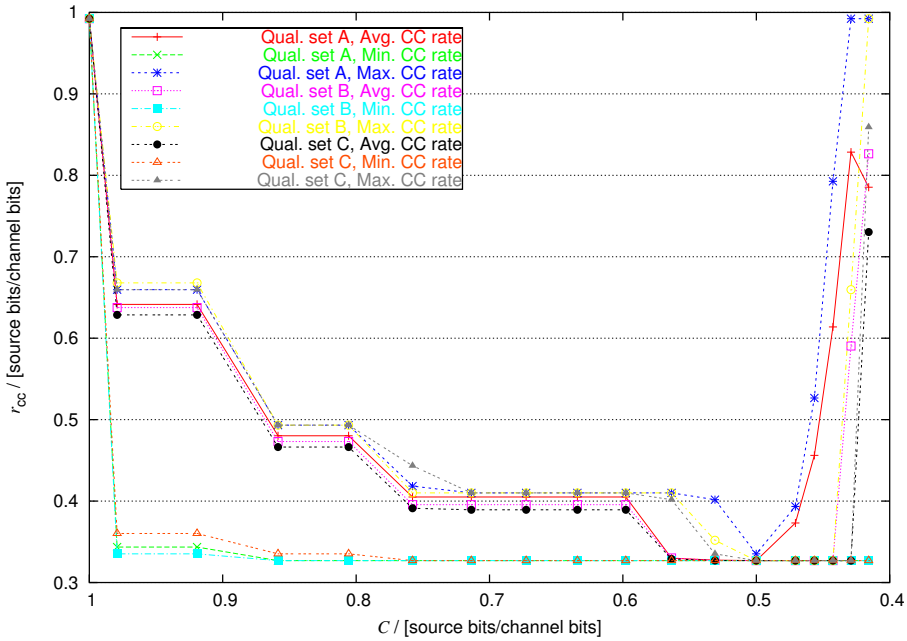


Figure 3.26 — Overall channel code rate per GOP as a function of the channel capacity with QCIF-size `Foreman`

most two channel codes are preferably assigned to the channel packets. The strongest available channel code has always been alloted, though, as can be seen from the curve of the minimum rate, but obviously not to a high degree. Accordingly, other work (concerning embedded code streams and still-image transmission) indicates nearly all of the gain is obtained with at most three channel codes operating in the vicinity of the target channel cross-over probability [ZAA00]. Yet, with a good channel, that is $C \to 1$, the code of minimum code rate is assigned less often, as the minimum-rate curve slowly rises. This is also what one would expect.

It is indeed surprising that the progress of r_{CC}, a stair-case function apart from the cut-off area, is almost the same for all tested videos at different source rates, i.e. quality sets, and sizes. Seemingly, the proposed scheme leads to a very general solution where the channel code rate only depends on the channel capacity. The curve's steps are quite distinct, i.e. steep, and appear to occur always at the same values of C. Compared to Tab. 3.4, it is found that r_{CC} at the four major steps is approximately equal to the code rates of the codes 1, 2, 3, and 4.

Another comparison between the location of steps of r_{CC} and the pro-

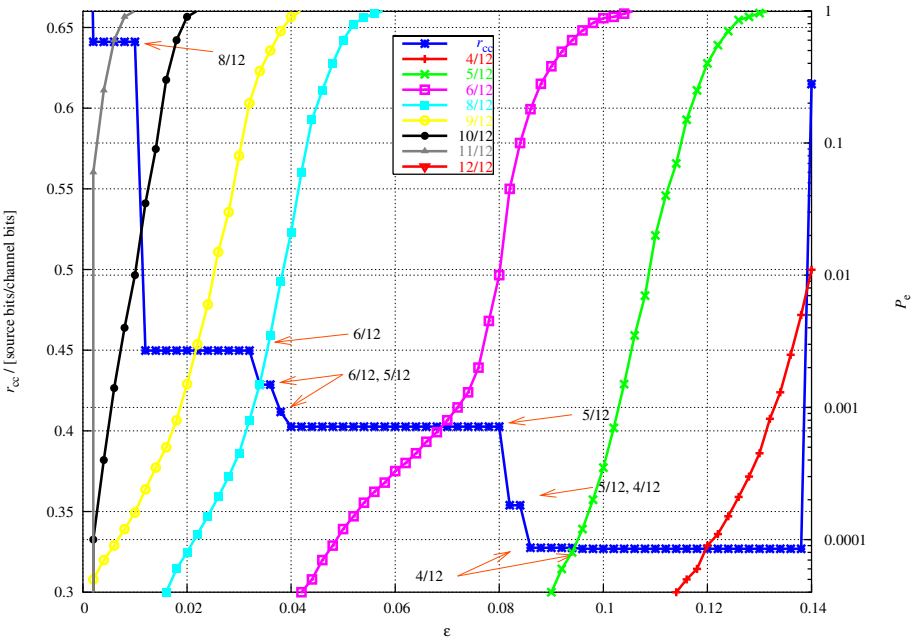


Figure 3.27 — Detail of the overall channel code rate of the fourth GOP of QCIF-size `Silent`, coded with quality set B, as a function of the channel capacity. The step size of ϵ is 0.002

gression of P_e of each channel code is therefore provided in Fig. 3.27 with a resolution of ϵ of 0.002 and only a single GOP. At an error-free channel, the code of rate 12/12 is used throughout the bit stream. With $\epsilon \in (0, 0.01]$, almost solely the code of rate 8/12 is assigned. An assignment of the 11/12-, 10/12-, and even the 9/12-rate code does obviously not lead to the minimum distortion for the GOP. For $\epsilon \in [0.012, 0.032]$, mainly the 6/12-rate code is used. As ϵ increases further, the distortion is not minimum anymore, and hence the channel rate allocation switches from a preference of the 6/12- to preferring the 5/12-rate code with $\epsilon \in [0.04, 0.08]$, after using mixtures of codes of rate 6/12 and 5/12 within the interval $[0.034, 0.038]$. With $\epsilon \in [0.082, 0.084]$, there is another mixture of codes of rate 5/12 and 4/12, after which — $\epsilon > 0.086$ — the 4/12-rate code becomes the most often alloted code. The scheme works up to a $CBER$ of 0.138; after that, the strategy as discussed above is followed to pack as many source data into the channel packets as possible. It is noted that the cut-off occurs the earlier, the more complex a video.

The progression of r_{CC} curves explains further the deviation of mis-

match curves from the error-free curves in for instance Fig. 3.21. The $PSNR$ differences occur always at the right boundary of a r_{CC} step, i.e. at C equal to 0.919, 0.806, and 0.598. At these points, a certain code distribution is assigned a last time before a substantially other distribution is used at lower channel capacities. This means in turn that the code capabilities are then exceeded, which leads to an increased PLR for that particular code and hereby a higher distortion. This contribution to the GOP distortion is the higher, the worse the channel conditions are. Consider e.g. the point $C = 0.598$, synonymous with $\epsilon = 0.08$, which is on the error floor of the code of rate 5/12. With a 15% mismatch, $\epsilon = 0.092$, which is located in the code's waterfall region. It appears thus that the VA manages to determine the transition area between error floor and waterfall region of each code, and that the use of a particular code is avoided when the channel bit error proceeds too far into the waterfall region.

It is concluded that the codes of rate 11/12, 10/12, and 9/12 are almost never used and can thus be removed from the set of channel codes with an insignificant change in performance. They may be needed for $CBER$s lower than $2 \cdot 10^{-3}$. The 12/12-rate code is, however, important for error-free transmission, i.e. a noiseless channel, whereas the other codes cover each a certain range of channel conditions. The number of assigned codes is seldom larger than three, often only two, but preferably one; a result which was obtained, too, for embedded bit streams in [Ban02]. Finally, before the PLR of a code takes on values which would lead to a strong contribution to the overall GOP distortion due to multiplication with the adequate channel packet distortion, the code is neglected by the allocation algorithm which switches to assign a stronger code of a minimally smaller rate, i.e. from e.g. $k = 5$ to $k = 4$.

When the channel quality deteriorates, the source is better protected, and as a consequence the transmission time goes up, which is synonymous with an increase of transmission rate. In other words, there is a trade-off between the quality of the reconstructed video and the delay needed to convey the compressed data. To avoid network congestion,

$$B \geq R_{\mathrm{tr}} \,. \tag{3.62}$$

The transmission rate R_{tr} is proportional to the channel packet size R_{CP} and the number N_{CP} of channel packets, and is hereby inverse proportional to r_{CC}. This is illustrated in Fig. 3.28 and Fig. 3.29 which both show an inverse stair-case function in contrast to e.g. Fig. 3.26. As the source rate differs with quality sets A to C and image size, also the re-

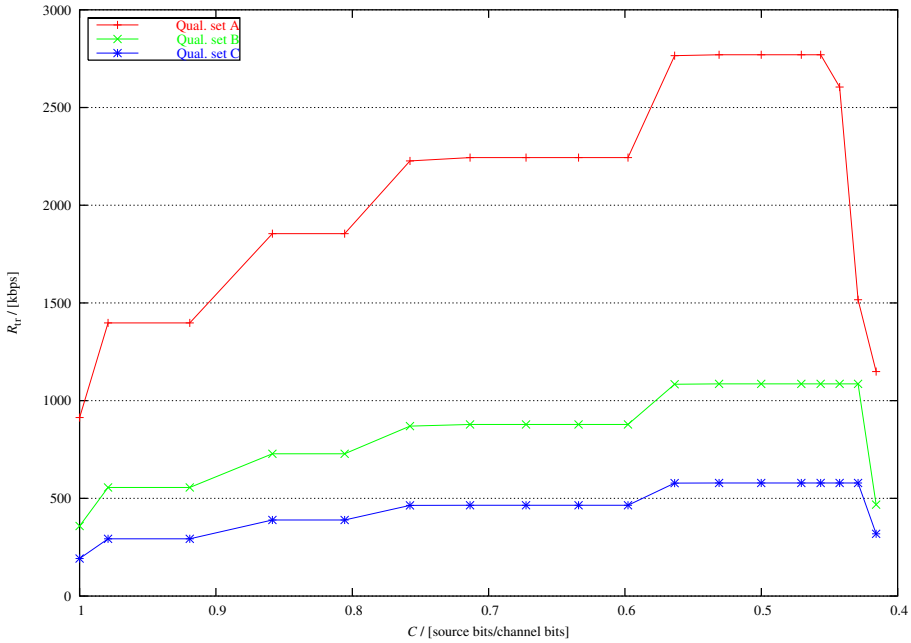


Figure 3.28 — GOP transmission rate as a function of the channel capacity with CIF-size `Mother&Daughter`

sulting transmission rate varies, even though an almost identical channel code allocation is used. It is clearly seen that the algorithm aims at the maximization of r_{CC}, which is synonym with a minimization of R_{tr}.

As plotted in Fig. 3.30 and Fig. 3.31, the code distributions are quite contrasting for the given $CBER$s. With $\epsilon = 0.008$, mostly the 8/12-rate code is alloted (determined by means of Tab. 3.4), while 63 channel packets are transmitted. This latter number rises to 90, 94, 101, 114, and 123 with the corresponding other values of ϵ, whereas the major code rate shifts from 6/12 to 5/12, 5/12, 4/12, and finally 4/12, respectively. The mixture distributions at $\epsilon = 0.034$ and $\epsilon = 0.082$ are further congruent with Fig. 3.27 which shows an overall code rate between 0.41 and 0.45, and between 0.33 and 0.41, respectively. According to the plots, there is no obvious relationship between the strength of protection and e.g. the layer to which the transmitted data belong. This is due to the non-embedded nature of the bit stream.

