

University of Southampton Research Repository ePrints Soton

Copyright © and Moral Rights for this thesis are retained by the author and/or other copyright owners. A copy can be downloaded for personal non-commercial research or study, without prior permission or charge. This thesis cannot be reproduced or quoted extensively from without first obtaining permission in writing from the copyright holder/s. The content must not be changed in any way or sold commercially in any format or medium without the formal permission of the copyright holders.

When referring to this work, full bibliographic details including the author, title, awarding institution and date of the thesis must be given e.g.

AUTHOR (year of submission) "Full thesis title", University of Southampton, name of the University School or Department, PhD Thesis, pagination

UNIVERSITY OF SOUTHAMPTON
FACULTY OF PHYSICAL SCIENCES AND ENGINEERING
ELECTRONICS AND COMPUTER SCIENCE

Iterative Source-and-Channel Decoding Aided Video Communications

by

Yongkai Huo
BEng, MEng

A thesis submitted for the award of Doctor of Philosophy
at the University of Southampton

November 2013

Supervisor: *Prof. Lajos Hanzo*
Dipl Ing, MSc, PhD, DSc, FIET, FIEEE, FREng
Chair in Telecommunications, Head of Group
Electronics and Computer Science
University of Southampton
Southampton, SO17 1BJ
United Kingdom

Dedicated to my family

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

Faculty of Physical Sciences and Engineering
School of Electronics and Computer Science

A thesis submitted for the award of Doctor of Philosophy

**Iterative Joint Source-and-Channel Decoding Aided Video
Communications**

by Yongkai Huo

Shannon's source-and channel-coding separation theorem states that reliable near-capacity transmission can be accomplished by separate source coding using lossless entropy codes and channel coding under the idealized assumption of Gaussian channels and potentially infinite encoding/decoding delay as well as complexity. However, it is impossible to remove all source redundancy with the aid of practical finite-delay and finite-complexity source encoders. As a remedy, joint source-channel coding (JSCC) has been used for achieving an improved system performance by exploiting the residual source correlation.

We propose a novel tree-structured multiple description coding (T-MDC) scheme that may be combined with arbitrary video codecs for the sake of creating multiple video descriptions. The technique advocated splits the original video signal into an appropriately chosen number of correlated descriptions in the time-domain, while retaining the correlation among the video frames within each description. Each description may be encoded using arbitrary video compression tools into a bitstream. T-MDC was also employed in the scenario of multiple description coding for multiview video communications. Furthermore, our proposed scheme is also capable of splitting the video stream into multiple descriptions of unequal importance.

Then, a novel inter-layer forward error correction (IL-FEC) coded video scheme is proposed, where the information of the base layer (BL) is incorporated into the systematic bits of the enhancement layers (ELs) with the aid of an exclusive or (XOR) operation. When the BL can be successfully decoded in its own right, the systematic bits of the ELs can be extracted by flipping the sign of the check information received without introducing any degradation, where the check information is generated by performing IL XOR operations on the BL and the ELs. However, when the BL cannot be correctly decoded without the assistance of the ELs, the IL-FEC decoding philosophy exchanging information between the BL and the ELs will be activated to assist in decoding the BL.

We then conceive a two-dimensional (2D) iterative Markov process aided decoder for a video receiver, which may be combined with channel decoding. Furthermore, a reduced-complexity first-order Markov model based source decoder will be derived. Iterative decoding is performed by exchanging extrinsic information between two source decoders. Explicitly, we propose the first-order Markov process aided three-dimensional (3D) iterative source-channel decoding (ISCD) concept relying on an recursive systematic convolutional (RSC) codec invoked for uncompressed video transmissions, where both the horizontal and vertical intra-frame correlations as well as the inter-frame correlations are exploited. The proposed technique is capable of exploiting both the intra-frame and inter-frame correlations for iterative source-channel decoding.