The VA's speed and complexity depend, as discussed above, on the input signal, the set of available channel codes, as well as the channel

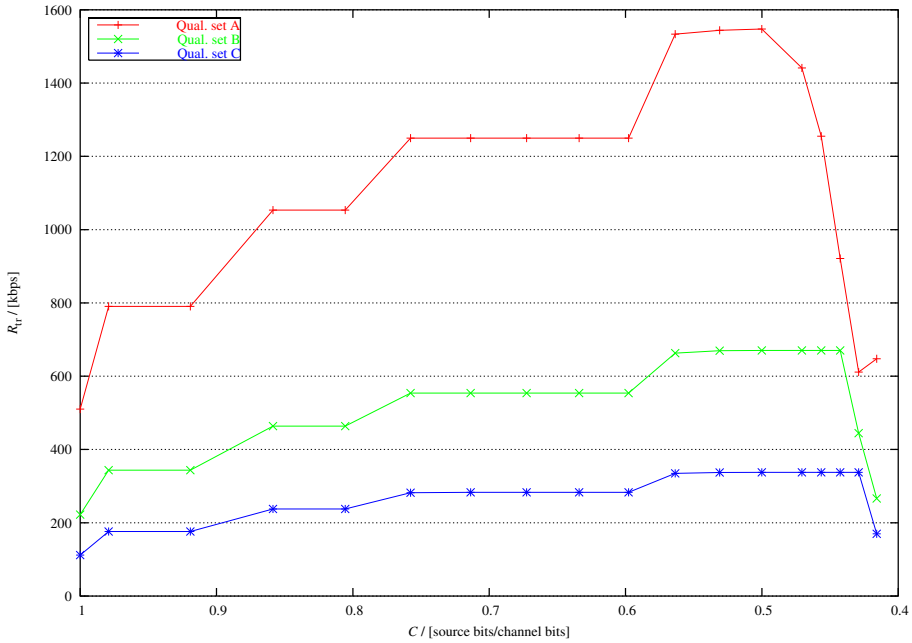


Figure 3.29 — GOP transmission rate as a function of the channel capacity with QCIF-size `Foreman`

conditions. On a general-purpose Intel Pentium II processing unit with 400 MHz, a non-optimized C implementation of the algorithm with the aforementioned parameters lasted on the average, i.e. averaged over all channel conditions and all GOPs, less then 1 s with QCIF-size `Foreman` coded at quality set B, with the quality set C even less than 0.3 s. Thus, if an optimized algorithm is implemented on specialized ASICs or DSP chips, the algorithm's latency is expected to be below 1 s with a substantial margin.

Concerning the VA's complexity, the new form of the rate constraint, Eq. 3.54, leads to a considerable reduction in complexity in contrast to the constraint given by Eq. 3.55, as the total number of trellis nodes is found to be on the average 73% smaller than what is suggested by the complexity bound $O(||\mathcal{C}|| \cdot N_{\mathrm{CP}}^2)$. This number includes all test sequences and all parameters as defined above. For example, instead of the theoretic maximum of 26,912 nodes with `Foreman` (QCIF) and quality set C, the trellis consists of only 7,132 nodes on the average.

It is finally worth mentioning that the channel packet length as employed in [Ban02] might be acceptable for embedded code streams, but

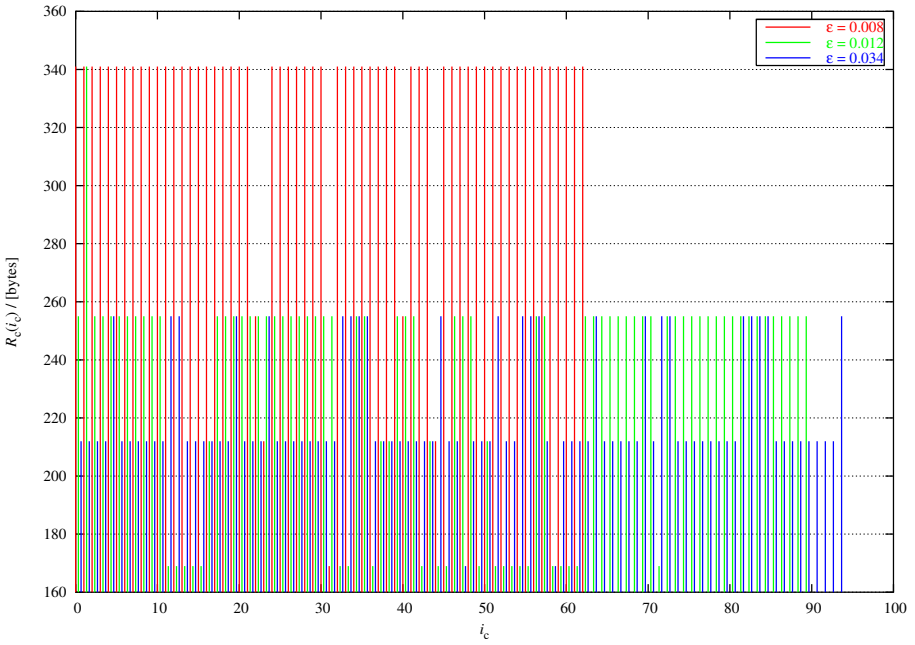


Figure 3.30 — Channel code distribution for the fourth GOP of QCIF-size `Silent`, coded with quality set B, as a function of the channel packet index under different channel conditions, i.e. with ϵ as a parameter. The total number of channel packets necessary to transmit this particular GOP is equal to the maximum index value. The graph is continued in Fig. 3.31

for hybrid streams coded at high rates it is advisable to choose larger packet sizes to avoid very large values of N_{CP}, such that the complexity of the VA be kept low. E.g., the image sequence `Mobile&Calendar` coded with quality set C was encoded with a delay and a complexity which are both far from real-time.

3.11.2 Number of quality layers

In this section, the number N_{l} of quality layers is altered.

The developed channel rate allocation scheme has so far been proved as quite robust with regard to various source signals, signal sizes, and rates. It is hence expected that a change of parameters, e.g. $\Delta QP \in \{12, 16\}$ or $N_{\mathrm{f}} \in \{15, 7.5\}$ (which corresponds to a skip of 1 and 3, respectively), does not change the channel coding performance substantially.

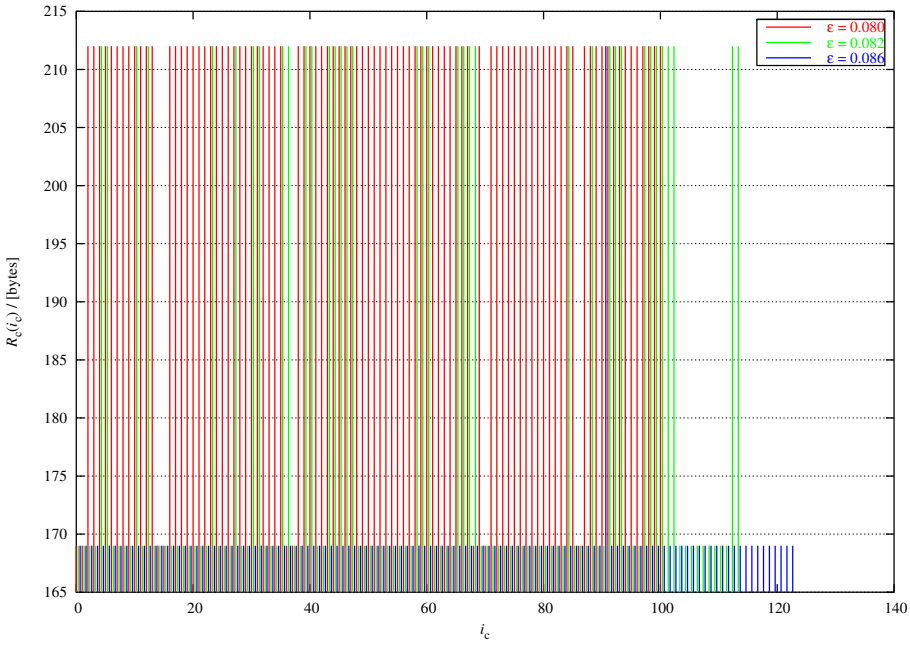


Figure 3.31 — Channel code distribution for the fourth GOP of QCIF-size `Silent`, coded with quality set B, as a function of the channel packet index under different channel conditions, i.e. with ϵ as a parameter. The total number of channel packets necessary to transmit this particular GOP is equal to the maximum index value. The graph is the continuation of Fig. 3.30

When essential parameters like e.g. N_l are changed, however, the system may behave differently. In the sequel, the set of test videos is reduced to the CIF-size `Container` and the QCIF-size `Foreman` to reduce the computational effort. First, N_l is set equal to two, i.e. there is one base layer and one high-quality layer. The quality sets A, B, and C correspond now to the QP sets $\{32, 24\}$, $\{40, 32\}$, and $\{48, 40\}$.

The $PSNR$ curves as a function of the channel capacity show basically the same behavior as in Fig. 3.21 and Fig. 3.22, and are therefore not rendered once more. Consequently, the same applies to the progression of the no-error probabilities of each layer over C. It is, however, interesting to note that the VA succeeds in keeping the P_ne values closer to one, such that the cliff region around $C = 0.45$ becomes steeper. The same applies to the GOP $PSNR$s whose quantities otherwise are more or less identical for all layer schemes, i.e. with one, two, and three layers, provided that

the QPs of the highest-quality layers are equal.

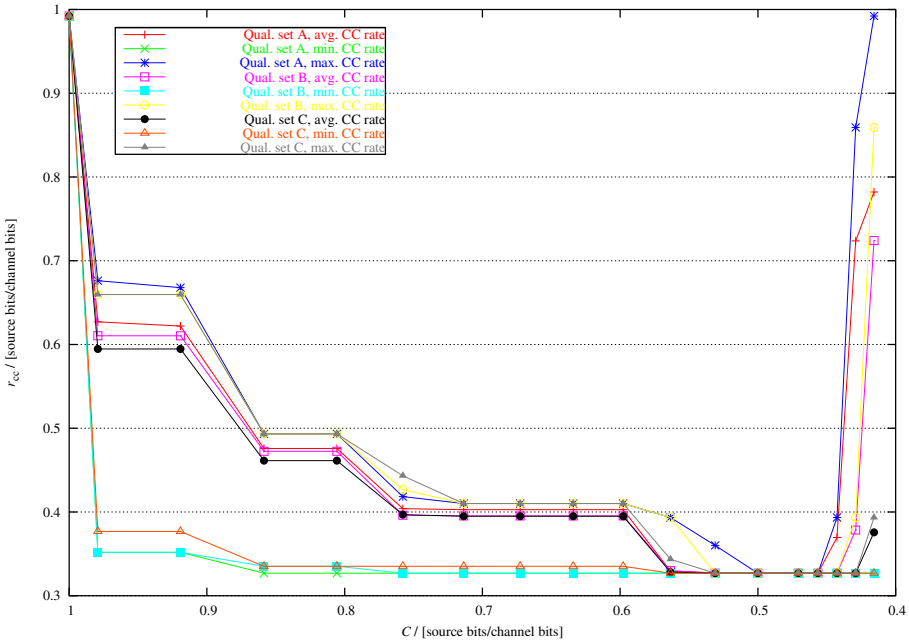


Figure 3.32 — Overall channel code rate per GOP as a function of the channel capacity with QCIF-size `Foreman` and two layers

The transmission rate curves show a monotonic rising with a deteriorating channel quality analog to Fig. 3.28, and the locations of stair-case steps over C are identical. As reported in Sec. 2.5, fewer layers mean less bit rate increase relative to a single-layer scheme, and thus the absolute values of R_{tr} with one layer only are of course lower than those of a double-layer or triple-layer scheme.

The progression of the curve of the double-layer scheme's average channel code rate r_{CC} looks similar to that of the triple-layer scheme, as Fig. 3.32 illustrates. It is stressed that the quality sets of both schemes do not coincide; instead, the sets B and C of the triple-layer correspond to the sets A and B of the double-layer scheme. An inspection of the curve's behavior shows roughly an identical appearance, and there is no clear general rule to determine the grade of protection of a certain packet. However, in this particular example, the code rate variance is larger than shown previously, i.e. a larger range of codes is allocated to the channel packets.

A different allocation behavior shows the single-layer scheme, see

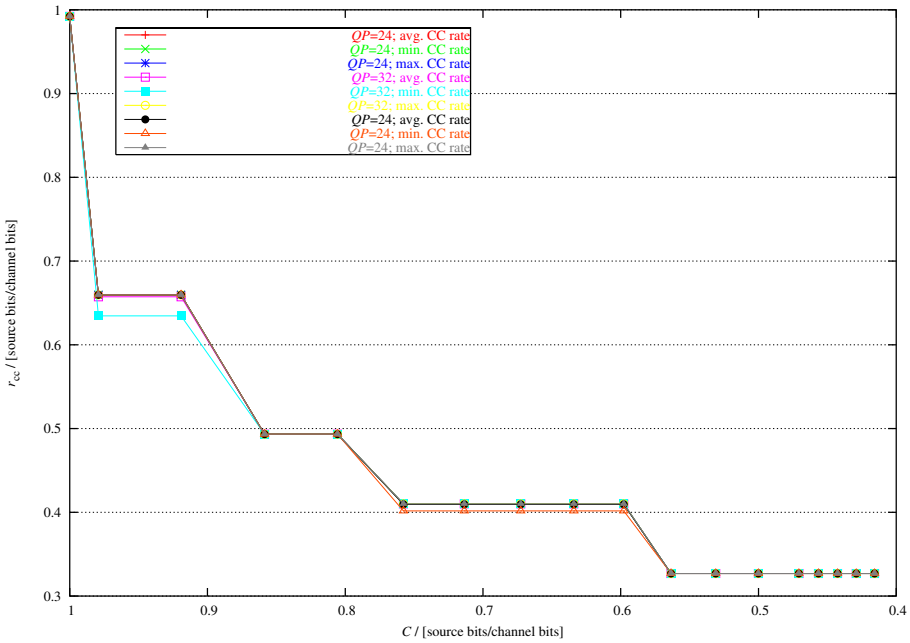


Figure 3.33 — Overall channel code rate per GOP as a function of the channel capacity with QCIF-size `Foreman` and one layer

Fig. 3.33. There, the channel rate allocation algorithm simply assigns all channel packets the highest-rate code which can nearly guaranty error-free transmission, and the code variance is practically zero. It is concluded that mainly layering and an accompanying layer concealment — which must be known to the encoder — lead to unequally protected channel packets, and mean and temporal concealment mechanism do not have a major impact on the rate allocation. Concerning the chosen code, the point specified by ϵ on the code's $PLR(\epsilon)$ curve is either on the error floor or in the beginning of the waterfall region (ϵ approaching from zero). With the proposed packetization scheme, a single layer does not require the application of a sophisticated tool for channel rate allocation.

3.11.2.1 Comparison to other research

An approach similar to the one of this work is presented in [ZBPK03], where the transmission of a double-layered scalable hybrid code stream generated by an H.263v2-compliant codec [ITU98] is optimized with regard to source and channel coding. The competing scheme, subsequently

called CSC1 in contrast to the new scheme (NSC), utilizes Reed-Salomon (RS) (block) codes[10]as channel codes. Unfortunately, the comparison is not straight forward due to the fact that the distortion minimization is subject to a rate constraint.

As a consequence, both schemes are compared to each other only at two points of the distortion capacity spectrum. With the QCIF-size `Foreman` video coded at 30 fps and two different fidelities, the two layer QPs of NSC are determined such that the rate of both layers is below the constraint given by $R_{\mathrm{tr}} r_{\mathrm{CC}}$, where r_{CC} is estimated from Fig. 3.26. The channel rate allocation is then applied to the source stream, and transmission rate and the expected $PSNR$ are recorded for all GOPs of the image sequence, see Tab. 3.5. In the error-free case, $C = 1$, the distortion difference is roughly 4 dB, which shows mainly the source coding efficiency of the new scheme, as its average code rate is equal to 0.99, i.e. the 12/12-rate code is exclusively chosen. The available rate of 360 kbps is equally distributed to both layers, $R_{\mathrm{src}}^{(\mathrm{BL})} = 176$ kbps and $R_{\mathrm{src}}^{(\mathrm{HQL})} = 177$ kbps, with $QP^{(\mathrm{BL})} = 29$ and $QP^{(\mathrm{HQL})} = 25$.

		CSC1		NSC		
C	R_{ch}	R_{tr}	$PSNR$	R_{tr}	$PSNR$	$\Delta PSNR$
1.0	360	360	34.8	355	**39.04**	4.24
0.6	216	360	28.3	330	**33.51**	5.21

Table 3.5 — A comparison of JSCC schemes. The unit of C is *src. bits/ch. bits*, R_{ch} and R_{tr} are given in *kbps*, and $PSNR$ represents the average GOP expectation measured in *dB*

In case of poor channel conditions, here represented by $C = 0.6$, it would not make sense to have the source rate equally distributed any longer to ensure that at least the base layer can be decoded. Hence, it is chosen to set $QP^{(\mathrm{BL})} = 48$ and $QP^{(\mathrm{HQL})} = 33$, with results in $R_{\mathrm{src}}^{(\mathrm{BL})} = 48.41$ kbps and $R_{\mathrm{src}}^{(\mathrm{HQL})} = 88.66$ kbps. Again, channel encoding is applied, which renders a quality superior to CSC1; the difference in $PSNR$ is now 5.21 dB. It can be concluded that the degradation of NSC is not as strong as the degradation of its competitor. This can partly be attributed to the protection capabilities of the channel codes employed, and partly to the rate allocation algorithm. The NSC has further the advantage of

[10]Unfortunately, the value of the block size is not mentioned; hence it is impossible to compare NSC and CSC1 with regard to how close they actually approach the channel capacity.

a successive source consumption, which in turn means less latency, less buffering requirements, and a more efficient channel coding, as padding is as far as possible reduced.

3.11.3 Equal error protection

In this section, the proposed UEP scheme is compared to a conservative code rate allocation, i.e. equal error protection (EEP). The motivation is the small code rate variance as reported in Sec. 3.11.1.

As the previous results have been shown to be consistent for all test videos, the number of simulations can be limited to only one image sequence, QCIF-size `Foreman`. The frame rate is 10 fps, the number of slices per frame is equal to one, and the number of layers is set to two. The layer QPs are 40 and 32. Two channel code sets are used; one consisting of eight codes as in the previous sections, called UEP code set, and one named EEP set, which is compound of only a single code, the 5/12-rate code. The performance of the channel encoder in terms of rate and distortion is analyzed for the same channel conditions as before, including a channel mismatch of +10%. The results are plotted in Fig. 3.34.

As there is only one channel code in the EEP case, the rate is constant over all channel conditions. With $\epsilon < 0.04$, the EEP scheme (ESC) scheme over-protects the code stream; that is, $R_{\text{tr}}^{(\text{EEP})} > R_{\text{tr}}^{(\text{UEP})}$, and because of the then increased accumulation of distortions in Eq. 3.48, the average GOP $PSNR$ is not optimum either, since $PSNR^{(\text{EEP})} < PSNR^{(\text{UEP})}$.

With $\epsilon \in [0.04, 0.08]$, the distortion achieved by the UEP scheme (USC) is slightly less than that by ESC. This is as expected, as the former system was designed for minimum distortion. However, a consequence of the fidelity constraint is that this comes at the price of a somewhat increased transmission rate, $R_{\text{tr}}^{(\text{EEP})} < R_{\text{tr}}^{(\text{UEP})}$. Within the given interval, the 5/12-rate code provides optimum EEP.

With $\epsilon > 0.08$, the rate of USC rises further, whereas $R_{\text{tr}}^{(\text{EEP})}$ stays constant; that is, ESC now under-protects the code stream, and as a result, $PSNR^{(\text{EEP})}$ drops much earlier than $PSNR^{(\text{UEP})}$ as ϵ goes up. A look at the mismatch curves reveals a similar relationship.

It is concluded that the minimally small gap in distortion between a very sophisticated tool like USC and a conservative system like ESC is not worth the optimization effort if the channel conditions can be estimated with high accuracy, both from a complexity and a distortion rate point of

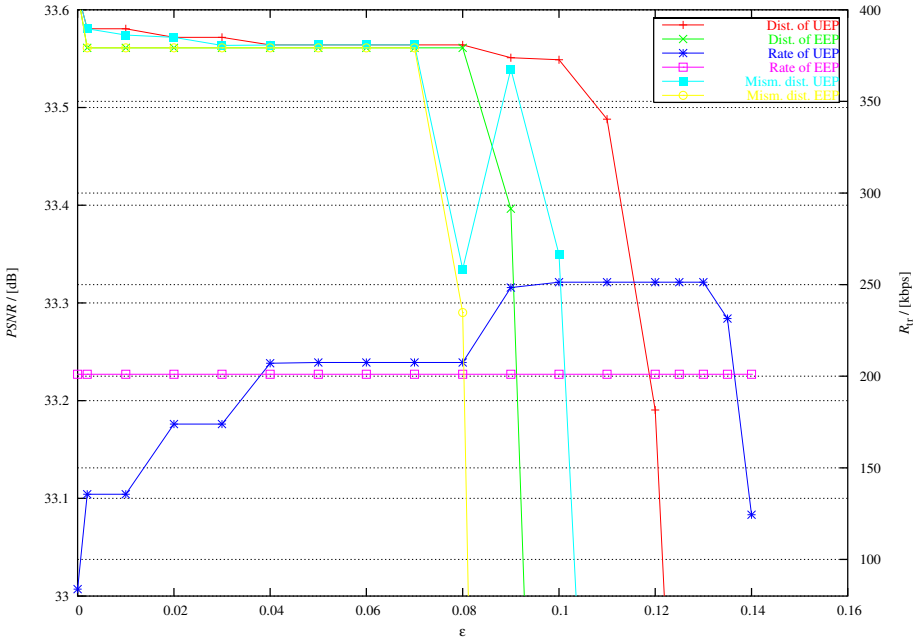


Figure 3.34 — Average GOP distortion and transmission rate over the bit error rate ϵ on the channel

view. This is, however, seldom the case. In fact, severe channel mismatch occurs frequently due to the non-stationary properties of most wireless channels and because of insufficient knowledge about the channel state. The real strength of the proposed scheme is the robustness of the code stream in mismatch situations. An additional 'bonus' is the monotonic improvement in $PSNR$ of USC over ESC as ϵ approaches zero. Finally, it should be mentioned that the observation of the small difference between the minimum distortion of UEP and that of EEP (in case the channel state is known to the encoder) can be confirmed by a similar result — somewhat surprisingly — with an embedded bit stream in [FCP04].