Finally, we study the application of ISCD conceived for distributed video coding (DVC), where the video signals are modelled by a first-order Markov process. A horizontal and a vertical source decoder are employed for exchanging their information using the iterative decoding philosophy. This scheme may be combined with the entire suite of classic FEC codecs employed in state-of-the-art DVC systems. We benchmark the attainable system performance against that of the existing pixel-domain Wyner-Ziv (PDWZ) video coding systems. Finally, we exploit the inter-view correlation with the aid of inter-view motion search in distributed multi-view video coding (DMVC). We rely on the system architecture of WZ coding invoked for multi-view video. We construct a novel mesh-structured pixel-correlation model from the inter-view motion vectors (MVs) and derive its decoding rules for joint source-channel decoding (JSCD). The proposed system was benchmarked against the existing PDWZ coding based DMVC scheme.

Declaration of Authorship

I, **Yongkai Huo**, declare that the thesis entitled

Iterative Source-and-Channel Decoding Aided Video Communications

and the work presented in it are my own and have been generated by me as the result of my own original research. I confirm that:

1. This work was done wholly or mainly while in candidature for a research degree at this University;
2. Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated;
3. Where I have consulted the published work of others, this is always clearly attributed;
4. Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work;
5. I have acknowledged all main sources of help;
6. Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself;
7. Parts of this work have been published, as seen in the list of publications.

Signed:

Date:

Acknowledgements

First and foremost, I would like to express my gratitude to Professor Lajos Hanzo for his valuable guidance and advice throughout my research. His enthusiasm, inspiration and perseverance not only cultivate in me to be a good researcher, but also benefit me in my life. His willingness to give his time so generously has been very much appreciated. It would not be possible to finish this thesis without his friendly and generous help.

I would also like to thank the co-authors of my papers: Lajos Hanzo, Tao Wang, Mohammed El-Hajjar, Xin Zuo, Robert G. Maunder, Chuan Zhu for their willingness to help, discussions, knowledge.

I must also acknowledge all my colleagues in the Communications group for their help and friendship. Special thanks go to Wenbo Zhang, Chao Xu, Soon Xin Ng, Chen Dong, Dandan Liang, Li Li, Kent Tsz Kan Cheung, Jing Zuo, Rong Zhang, Bo Zhang, Hong Chen and all others not mentioned here.

Finally, to my family, particularly my parents, my sisters and my brother, for their unconditional love, support and care. Without you, I would not be where I am today.

Contents

Abstract	iii
Declaration of Authorship	v
Acknowledgements	vi
List of Publications	xiii
Chapter 1 Introduction	1
1.1 Historical Overview of Video Compression	1
1.1.1 Brief History of Video Compression	1
1.1.2 Multiple Description Coding	6
1.1.3 Standardized Layered Video Coding Techniques	8
1.1.4 Preliminaries for the Evaluation of Video Transmission	10
1.1.4.1 Video Quality Metric	10
1.1.4.2 Annex B Format of H.264/AVC	11
1.1.4.3 Experimental Evaluation of Scalable Video Coding	12
1.2 Video Transmission Exploiting Redundancy	17
1.2.1 Unequal Error Protection	18
1.2.2 Joint Source-Channel Decoding	22
1.2.3 Distributed Video Coding	24
1.2.3.1 Wyner-Ziv Coding Theory	24

1.2.3.2	Advances in Distributed Video Coding	28
1.3	Extrinsic Information Transfer Charts	32
1.4	Motivation and Methodology	34
1.5	Outline of the Thesis	35
1.6	Novel Contributions	38
Chapter 2	Tree-Structured Multiple Description Coding	40
2.1	Introduction	40
2.2	Tree-Structured Multiple Description Creation	41
2.2.1	Descriptions of Equal Importance	42
2.2.2	Descriptions of Unequal Importance	44
2.3	Video Reconstruction	45
2.3.1	Reconstruction for Lossless Aliquot Encoding	48
2.3.2	Reconstruction for Lossy Aliquot Encoding	49
2.4	Multiple Description Coding of Multiview Video	52
2.5	Simulation Results	54
2.5.1	Multiple Description Codec Performance	54
2.5.2	Performance of MVC with T-MDC	57
2.6	Summary and Conclusions	58
Chapter 3	Inter-Layer Coded Unequal Error Protection for Layered Video Trans- mission	62
3.1	Introduction	62
3.2	Inter-Layer FEC Coded Layered Video Streaming	64
3.2.1	Transmitter Model	67
3.2.2	Receiver Model	68
3.2.2.1	With Inter-Layer Feedback	72
3.2.2.2	Without Inter-Layer Feedback	72
3.2.3	Example of IL FEC Coding and Decoding	73
3.2.3.1	Operation Without IL Feedback	74