3.11.4 Simulation of channel code rate allocation

In this section, the formulation of the expected GOP distortion in Eq. 3.48 and the functionality of the Viterbi algorithm are verified by a simulation.

300 Frames of the QCIF-size `Foreman` video, encoded in source coding mode IPP at 10 fps, with a skip of two, two quality layers, and the corresponding QPs 40 and 32 are channel-encoded by the rate allocation

algorithm as devised above. The distribution of channel codes is passed as input to a simulation program which produces a pseudo-random bit error pattern for each GOP. Then, the probabilities of residual bit errors in the payload data of each channel packet are determined, and if an error is found, the source packets transmitted error-free are source-decoded, and the data of the remaining packets are concealed. Finally, the GOP distortion is calculated.

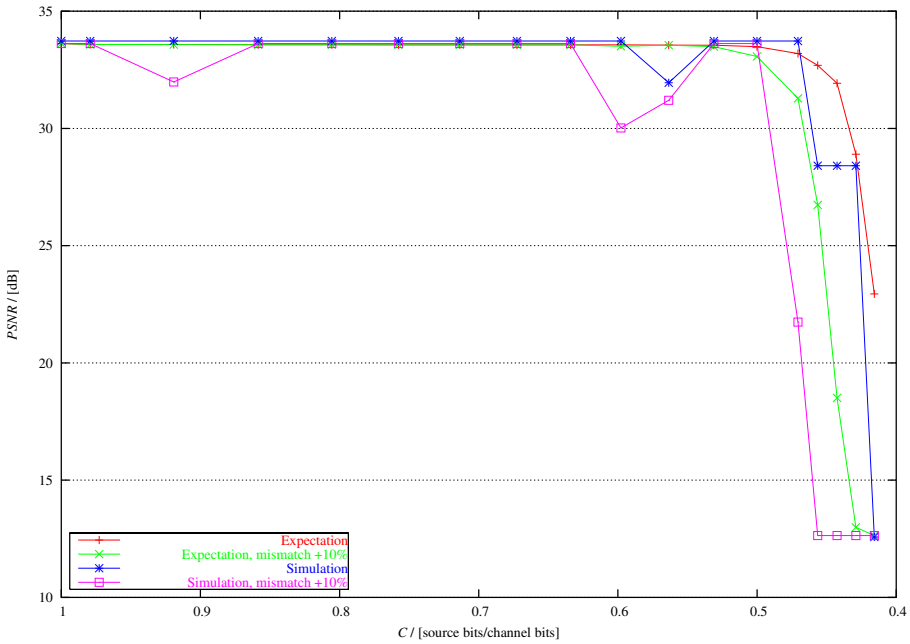


Figure 3.35 — Average GOP $PSNR$ (in dB) over the channel capacity. The distortion is in one case computed by means of the expectation and in the other case determined by means of a simulation

The results in Fig. 3.35 show the decoder image distortion in dB, averaged for 10 GOPs and over various error patterns with different seeds. Also the $PSNR$ values for a channel mismatch of $+10\%$ are given.

Even though the curve of the expectation is much smoother than the curve determined by the simulation, the main behavior depending on the channel capacity is the same. The deviation of the transmission realization from the theoretical performance is rather small, except for capacities less than roughly 0.5. To smooth the curve of simulation $PSNR$s at those high channel bit error rates, the weak law of large numbers suggests to increase the number of seeds. However, to avoid exhaustively long sim-

ulation times, the number of seeds here is limited to 50. The simulation results also reflect the position of the $PSNR$ 'outliers' as determined e.g. in Fig. 3.21, namely $C \simeq 0.9$ and $C \simeq 0.6$.

Chapter 4

Summary, conclusions, and outlook

This summary concludes the second part, i.e. Chap. 3, of the dissertation. For a summary of topics and results covered in the first part, Chap. 2, it is referred to Sec. 2.6.

A brief review of previous research in the field of robust transmission of embedded and hybrid bit streams was given. The error propagation in a hybrid code stream was then exemplified by means of the video coding standard H.264, and the sensitivity to errors of the three coding techniques run length, variable-length, and predictive coding were examined, hereby giving the rationale for the application of layered coding and sophisticated error protection mechanisms, as all presented coding schemes were found to be very vulnerable to channel disturbances.

Error spreading considerations led to the definition of spatial and temporal structures in the image sequence, namely layer slices and GOPs, which efficiently limit spatio-temporal error propagation. Following this definition, it was proposed to split the code streams into segments, each of which depends on a certain set of previously encoded segments, hereby establishing clear inter-segment dependencies.

A packetization scheme based on successive segment / source packet (SP) consumption was developed, requiring the specification of two mappings, the mapping of source segment indices to SP indices and vice versa, and the mapping of SP indices to channel packet (CP) indices and vice versa.

Considering packet-switched wireless network transmission, the for-

mulation of the distortion of a potential CP loss was established as a combination of distortions of source segments which are transported with that particular CP. Hereby, three distortion types have been accounted for: quantization, channel, and concealment distortion. The quantization distortion of a segment of a certain frame varies according to which quality layer the slice belongs to. The distortion of channel and concealment errors was jointly defined by proposing a terminate-on-error decoding strategy and the deployment of error concealment to bound error propagation within one GOP and visually minimize the error impact, respectively. Three different concealment methods were considered: mean, layer, and temporal concealment. Additionally, the formula of a joint-error distortion was derived for certain cases of segment constellations in a particular CP.

A channel code rate allocation scheme was proposed which minimizes the expected GOP distortion subject to a decoder quality constraint. A term for the expected distortion was found based on CP distortions, probabilities for packet losses, and inter-packet dependencies which rely in turn on inter-slice dependencies. The relationship of the quality constraint to a source rate constraint and the channel conditions was discussed in detail. Both encoder and decoder structures of this asymmetric channel coding scheme were developed.

Finally, the Viterbi algorithm (VA) was presented as a low-complexity solution to the channel code optimization problem, and the performance results of its software implementation were given. The functionality of the VA and the properness of the expectation's use were further verified by a software simulation.

4.1 Summary of results and conclusions

As the target application is a wireless ATM channel, the transmission over a BSC was considered, and punctured parallel concatenated recursive convolutional channel codes were employed to protect the payload data of the CPs against random bit errors.

A set of test videos was encoded at different rates and passed to the channel rate allocation algorithm, the performance results of which are consistent for all visual material of different sizes and source rates. The system aims successfully at the maximization of the overall channel code rate subject to a fidelity constraint. It was found that the code distribution achieving minimum distortion leads to a stair-case function

of channel code rates over the channel capacity. The variance of CP code rates was shown to be quite small, i.e. the VA prefers to allot only two to three out from the set of eight available channel codes to the CPs.

The distribution of channel codes varies strongly with the channel conditions, hereby leading to unequal error protection, even though this is not explicitly formulated as a requirement. No direct relationship could be determined between the strength of the channel code assigned to a particular packet and the layer or frame affiliation of source segments transported by that CP. The overall strength depends strongly on the assumed channel conditions; the higher the probability for residual errors in the bit stream after decoding, the stronger the assigned channel codes. The code allocation scheme works well until the protection capability of the strongest code is exceeded; then, the algorithm switches to the strategy to convey as much source data as possible before the first error is encountered.

Simultaneously, it was observed that the VA achieves an almost constant image quality up to the cut-off channel error rate, and further depending on the source rate. The determined minimum-distortion code distribution was also shown to give the code stream the property of graceful performance degradation in case of channel mismatch of 5% and 15%. It could be observed that the $PSNR$ curves coincide with the curves of the no-error probabilities of each layer, which show a monotonic decrease in value.

Due to the constant-quality approach, the system was found to increase the transmission rate considerably for a deteriorating channel quality, hereby producing a variable-rate output. As a possible extension of the novel system, it was hence discussed to include a feed-back loop for rate control purposes if desired.

The complexity of the VA has been examined. It is noted that the optimality of the solution as proposed is valid only with regard to the VA which, on the other side, achieved always the global optimum in a small number of optimality comparisons. The presented implementation with a source rate constraint was observed to yield complexity reductions relative to the theoretically possible numbers of on the average 73%. Yet, the latency of the presented solution was discussed, and it was shown that, for sufficiently large channel packets (with respect to the average source rate), a real-time computation of the rate allocation can be accomplished.

Finally, it was investigated in how far a different number of quality layers influences the allocation process. With two layers, the variance

of the GOP code rate is larger than in the three-layer case mentioned above, whereas the other performance-describing parameters are similar to those reported with three layers. Furthermore, the double-layer system performs superior to other JSCC schemes, the improvement being about 4–5 dB. With one layer, it was found that the channel coding scheme achieves no improvement over plain channel code allocation simply based on the statistics of channel codes and the estimated channel error rate. A comparison to EEP revealed a small difference between the minimum distortion of UEP and that of EEP in case the channel state is known with high probability, and the main property of UEP is identified to be the code stream's robustness in mismatch situations.

It is concluded that the joint source channel coding scheme presented here succeeds in providing a robust code stream suitable for transmission over (unreliable) wireless packet-switched networks. It is engineered to have knowledge of a rough estimate of the channel conditions, although the performance degradation in mismatch situations is very small. Moreover, no significant loss in performance has to be accepted in the error-free case.

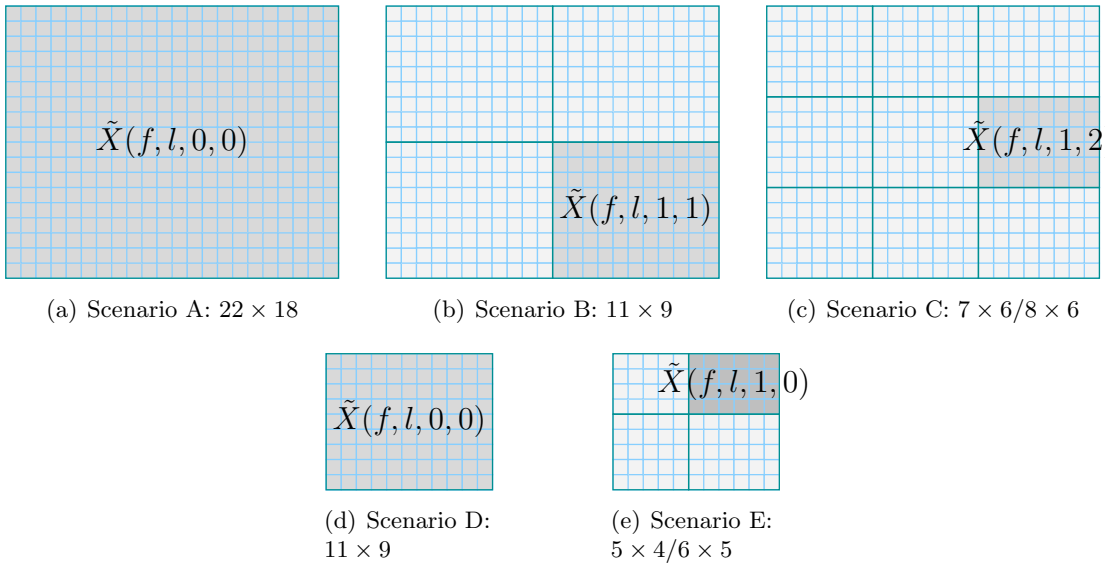
To the author's knowledge, this is the first joint source channel coding approach with pure VBR video and a decoder fidelity constraint.

4.2 Recommendations for future work

Untouched areas which may be investigated in future research are the employment of other channel codes with different fixed or varying packet lengths, and the system performance in case of instationary, i.e. slowly fading, channels. This could easily be included in the rate allocation approach as it is not mandatory for ϵ to be constant over the entire GOP. Instead, the channel is assumed to be stationary during the transmission of one channel packet of length R_{CP}.

Especially the usage of convolutional codes seems promising.

Unfortunately, the flexible MB ordering functionality in H.264's reference software — albeit readily contained in the specifications — was not implemented until recently, such that no experiments could be carried out to investigate the influence of parameters like slice geometry as defined in Sec. 3.3. However, it is stressed that the channel rate allocation algorithm as presented is capable of dealing with various slice sizes. By altering the number of slices to e.g. four and nine as shown in Fig. 4.1, it is expected to observe a further trade-off between coding efficiency and error prop-



(a) Scenario A: 22×18 (b) Scenario B: 11×9 (c) Scenario C: $7 \times 6/8 \times 6$

(d) Scenario D: 11×9 (e) Scenario E: $5 \times 4/6 \times 5$

Figure 4.1 — Different slice sizes, measured in MB units ($x \times y$). Scenarios A through C are applied to CIF-size videos, whereas scenarios D and E are used with QCIF-size image sequences

agation limitation, and the GOP $PSNR$ is likely to show an even more graceful degradation in mismatch situations than the presented scheme.

It seems that it is advantageous for the packetization scheme not to allow packets associated with a joint distortion error. The author believes that such a packetization scheme would also lead to unequally protected single-layer code streams. An additional benefit can be expected by data partitioning as specified in the Extended profile in H.264. The data of one slice are split into three partitions of data with different importance and sensitivity to errors: a header, an INTER, and an INTRA partition, each of which can be conveyed in a separate channel packet.

A listing of other interesting topics includes

- the extension of the proposed technique to account for the inhomogeneity of networks by additionally considering packet erasures,
- a decoder implementing more sophisticated error concealment algorithms than those utilized in this thesis,
- the generalization of SNR layer concealment to frameworks with spatial and temporal scalability, and

- the determination of a GOP's maximum number of frames, provided N_l is fixed, depending on the channel conditions, and requiring a rate constraint.

Appendices

Appendix A

Original videos

Six different natural progressive-scan image sequences of size QCIF or CIF are used throughout this work. All have a $YCbCr$ color space, where the chrominance components are subsampled according to 4:2:0. The videos are recorded with 30 fps and have hence — with 300 frames/pictures — a duration of 10 s. A pixel/sample is represented by 8 bits. The QCIF-size videos have been processed from CIF size by low-pass filtering and subsampling. QCIF denotes an image size of ($width \times height$) 176 × 144 pixels, and a CIF frame has the dimensions 352 × 288 pixels.

A non-scaled colored[1] representative frame of each sequence is shown subsequently, and a short description of each video follows.

- `Container` is recorded with a fixed camera. It contains little, translatoric motion and a small amount of spatial details.
- `Foreman` is a head-and-shoulder sequence with a static background and little spatial detailedness. The camera is mobile and fulfills a right turn near the end of the sequence. The amount of object motion is moderate.
- `Mobile&Calendar` is a highly detailed sequence with complex motion. While the camera turns left, it zooms out of the picture containing many moving objects.
- Another head-and-shoulder video is `Mother&Daughter`, recorded with a fixed camera. The foreground is moderately detailed, the background is quite plain. The amount of object motion is small.

[1] In the printed book, they may be displayed monochromatically.

- `Silent` shows head and shoulders of a moderately moving object in front of a still background of high detail. The camera is fixed.
- Also `Akiyo` is a head-and-shoulders sequence. It is of low detail, as well as low object and low camera motion. `Akiyo` is only involved in performance comparisons with other research.

Name	Size	Cross-reference
`Container`	CIF	Fig. A.1
`Foreman`	CIF	Fig. A.2
`Mobile&Calendar`	CIF	Fig. A.3
`Mother&Daughter`	CIF	Fig. A.4
`Foreman`	QCIF	Fig. A.5(a)
`Silent`	QCIF	Fig. A.5(b)
`Akiyo`	QCIF	Fig. A.5(c)

Table A.1 — Overview of image sequences

Figure A.1 — Frame 1 of CIF-size image sequence `Container`

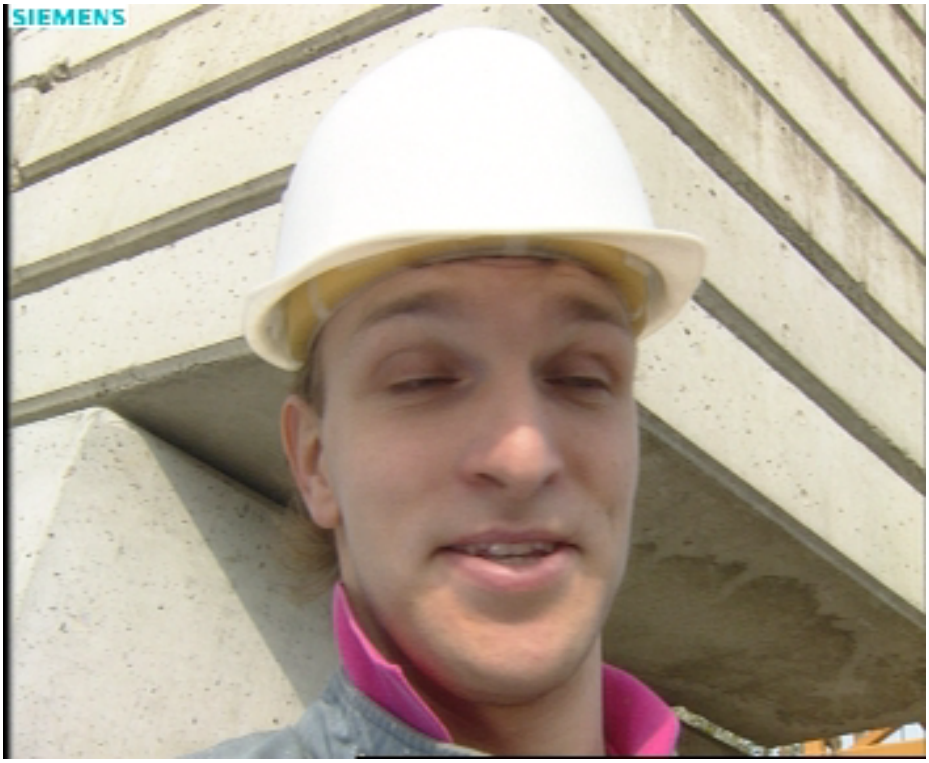
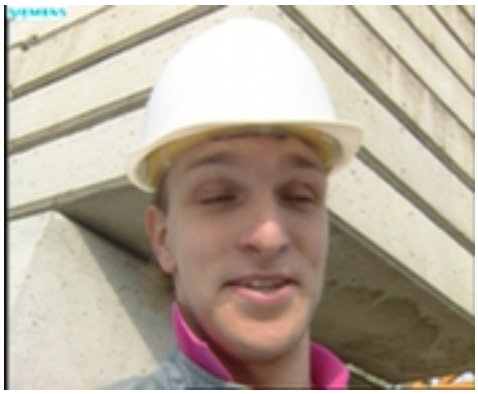


Figure A.2 — Frame 1 of CIF-size image sequence `Foreman`

Figure A.3 — Frame 1 of CIF-size image sequence `Mobile&Calendar`

Figure A.4 — Frame 1 of CIF-size image sequence `Mother&Daughter`



(a) Frame 1 of video `Foreman`

(b) Frame 1 of video `Silent`

(c) Frame 1 of video `Akiyo`

Figure A.5 — QCIF-size image sequences

Appendix B

Human visual perception and relation to communication schemes

Even though humans are highly visual creatures, the complexity of the human visual system (HVS) has been and is still the main barrier for a rapid progression of science in gaining a deeper understanding about many of its properties. This chapter discusses briefly in how far the human perception should be accounted for in visual communication systems. For an in-depth discussion of the properties of the visual system, see [Win00].

Human visual perception depends on two major components. First, there are the physics and the chemistry of the human eye. The other component is the center for image processing in the human brain. Both components together form the *HVS* which is hereby the *sink*, i.e. final recipient, of visual information in most communication system. For image compression, the properties of the HVS are of high interest as they can be exploited to achieve great compression factors, which is crucial in most applications. Nowadays, there are still numerous unanswered questions with regard to the proper modeling of the HVS. Some of the visual systems's properties are, however, known.

First of all, the property exploited most often, for instance in early analog visual compression standards like PAL, SECAM, and NTSC, is the decomposition of the visual signal into *channels* (e.g. the RGB color space). This, combined with the knowledge about the low chromatic

acuity of the HVS and its non-linear perception of lightness, is the motivation for transforming the signal to a space of color differences, for example YUV or $YCbCr$, and a subsampling of the chrominance signals before coding.

Next, *interlaced* coding is originally an analog compression technique which was introduced to efficiently compress signals of high temporal frequencies (50 and 60 Hz). The high frequencies were used to reduce flickering encountered often with analog television. Interlacing trades off vertical image resolution with temporal resolution, hereby reducing the bandwidth required for representing the signal by a factor of two. The exploited property of the HVS is the significant decrease of spatial resolution of the *contrast sensitivity function* CSF at such high frequencies. The CSF attempts to describe the visual system's sensitivity to both achromatic and chromatic contrasts.

The fact that quantization noise in image regions of high activity, i.e. sample variance, can be easier hidden than in homogeneous regions is due to spatial *masking*. A similar property of the HVS exists for temporal masking, which allows a coarse quantization e.g. in the frame directly after a scene cut in a video. A course quantization enables in turn bandwidth savings.

Finally, the model of local pattern adaptation of the HVS is generally accepted. It says that the visual system's frequency sensitivity is reduced for spatial signals of equal frequency, i.e. locally repeating textures. These structures can therefore be quantized stronger than other regions.

A properly designed coding scheme should reflect the influence of the HVS in the measurement of image quality. However, apart from content or image material, relevant factors that influence the opinion of the human viewer with regard to the image quality are viewing distance, display size, resolution, brightness, contrast, sharpness, colorfulness, naturalness, and other parameters. With other words, the *mean opinion score*, MOS, is, though expensive and time-consuming, the ultimate quality measure. Various quality metrics have been proposed in the literature to estimate the MOS, but unfortunately, most of those metrics render mathematically intractable optimization problems. Thus, this work resorts to use the well known sample-based error metrics listed in App. C.

Appendix C

Image quality assessment

As discussed in App. B, the MOS is the ultimate visual quality assessment measure. However, such measurements are, if sometimes not impossible to apply, very costly processes. Instead, the quality of image compression algorithms is usually measured arithmetically by the application of error metrics. Objective metrics are used due to their minor computational complexity and easy inclusion into compression algorithms, and subjective metrics aim at approximating the perceptual properties of the HVS. Subjective metrics are not applied in this work due to their relatively high computational complexity.

C.1 Objective error metrics

All subsequent definitions assume ergodic stochastic processes and 2-D finite-length random variables.

C.1.1 Sum of squared differences

Based on two 2-D signals denoted as X and Y, both of size $N \times M$ samples, the sum of squared differences of two signals is defined as

$$SSD(X,Y) = \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} |X(i,j) - Y(i,j)|^2 \,, \tag{C.1}$$

where $X(i,j)$ represents a single signal sample.