3.2.3.2	Operation With IL Feedback	76
3.2.4	Inter-Layer FEC Coding for Layers Having Unequal Length	79
3.2.5	IL-FEC Overheads	81
3.3	EXIT Analysis of IL-FEC Coded Systems	83
3.3.1	Mutual Information Analysis Using RSC	83
3.3.2	EXIT Analysis Using Turbo	84
3.4	Distortion Minimization by Adjusting the FEC Coding Rate	89
3.4.1	Preliminaries	92
3.4.1.1	Lookup Tables	92
3.4.1.2	MI Flow of the VND	93
3.4.1.3	MI Flow of the CND	93
3.4.2	PER Estimation	95
3.4.2.1	IL-FEC Coded BL	95
3.4.2.2	FEC Coded EL	98
3.4.3	Optimized Coding Rates	99
3.4.4	Overheads	100
3.4.4.1	Estimation of $d(\cdot)$	101
3.4.4.2	Generation of LUTs	101
3.4.4.3	Estimation of $p(\cdot)$	101
3.4.4.4	Full Search of r_0	101
3.4.4.5	Delay	102
3.5	System Performance for Partitioned H.264	102
3.5.1	Error Protection Arrangements	103
3.5.2	System Performance using RSC Codec	105
3.5.3	System Performance using Turbo Coding	111
3.5.4	Complexity Analysis	116
3.6	System Performance for Scalable Video Coding	120
3.6.1	Self-Concatenated Convolutional Code	121
3.6.1.1	Encoder Structure	121

3.6.1.2	Decoder Structure	121
3.6.1.3	Performance Study	122
3.6.2	Error Protection Arrangements	124
3.6.3	System Performance Using SECCC	124
3.6.4	Complexity Analysis	130
3.7	System Performance for Optimized Scalable Video Coding	132
3.7.1	Off-line LUTs Generation	133
3.7.2	System Performance	134
3.7.3	Optimized Coding Rates	141
3.8	Summary and Conclusions	141
Chapter 4	Iterative Source-Channel Decoding Aided Video Transmission	147
4.1	Introduction	147
4.2	Intra-Frame Iterative Source-Channel Decoding	149
4.2.1	Iterative Source Decoding Model	149
4.2.1.1	Softbit Input	150
4.2.1.2	Iterative Source Decoding	151
4.2.2	Existing First-Order Markov Modeled Softbit Source Decoding	155
4.2.3	Reduced Complexity Markov-Modelled Softbit Source Decod- ing	157
4.2.3.1	Trellis Representation of First-Order Markov Chain .	157
4.2.3.2	BCJR Decoding of First-Order Markov Chain	158
4.2.3.3	Extrinsic Information Exchange for Iterative Decoding	159
4.2.3.4	Complexity Analysis	161
4.2.3.5	Parameter Training for Markov Processes	162
4.2.4	ISCD Aided Uncompressed Video Transmission	163
4.2.4.1	System Configuration	164
4.2.4.2	Benchmarks	167
4.2.4.3	Numerical Results	168
4.3	Spatio-Temporal Iterative Source-Channel Decoding	174

4.3.1	System Overview	174
4.3.1.1	Transmitter	175
4.3.1.2	Receiver	176
4.3.2	Performance Analysis	180
4.3.2.1	Scenario	180
4.3.2.2	Three-Dimensional EXIT Charts	181
4.3.2.3	Benchmarks	183
4.3.2.4	Numerical Results	186
4.3.2.5	System Optimization	190
4.4	Summary and Conclusions	192
Chapter 5 Iterative Source-Channel Decoding Aided Distributed Video Coding		197
5.1	Introduction	197
5.2	A Pixel-Domain Wyner-Ziv Architecture	198
5.2.1	Encoder Operations	199
5.2.2	Decoding Operations	201
5.3	IHVSM aided Wyner-Ziv Monoscopic Video Coding	202
5.3.1	System Configuration	203
5.3.2	Numerical Results	205
5.4	Motion-Aware Mesh-Structured Correlation Modelling	208
5.4.1	Wyner-Ziv Coding for Multi-view Video: System Model	208
5.4.1.1	Transmitter	209
5.4.1.2	Receiver	210
5.4.2	Inter-view Correlation Modelling	212
5.4.2.1	Mesh-Structured Trellis Representation	213
5.4.2.2	Trellis Decoding	215
5.4.2.3	Iterative MSSM-Turbo Decoding	216
5.4.2.4	Training for Mesh-Structured State Transition	218
5.4.2.5	Complexity Analysis	218
5.4.3	Performance Study	219