C.1.2 Mean squared error

The most commonly used error criterion in evaluating the difference between two signals is the MSE. Denoting the 2-D signals as X and Y, both of size $N \times M$ samples, the MSE equals the normalized SSD of these signals:

$$MSE(X,Y) = \frac{1}{NM} SSD(X,Y)\,, \tag{C.2}$$

with SSD as defined in Eq. C.1. Thus, the MSE corresponds to the power of the difference signal of X and Y.

C.1.3 Peak-signal-to-noise ratio

The related measure of $PSNR$ in dB is computed using

$$PSNR(X,Y) = 10 \log_{10} \left(\frac{X_{\max}^2}{MSE(X,Y)} \right). \tag{C.3}$$

The maximum value $X_{\max}$ that the signals X and Y can take on is 255 for a pixel representation of 8 bpp.

It should be kept in mind that the absolute value of $PSNR$ may vary significantly for decoded images having the same perceived visual quality. This means that a $PSNR$ of 30 dB may indicate a high visual quality for one image, whereas for another one with the same $PSNR$, the subjective evaluation may disclose a poor codec performance.

C.1.4 Signal variance

The true variance σ_X^2 of a signal X, the mean-removed expectation $E\{X - \bar{m}_X\}$, is in this work approximated by biased normalization:

$$\sigma_X^2 = \frac{1}{NM} \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} (X(i,j) - \bar{m}_X)^2\,. \tag{C.4}$$

The true mean $\bar{m}_X$ of a signal X, the first-order moment $E\{X\}$, is approximated by $\bar{m}_X = \frac{1}{NM} \sum_i \sum_j X(i,j)$. With $Y(i,j) = \bar{m}_X$ for all i and j, the MSE in (C.2) becomes identical with (C.4).

C.1.5 Rate distortion product

The rate distortion product RDP is defined as the normalized sum over all products of rates and distortions achieved by an encoding process with

N_{par} different parameters like quantizer identifier and rate constraint,

$$RDP = \frac{1}{N_{\text{par}}} \sum_{i=1}^{N_{\text{par}}} MSE(i)R(i)\,, \tag{C.5}$$

and characterizes rate distortion curves in the unit *bits*, i.e. it is a means of measure of the relationship between rate and corresponding distortion. A high score indicates a poor coding or the fact that a sequence is difficult to code due to its statistics.

C.1.6 Rate savings

Two compression algorithms can be compared to each other in terms of rate savings, while achieving the same visual quality. The algorithm with the poorest performance, say algorithm A, is used as the common base, providing the anchor rate R_{A}, while the other algorithm (B) achieves rate R_{B} ($R_{\text{B}} < R_{\text{A}}$). The rate savings (in %) are then expressed as

$$S(PSNR) = 100 \cdot \frac{R_{\text{A}}(PSNR) - R_{\text{B}}(PSNR)}{R_{\text{A}}}\,, \tag{C.6}$$

where the $PSNR$, which must be equal to be able to compare both algorithms, is chosen to measure the visual quality.

References

The number following each reference entry indicates the number of the page where the citation is made.

[ADR96] S. B. Zahir Azami, P. Duhamel, and O. Rioul. *Combined source-channel coding: Panorama of methods*. In *CNES Workshop on Data Compression*. Nov. 1996 7

[AKS+02] S. Adachi, S. Kato, K. Sugimoto, T. K. Tan, and M. Etoh. *Structured VLC based on Golomb code for CAVLC*. Tech. Rep. D083, ITU-T VCEG | ISO/IEC MPEG (JVT), Jul. 2002
Considers the replacement of unstructured variable-length code tables used so far in H.264 by tables with truncated Golomb codes. 25

[AR99] Ö. Açikel and W. Ryan. *Punctured turbo-codes for BPSK/QPSK channels*. IEEE Trans. Commun., Vol. 47, No. 9, Sep. 1999 86

[Au02] James Au. *Complexity reduction for CAVLC*. Tech. Rep. D034, ITU-T VCEG | ISO/IEC MPEG (JVT), Jul. 2002
Some minor modifications of the entropy coding engine for reduced complexity. 25

[Ban02] B. A. Banister. *Robust image and video transmission across heterogeneous networks with packet erasures and bit errors*. Ph.D. thesis, School of Electrical Engineering and Computer Science, Washington State University, Pullman (WA, USA), May 2002
Considers embedded bit streams. 50, 83, 84, 87, 95, 97

[BB98] M. R. Banham and J. C. Brailean. *Video coding standards: Error resilience and concealment*, Chap. 5, pp. 133–174. Kluwer Academic Publishers, 1998 20

[BBF02] B. A. Banister, B. Belzer, and T. R. Fischer. *Robust image transmission using JPEG2000 and turbo-codes.* IEEE Signal Processing Letters, Vol. 9, No. 4, pp. 117–119, Apr. 2002 60, 86

[Bel57a] Richard Ernest Bellman. *Dynamic Programming.* Princeton University Press, Princeton (NJ, USA), 1957
The first publication about modelling multi-stage decision processes and solving related optimization problems. 81

[Bel57b] Richard Ernest Bellman. *A Markov decision process.* Journal of Mathematics and Mechanics, Vol. 6, No. 5, pp. 679–684, 1957 82

[BHM00] G. Blättermann, G. Heising, and D. Marpe. *A quality scalable mode for H.26L.* Tech. Rep. J24, ITU-T SG16 Q.15, May 2000
A variant of fine-granular scalability for H.26L. 23

[Bjø02] Gisle Bjøntegård. *Improved low complexity entropy coding for transform coefficients.* Tech. Rep. B045, ITU-T VCEG | ISO/IEC MPEG (JVT), Jan. 2002
The first proposal of a series of technical reports concerning CAVLC (improved variable-length coding) for H.26L. 25

[BL02] Gisle Bjøntegård and Karl Lillevold. *Context-adaptive VLC (CVLC) coding of coefficients.* Tech. Rep. C028, ITU-T VCEG | ISO/IEC MPEG (JVT), May 2002
This document specifies in detail the new variable-length coding method for H.264. It is part of the series of contributions JVT-B045, JVT-C028, JVT-D034, JVT-D036, and JVT-D083, which all deal with the same topic. 25

[CF99] V. Chande and N. Farvardin. *Joint source-channel coding for progressive transmission of embedded source coders.* In *Proc. Data Compression Conference (DCC)*, pp. 52–61. Snowbird (UT, USA), Mar. 1999 50

[CGG90] G. Castagnoli, J. Ganz, and P. Graber. *Optimum cyclic redundancy-check codes with 16-bit redundancy.* IEEE Trans. Commun., Vol. 38, No. 1, pp. 111–114, Jan. 1990 60

[CM03] C. M. Calafate and M. P. Malumbres. *Testing the H.264 error-resilience on wireless ad-hoc networks.* In *Proc.*

EURASIP Video/Image Processing and Multimedia Communications Conference (VIPMCC), pp. 789–796. 2003 57

[CZZC00] J. Cai, Qian Zhang, Wenwu Zhu, and C. W. Chen. *An FEC-based error control scheme for wireless MPEG-4 video transmission*. In *Proc. IEEE Wireless Comm. and Networking Conf.*, Vol. 3, pp. 1243–1247. Sep. 2000 51

[DV86] M. Duponcheel and W. Verbiest. *Simulation results for a hybrid transform video coding algorithm*. Tech. Rep. BTM-A 11-05-PR, RACE project, 1986
Mentions layered video coding for the first time in the literature. 11

[ELP+02] Yiftach Eisenberg, Carlos E. Luna, Thrasyvoulos N. Pappas, Randall Berry, and Aggelos K. Katsaggelos. *Joint source coding and transmission power management for energy efficient wireless video communications*. IEEE Trans. Circuits, Syst. for Video Technol., Vol. 12, No. 6, pp. 411–424, Jun. 2002 51

[Eve63] H. Everett. *Generalized Lagrange multiplier method for solving problems of optimum allocation of resources*. Oper. Res., Vol. 11, pp. 399–417, 1963
The first paper concerning Lagrange optimization. 28

[FCP04] Masoud Farshchian, Sungdae Cho, and William A. Pearlman. *Optimal error protection for real time image and video transmission*. In *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Proc. (ICASSP)*, Vol. 4, pp. 625–628. Montreal (Quebec, Canada), May 2004 104

[For73] G. D. Forney. *The Viterbi algorithm*. Proc. IEEE, Vol. 61, pp. 268–278, Mar. 1973
Introduces the Viterbi algorithm for the first time in the literature. 81

[GF98] Bernd Girod and Niko Färber. *Error-Resilient Standard-Compliant Video Coding*, Chap. 5, pp. 175–197. Kluwer Academic Publishers, 1998 6, 56

[GFH97] B. Girod, N. Färber, and U. Horn. *Scalable codec architectures for internet video-on-demand*. In *Proc. Asilomar Conf. on Signals, Systems, and Computers*. Pacific Grove (CA, USA), Nov. 1997 22

[GK99] Micheal Gallant and Faouzi Kossentini. *Efficient scalable DCT-based video coding at low bit rates.* In *Proc. IEEE Int. Conf. on Image Processing (ICIP)*. Kobe, Japan, Oct. 1999 21, 28, 41, 42

[GK01] Michael Gallant and Faouzi Kossentini. *Rate-distortion optimized layered coding with unequal error protection for robust Internet video.* IEEE Trans. Circuits, Syst. for Video Technol., Vol. 11, No. 3, pp. 357–372, Mar. 2001 51

[GLM+03] Justin Goshi, Richard E. Ladner, Alexander E. Mohr, Eve A. Riskin, , and Alan Lippman. *Unequal loss protection for H.263 compressed video.* In *Proc. Data Compression Conference (DCC)*, pp. 73–82. Snowird (UT, USA), Apr. 2003 52

[Hal01] Till Halbach. *TML-8.4 error resilience performance.* Tech. Rep. N67, ITU-T Q.6/SG 16 (VCEG), Sep. 2001
This paper mainly tests the stability of the reference software. 20

[Hal02a] Till Halbach. *Enhanced variable-length coding.* Tech. Rep. B034r1, ITU-T VCEG | ISO/IEC MPEG (JVT), Jan. 2002
Contains a proposal for a new reversible entropy coding method. 20, 54

[Hal02b] Till Halbach. *Motivation for error-tolerant communication.* In *Proc. SPIE's Visual Communications and Image Processing (VCIP)*, Vol. 4671 (Part Two), pp. 1210–1218. San Jose (CA, USA), Jan. 2002
Discusses and corrects research results of other research, hereby also investigating the trade-offs between transmission delay, packet error rate, and residual bit error rate. 49

[Hal02c] Till Halbach. *Performance comparison: H.26L intra coding vs. JPEG2000.* Tech. Rep. D039, ITU-T VCEG | ISO/IEC MPEG (JVT), Jul. 2002
Reports the results of a standard comparison. 16

[Hal02d] Till Halbach. *Reduced slice headers and bit error resilience.* Tech. Rep. C129, ITU-T VCEG | ISO/IEC MPEG (JVT), May 2002
Reports the results of an experiment concerning data partitioning, resynchronization markers, and error detecting variable-length codes in the H.26L (JM-1.9) codec. 20

[Hal02e] Till Halbach. *Some important international standards and organizations in visual telecommunications, Part I: Organizations in international standardization.* NORSIGnalet, Vol. 2, pp. 9–13, Sep. 2002
A rough overview of organizations in standardization of still-image and video compression. 14

[Hal03a] Till Halbach. *The H.264 video compression standard.* In *Proc. Nordic Signal Processing Symposium (NORSIG).* Bergen (Norway), Oct. 2003
A comprehensive introduction to H.264. All basic features are explained, and a performance evaluation is given. 13, 27

[Hal03b] Till Halbach. *Some important international standards and organizations in visual telecommunications, Part III: Video compression standards.* NORSIGnalet, Vol. 1, pp. 12–19, Apr. 2003
A historical overview of the video compression standards H.120 and H.26x as well as MPEG-x. 14

[Hal04a] Till Halbach. *Optimum unequal error protection of SNR-scalable DPCM-coded video.* In *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Proc. (ICASSP)*, Vol. 4, pp. 629–632. Montreal (Quebec, Canada), May 2004
Similar to [Hal04b], but with different parameters and greatly extended simulation section. 85

[Hal04b] Till Halbach. *Unequal error protection of SNR-scalable DPCM-coded video.* In *Proc. Data Compression Conference (DCC).* Snowbird (UT, USA), Mar. 2004
A variation of [HF03a] with a quality instead of a rate constraint, suitable for variable-rate coding. 85, 133

[HBM+00] G. Heising, G. Blättermann, D. Marpe, T. Stockhammer, G. Baese, J. Pandel, and M. Wien. *H.26L Core Experiment description on granular quality scalability.* Tech. Rep. J70, ITU-T SG16 Q.15, May 2000
Motivation for using scalability in H.26L and common testing conditions for the Core Experiment. 22

[HF03a] Till Halbach and Thomas R. Fischer. *Channel rate allocation for error-robust packet transmission of SNR-scalable DPCM-coded video.* In *Proc. International Symposium*

on SignalProcessing and its Applications (ISSPA). Paris (France), Jul. 2003
A joint source channel coding scheme for optimum low-complexity channel coding. 80, 133

[HF03b] Till Halbach and Thomas R. Fischer. *SNR scalability by transform coefficient refinement for block-based video coding*. In *Proc. SPIE's Visual Communications and Image Processing (VCIP)*. Lugano (Switzerland), Jul. 2003
A novel SNR scalable scheme as a possible extension of H.264. 25, 33

[HO04] Till Halbach and Steffen Olsen. *Error robustness evaluation of H.264/MPEG-4 AVC*. In *Proc. SPIE's Visual Communications and Image Processing (VCIP)*. San Jose (CA, USA), Jan. 2004
An overview and discussion of the error resilience features in H.264's Baseline and Extended profile with the focus on rate distortion issues. 18, 57

[HSLG99] Uwe Horn, K. Stuhlmüller, M. Link, and B. Girod. *Robust Internet video transmission based on scalable coding and unequal error protection*. Image Communication, Vol. 15, No. 1–2, pp. 77–94, Sep. 1999 52

[HW02] Till Halbach and Mathias Wien. *Concepts and performance of next-generation video compression standardization*. In *Proc. Nordic Signal Processing Symposium (NORSIG)*. on board Hurtigruten (Norway), Oct. 2002
A tutorial to H.26L/H.264/MPEG-4 AVC in FCD status plus an intra-frame coding mode performance comparison to JPEG, JPEG2000, and Motion JPEG2000. 13

[HWG00] Sang-Eun Han, Thomas Wedi, and Bernd Girod. *SNR scalable coding with leaky prediction*. Tech. Rep. N53, ITU-T Q.6/SG 16 (VCEG), Aug. 2000
An SNR scalability scheme with temporal prediction utilizing both low- and high-quality references. 23

[HYWL01] Yuwen He, Rong Yan, Feng Wu, and Shipeng Li. *H.26L-based fine granularity scalable video coding*. Tech. Rep. O60, ITU-T Q.6/SG 16 (VCEG), Dec. 2001

A variant of FGS exploiting both low- and high-quality reference frames to code the enhancement layer. 23

[IB00] Klaus Illgner and Gero Baese. *Experiment in H.26L fine granularity scalability*. Tech. Rep. K09, ITU-T SG16 Q.15, Aug. 2000
A report of results complementing the document J24 from the previous meeting. 23

[ISO94a] ISO/IEC. International Standard 10918-1, *Digital compression and coding of continuous-tone still images, Part 1: Requirements and guidelines*, (JPEG). 1994
The famous first still-image compression standard JPEG, also known as CCITT Recommendation T.81. 22

[ISO94b] ISO/IEC. International Standard 13818-2, *Generic coding of moving pictures and associated audio information — Part 2: Video*, (MPEG-2 Video). Nov. 1994
Identical with ITU-T Rec. H.262. 21

[ISO98] ISO/IEC. International Standard 15444-1, *JPEG2000 Image Coding System, Part I*, (JPEG2000). Dec. 1998
The first of second-generation still-image compression standards. 50

[ISO99] ISO/IEC. International Standard 14496-2, *Information technology — coding of audio-visual objects — Part 2: Visual*, (MPEG-4 Visual, version 1). Apr. 1999
Also published as Technical Report ISO/IEC JTC 1/SC 29/WG 11 N4350 in July 2001. 21

[ISO00] ISO/IEC. International Standard 14496-2, *Information technology — coding of audio-visual objects — Part 2: Visual*, (MPEG-4 Visual, version 2). Jan. 2000
This is IS 14496-2 with Amendment 1. 12

[ISO03] ISO/IEC. International Standard 14496-10, *Information technology — coding of audio-visual objects — Part 10: Advanced video coding*, (MPEG-4 AVC). Oct. 2003
Also known as H.26L. Identical with ITU-T Rec. H.264. 13

[ITU98] ITU-T. Recommendation H.263 version 2, *Video coding for low bitrate communication*, (H.263+ (H.263v2)). Jan. 1998
Also known as H.263+. 21, 101

[ITU03] ITU-T. Recommendation H.264, *Advanced video coding for generic audiovisual services*, (H.264). May 2003
Also known as H.26L. Identical with ISO/IEC IS 14496-10. 6, 13

[JHJ+03] Bongsoo Jung, Young Hooi Hwang, Byeungwoo Jeon, Myung Don Kim, and Song-In Choi. *Error resilient performance evaluation of MPEG-4 and H.264.* In *Proc. SPIE's Visual Communications and Image Processing (VCIP)*, pp. 1050–1061. Jul. 2003 57

[JKL98] Han-Seung Jung, Rin-Chul Kim, and Sang-Uk Lee. *On the robust transmission technique for H.263 video data stream over wireless networks.* In *Proc. IEEE Int. Conf. on Image Processing (ICIP)*, pp. 463–466. Oct. 1998 51

[KG98] A. K. Katsaggelos and N. P. Galatsanos (editors). *Signal Recovery Techniques for Image and Video Compression and Transmission.* Kluwer Academic Publishers, 1998 6, 70

[KIK+98a] A. K. Katsaggelos, F. Ishtiaq, I. P. Kondi, M.-C. Hong, M. Banham, and J. Brailean. *Error resilience and concealment in video coding.* In *Proc. of Eur. Signal Proc. Conf. (EUSIPCO)*, pp. 221–228. Rhodos (Greece), 1998 20

[KIK98b] L. P. Kondi, F. Ishtiaq, and A. K. Katsaggelos. *On video SNR scalability.* Proc. IEEE Int. Conf. on Image Processing (ICIP), Vol. 3, pp. 934–938, Oct. 1998
Proposes a new SNR scalability scheme with three layers, based on H.263. 44

[KIK02] Lisimachos P. Kondi, Faisal Ishtiaq, and Aggelos K. Katsaggelos. *Joint source-channel coding for motion-compensated DCT-based SNR scalable video.* IEEE Trans. Image Processing, Vol. 11, No. 9, pp. 1043–1052, Sep. 2002 7, 51

[KK01] L. P. Kondi and A. K. Katsaggelos. *An operational rate-distortion optimal single-pass SNR scalable video coder.* IEEE Trans. Image Processing, Vol. 10, No. 11, pp. 1613–1620, Nov. 2001 26

[KXP00] Beong-Jo Kim, Zixiang Xiong, and William A. Pearlman. *Low bit-rate scalable video coding with 3-D set partitioning in hierarchical trees (3-D SPIHT).* IEEE Trans. Circuits, Syst. for Video Technol., Vol. 10, No. 8, pp. 1374–1387, Dec.

2000
Describes the extension of the famouse SPIHT algorithm (for still images) to 3-D SPIHT (for videos). 50

[LCM84] S. Lin, D. J. Costello, and M. J. Miller. *Automatic repeat error control schemes.* IEEE Communications Mag., Vol. 22, pp. 5–17, 1984 49

[LEB+03] Carlos E. Luna, Yiftach Eisenberg, Randall Berry, Thrasyvoulos N. Pappas, and Aggelos K. Katsaggelos. *Joint source coding and data rate adaptation for energy efficient wireless video streaming.* J. Select. Areas Comm., Vol. 21, No. 10, pp. 1710–1720, Dec. 2003 51

[Li99] Weiping Li. *Overview of fine granularity scalability in MPEG-4 video standard.* J. Select. Areas Comm., Vol. 7, No. 5, pp. 771–781, Jun. 1999 21, 41

[Lil02] Karl Lillevold. *Proposed updates to CAVLC.* Tech. Rep. D036, ITU-T VCEG | ISO/IEC MPEG (JVT), Jul. 2002
Minor modifications of the H.264 modifications concerning entropy coding. 25

[LJL+03] Peter List, Anthony Joch, Jani Lainema, Gisle Bjøntegaard, and Marta Karczewicz. *Adaptive deblocking filter.* IEEE Trans. Circuits, Syst. for Video Technol., Vol. 13, No. 7, pp. 614–619, Jul. 2003
Contains references to all standardization contributions concerning the filter development. 31

[LOR98] T. V. Laksham, Antonio Ortega, and Amy R. Reibman. *VBR video: Tradeoffs and potentials.* Proc. IEEE, Vol. 86, No. 5, pp. 952–973, May 1998
An overview over subfields of research in variable bit rate video, including the historical development. Clarifies further frequently used terminology. 4

[LRL93] W.-M. Lam, A. R. Reibman, and B. Liu. *Recovery of lost or erroneously received motion vectors.* In *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Proc. (ICASSP)*, Vol. 1, pp. 417–420. Minneapolis (USA), Apr. 1993 56

[LZ97] Wenjun Luo and Magda El Zarki. *Quality control for VBR video over ATM networks.* J. Select. Areas Comm., Vol. 15, No. 6, pp. 1029–1039, Aug. 1997 52

[MBH00a] D. Marpe, G. Blättermann, and G. Heising. *Experiment in H.26L fine granularity scalability.* Tech. Rep. K10, ITU-T SG16 Q.15, Aug. 2000
Results for Techn. Report J70. 23

[MBH00b] D. Marpe, G. Blättermann, and G. Heising. *Technical description of a quality-scalable coder for H.26L.* Tech. Rep. K12, ITU-T SG16 Q.15, Aug. 2000 23

[MSW03] D. Marpe, H. Schwarz, and T. Wiegand. *Context-based adaptive binary arithmetic coding in the H.264/AVC video compression standard.* IEEE Trans. Circuits, Syst. for Video Technol., Vol. 13, No. 7, pp. 620–636, Jul. 2003
Describes the entropy coding engine CABAC of H.264's Main profile. 47