5.4.3.1	Scenario	219
5.4.3.2	Numerical Results	220
5.5	Summary and Conclusions	228
Chapter 6	Conclusions and Future Work	231
6.1	Summary and Conclusions	231
6.2	Design Guidelines	233
6.3	Suggestions for Future Work	235
6.3.1	Optimization of T-MDC	235
6.3.2	Joint Encoding of the BL and ELs	236
6.3.3	ISCD Aided DVC	236
6.3.4	Joint Intra- and Inter-View ISCD	237
6.3.5	Reduced-Complexity ISCD	237
Appendices		238
Appendix A	Decoding of First-Order Markov Process	238
Appendix B	Derivation of MSSM Decoder	243
Glossary		247
Bibliography		254
Author Index		270
Subject Index		277

List of Publications

Journal Paper

1. **Y. Huo**, M. El-Hajjar, R. G. Maunder, and L. Hanzo, “Layered wireless video telephony relying on minimum-distortion inter-layer FEC coding,” *IEEE Transactions on Multimedia*, early access, 2014.
2. **Y. Huo**, T. Wang, R. G. Maunder, and L. Hanzo, “Two-dimensional iterative source-channel decoding for distributed video coding,” *IEEE Communications Letters*, vol. 18, no. 1, pp. 90–93, 2014.
3. **Y. Huo**, M. El-Hajjar, and L. Hanzo, “Wireless video: An inter-layer error protection aided multi-layer approach,” *IEEE Vehicular Technology Magazine*, submitted for publication.
4. **Y. Huo**, T. Wang, R. G. Maunder, and L. Hanzo, “Motion-aware mesh-structured trellis for correlation modelling aided distributed multi-view video coding,” *IEEE Transactions on Image Processing*, vol. 23, no. 1, pp. 319–331, 2014.
5. **Y. Huo**, T. Wang, R. G. Maunder, and L. Hanzo, “Iterative source and channel decoding relying on correlation modelling for wireless video transmission,” *IET Communications*, vol. 7, pp. 1465–1475, September 2013.
6. **Y. Huo**, M. El-Hajjar, and L. Hanzo, “Inter-layer FEC aided unequal error protection for multi-layer video transmission in mobile TV,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 23, no. 9, pp. 1622–1634, 2013.
7. **Y. Huo**, C. Zhu, and L. Hanzo, “Spatio-temporal iterative source-channel decoding aided video transmission,” *IEEE Transactions on Vehicular Technology*, vol. 62, no. 4, pp. 1597–1609, 2013.

Conference Paper

1. C. Dong, J. Zuo, L.-L. Yang, **Y. Huo**, S. X. Ng, and L. Hanzo, “Energy-Efficient Buffer-Aided Relaying Relying on Non-Linear Channel Probability Space Division,” *IEEE Wireless Communications and Networking Conference (WCNC)*, in press.
2. **Y. Huo**, M. El-Hajjar, and L. Hanzo, “Inter-layer turbo coded unequal error protection for multi-layer video transmission,” in *IEEE Vehicular Technology Conference (VTC Fall)*, Las Vegas, USA, September 2013.

3. **Y. Huo**, M. El-Hajjar, M. F. U. Butt, and L. Hanzo, “Inter-layer-decoding aided self-concatenated coded scalable video transmission,” IEEE Wireless Communications and Networking Conference (WCNC), pp. 4647–4652, 2013.
4. C. Zhu, **Y. Huo**, R. G. Maunder, S. Kawade, and L. Hanzo, “Iterative joint source-channel decoding aided transmission of losslessly compressed video,” in IEEE Wireless Communications and Networking Conference (WCNC), pp. 4629–4634, 2013.
5. **Y. Huo**, X. Zuo, R. G. Maunder, and L. Hanzo, “Inter-layer FEC decoded multi-layer video streaming,” in IEEE Global Telecommunications Conference (GLOBECOM), pp. 2113–2118, 2012.
6. **Y. Huo**, and L. Hanzo, “Tree-structured multiple description coding for multiview mobile TV and camera-phone networks,” in IEEE Vehicular Technology Conference (VTC Fall), 2011, pp. 1–5, September 2011.