[MW00] Claudia Mayer and Mathias Wien. *Results of H.26L Core Experiment on fine granularity scalability.* Tech. Rep. K26, ITU-T SG16 Q.15, Aug. 2000
Results for Techn. Report J70. 23

[OLL+03] EePing Ong, Weisi Lin, Zhongkang Lu, Xiaokang Yang, Susu Yao, Xiao Lin, and F. Moschetti. *Qualitv evaluation of MPEG-4 and H.26L coded video for mobile multimedia communications.* In *Proc. International Symposium on SignalProcessing and its Applications (ISSPA)*, Vol. 1, pp. 473–476. Jul. 2003 57

[PLC+02] Gwang Hoon Park, Yoon Jin Lee, Won-Sik Cheong, Kyuheon Kim, Jinwoong Kim, and Young Kwon Lim. *Water ring scan method for H.26L based FGS.* Tech. Rep. B094, ITU-T VCEG | ISO/IEC MPEG (JVT), Feb. 2002
Combining a central region of interest with fine granular scalability. 23

[PM92] William P. Pennebaker and Joan L. Mitchell. *JPEG Still Image Data Compression Standard.* Van Norstrand Reinhold, New York (NY, USA), 1992
The ultimate description of the famous standard. 22, 42

[RAG04] S. Rane, A. Aaron, and B. Girod. *Systematic lossy forward error protection for error resilient digital video broadcasting.* In *Proc. SPIE's Visual Communications and Image Processing (VCIP).* Jan. 2004 52

[RB99] K. R. Rao and Zoran S. Bojkovic. *Packet video communications over ATM networks.* Prentice Hall, Upper Saddle River (NJ, USA), 1999 4

[RC99] H. Radha and Y. Chen. *Fine-granular-scalable video for packet networks.* In *Proc. Packet Video Workshop.* New York City (NY, USA), Apr. 1999 12

[RH96] K. R. Rao and J. J. Hwang. *Techniques and Standards for Image, Video, and Audio Coding.* Prentice Hall, New Jersey (USA), 1996 21, 79

[Rhe98] I. Rhee. *Error control techniques for interactive video transmission over the Internet.* In *Proc. ACM SIGCOMM*, pp. 290–301. Vancouver (Canada), Sep. 1998 12

[RKK98] M. A. Robers, L. P. Kondi, and A. K. Katsaggelos. *SNR scalable video coder using progressive transmission of DCT coefficients.* Proc. SPIE, pp. 201–212, 1998 22

[RM00] D. Rowitch and L. Milstein. *On the performance of hybrid FEC/ARQ systems using rate compatible punctured turbo (RCPT) codes.* IEEE Trans. Commun., Vol. 48, No. 6, Jun. 2000 86

[Sha48] C. E. Shannon. *A mathematical theory of communication.* Bell Syst. Tech. J., Vol. 27, pp. 379–423 and 623–656, Oct. 1948
The famous paper which provides a mathematical description of communication, including the source coding theorem and the channel coding theorem. 7

[Sha59] C. E. Shannon. *Coding theorems for a discrete source with a fidelity criterion.* IRE National Convention Record, Vol. Part 4, pp. 142–163, 1959
A mathematical framework describing rate distortion theory. 5

[SHW03] Thomas Stockhammer, Miska M. Hannuksela, and Thomas Wiegand. *H.264/AVC in wireless environments.* IEEE

Trans. Circuits, Syst. for Video Technol., Vol. 13, No. 7, pp. 657–673, Jul. 2003
!!!! 57

[SMW03] Heiko Schwarz, Detlev Marpe, and Thomas Wiegand. *SNR-scalable extension of H.264/AVC*. Tech. Rep. I032, ITU-T VCEG | ISO/IEC MPEG (JVT), Sep. 2003
Combining hybrid and subband coding based on lifting filter structures. 22, 23

[SMW04] Heiko Schwarz, Detlev Marpe, and Thomas Wiegand. *Subband extension of H.264/AVC*. Tech. Rep. K023, ITU-T VCEG | ISO/IEC MPEG (JVT), Mar. 2004
A 3-D video coding scheme based on lifting structures superior to the newly standardized H.264. Also published as VCEG-U05. 24

[SSD98] P. Salama, N. B. Shroff, and E. J. Delp. *Error concealment in encoded video streams*, Chap. 7. Kluwer Academic Publishers, 1998 56

[SW98] Gary J. Sullivan and Thomas Wiegand. *Rate-distortion optimization for video compression*. IEEE Signal Processing Mag., Vol. 15, pp. 74–90, Nov. 1998
On coding mode decision. With a nive overview over video compression standards. 28

[TC67] R. E. Totty and G. C. Clark. *Reconstruction error in waveform transmission*. IEEE Transactions on Information Theory, Vol. 13, pp. 336–338, Apr. 1967 65

[TN02] J. Tavares and A. Navarro. *Forward error protection of H.263-DG bit streams*. In *IEEE MELECON*, pp. 390–394. May 2002 51

[tT00] Wai tian Tan. *Video Compression and Streaming over Packet-switched Networks*. Ph.D. thesis, Graduate Division, University of California at Berkeley, Berkely (CA, USA), 2000 12, 83

[Ver86] W. Verbiest. *Video coding in an ATM environment*. In *Int. Conf. on New Systems and Services in Telecomm.* Liege (Belgium), Nov. 1986
Mentions layered video coding for the first time in the literature. 11

[VU92] M. Vetterli and K. Uz. *Multiresolution coding techniques for digital television: A review.* Multidimensional Systems and Image Processing, Vol. 3, pp. 161–187, 1992
An overview on scalability techniques. 22

[VZW99] J. D. Villasenor, Y.-Q. Zhang, and J. Wen. *Robust video coding algorithms and systems.* Proc. IEEE, Vol. 87, pp. 1724–1733, Oct. 1999 20

[Wen02] Stephan Wenger. *On the equivalence of BERs and packet loss rates.* Tech. Rep. B025, ITU-T VCEG | ISO/IEC MPEG (JVT), Jan. 2002
A somewhat polarizing report with heuristic scientific reasoning. 6

[Wen03] Stephan Wenger. *H.264/AVC over IP.* IEEE Trans. Circuits, Syst. for Video Technol., Vol. 13, No. 7, pp. 645–656, Jul. 2003
Explains the interface between video coding engine and underlying network. H.264's error resilience tools are presented, as well as simulation results. 20, 57

[WG96] D. Wilson and M. Ghanbari. *Transmission of SNR scalable two-layer MPEG-2 coded video through ATM networks.* In *Proc. Packet Video Workshop*, pp. 37–42. Brisbane, Australia, Mar. 1996
Includes a performance evaluation for layered coding in MPEG-2. 40

[WG97] D. Wilson and M. Ghanbari. *Optimization of two-layer SNR scalability for MPEG-2 video.* In *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Proc. (ICASSP)*, Vol. 4, pp. 2637–2640. Apr. 1997
A comparison of SNR scalability compliant with MPEG-2 and an own scheme based on data partitioning. 35

[Wie02] Thomas Wiegand. *Joint Final Committee Draft (JFCD) of joint video specification ITU-T Rec. H.264 | ISO/IEC 14496-10 AVC.* Tech. Rep. D157, ITU-T VCEG | ISO/IEC MPEG (JVT), Aug. 2002
The third last version of the description of the video compression standard H.264 before becoming International Standard or Recommendation, respectively. 33

[Win00] Stefan Winkler. *Vision Models and Quality Metrics for Image Processing Applications*. Ph.D. thesis, École Polytechnique Fédérale de Lausanne, (Switzerland), 2000
Includes an excellent introduction to the human visual system. 123

[WSBL03] Thomas Wiegand, Gary J. Sullivan, Gisle Bjøntegaard, and Ajay Luthra. *Overview of the H.264/AVC video coding standard*. IEEE Trans. Circuits, Syst. for Video Technol., Vol. 13, No. 7, pp. 560–576, Jul. 2003
A survey of the standard from the key persons. Lists all technical features and relates them to other standards. Gives further an outline about the standard's history and the standardization process. 3, 22, 32, 33

[WSJ+03] Thomas Wiegand, Heiko Schwarz, Anthony Joch, Faouzi Kossentini, and Gary J. Sullivan. *Rate-constrained coder control and comparison of video coding standards*. IEEE Trans. Circuits, Syst. for Video Technol., Vol. 13, No. 7, pp. 688–703, Jul. 2003
A technical comparisons of the key elements of standards like MPEG-2 Video, H.263, MPEG-4 Visual, and H.264, followed by a theoretic part on determination of the optimum coding mode. More importantly, very detailed and extensive comparisons among the abovementioned standards are given. 46

[WWWK00] Yao Wang, Stephan Wenger, Jiangtao Wen, and Aggelos K. Katsaggelos. *Error resilient video coding techniques*. IEEE Signal Processing Mag., Vol. 17, No. 4, pp. 61–82, Jul. 2000 20

[WZ98] Y. Wang and Q. Zhu. *Error control and concealment for video communication: A review*. Proc. IEEE, Vol. 86, pp. 974–997, May 1998 20

[WZZ02] G. Wang, Q. Zhang, and W. Zhu. *Channel-adaptive error protection for scalable video over channels with bit errors and packet erasures*. In *Proc. Int. Symp. on Circuits and Systems (ISCAS)*. 2002 52

[WZZZ00] G. Wang, Q. Zhang, W. Zhu, and Y.-Q. Zhang. *Channel-adaptive error control for scalable video over wireless channel*. In *Proc. Int. Workshop Mobile Multimedia Comm. (MoMuC)*. Oct. 2000 52

[WZZZ01] G. Wang, Q. Zhang, W. Zhu, and Y.-Q. Zhang. *Channel-adaptive unequal error protection for scalable video transmission over wireless channel.* In *Proc. SPIE's Visual Communications and Image Processing (VCIP).* Jan. 2001 52

[YZLF02] X. K. Yang, C. Zhu, Z. G. Li, and G. N. Feng. *Unequal error protection for motion compensated video streaming over the Internet.* In *Proc. IEEE Int. Conf. on Image Processing (ICIP)*, pp. 717–720. 2002 52

[ZA00] M. Zhao and A. N. Akansu. *Optimization of dynamic UEP schemes for embedded image sources in noisy channels.* In *Proc. IEEE Int. Conf. on Image Processing (ICIP).* Vancouver (Canada), Sep. 2000 50

[ZAA00] M. Zhao, A. A. Alatan, and A. N. Akansu. *A new method for optimal rate allocation for progressive image transmission over noisy channels.* In *Proc. Data Compression Conference (DCC)*, pp. 213–222. Snowbird (UT, USA), Mar. 2000 93

[ZBPK03] Fan Zhai, Randall Berry, Thrasyvoulos N. Pappas, and Aggelos K. Katsaggelos. *A rate-distortion optimized error control scheme for scalable video streaming over the Internet.* In *Proc. IEEE Int. Conf. on Multimedia and Expo.* Baltimore (MD, USA), Jul. 2003 51, 101

[ZEL+03] F. Zhai, Y. Eisenberg, C. E. Luna, T. N. Pappas, R. Berry, and A. K. Katsaggelos. *Packetization schemes for forward error correction in Internet video streaming.* In *Proc. Allerton Conf. Comm., Control, and Computing.* Oct. 2003 52

[ZLE+03] F. Zhai, C. E. Luna, Y. Eisenberg, T. N. Pappas, R. Berry, and A. K. Katsaggelos. *Joint source coding and packet classification for real-time video transmission over differentiated services networks.* IEEE Trans. Multimedia, Jan. 2003 52