List of Symbols

Tree-structured Multiple Description Coding (TS-MDC)

$x_{i,j}$	The pixel value at position (i, j) in a specific video frame.
L	The quantizer resolution.
S_L	The set of descriptions $\{0, \dots, L-1\}$ received at the receiver, where each element corresponds to an offset invoked for creating a specific description.
X_m	The set of m -bit luminance pixel values, namely $\{0, \dots, 2^m - 1\}$.
$x_{i,j}^l$	The aliquot part of the pixel $x_{i,j}$ upon division by L in the l^{th} description, $l \in S_L$.
d^l, \tilde{d}^l	The l^{th} bitstream at the transmitter and receiver respectively, $l \in S_L$.
$y_{i,j}^l$	The l^{th} aliquot part of the pixel $x_{i,j}$ reconstructed by the aliquot decoder at the receiver, $l \in S_L$.
$\hat{x}_{i,j}$	The estimated pixel value at position (i, j) .
δ^l	The aliquot reconstruction error $y^l - x^l$ of the l^{th} description.
K	Number of views in the source multiview video.
$p(x y_0^{L-1})$	The APP occurrence of pixel x conditioned on the received descriptions y_0^{L-1} .
$p(x)$	The distribution probability of pixel x .

Inter-layer Coded Unequal Error Protection (IL-UEP)

u_i	Random binary variable.
$L(u_i)$	LLR of the variable u_i .
x_a	Binary sequence representing partition A .
x_b	Binary sequence representing partition B .
x_c	Binary sequence representing partition C .
x_{ab}	Check bits generated from partition A and partition B .

x_{ac}	Check bits generated from partition A and partition C .
$x_{a,p}$	Parity bits generated for systematic bits x_a .
$x_{b,p}$	Parity bits generated for systematic bits x_b .
$x_{c,p}$	Parity bits generated for systematic bits x_c .
π	Bit level interleaver.
N_t	Number of transmitter antennas.
N_r	Number of receiver antennas.
y_a	Received soft version of partition A .
y_b	Received soft version of partition B .
y_c	Received soft version of partition C .
y_{ab}	Received soft version of check bits x_{ab} .
y_{ac}	Received soft version of check bits x_{ac} .
$y_{a,p}$	Received soft version of parity bits $x_{a,p}$.
$y_{b,p}$	Received soft version of parity bits $x_{b,p}$.
$y_{c,p}$	Received soft version of parity bits $x_{c,p}$.
$L_e(x)$	Extrinsic LLR of bits x .
$L_a(x)$	<i>A-priori</i> LLR of bits x .
\hat{x}_a	Estimated systematic bits x_a .
\hat{x}_b	Estimated systematic bits x_b .
\hat{x}_c	Estimated systematic bits x_c .
n_a	Number of bits contained in the partition A .
n_b	Number of bits contained in the partition B .
n_c	Number of bits contained in the partition C .
T_b^i	A set containing a number of bits.
t_b	A bit sequence generated from $T_b^1, \dots, T_b^{n_b}$.
T_c^i	A set containing a number of bits.
t_c	A bit sequence generated from $T_c^1, \dots, T_c^{n_c}$.
L_0	The base layer (BL).
L_i	An enhancement layer (EL).
x_i	Bit sequence representing layer L_i .
x_{01}	Check bits generated for layer L_0 and layer L_1 .
$d(L_0)$	The video distortion induced by the corruption of the BL L_0 .
$d(L_1)$	The peak signal-to-noise ratio (PSNR) distortion, when the BL L_0 is correct while the EL L_1 is corrupted.
$ L_i $	The length of the bitstream of layer L_i .
R	The overall coding rate of the system.
r_i	The coding rate of layer L_i .
$p(L_0)$	The packet error ratio (PER) of layer L_0 .

$p(L_1)$	The PER of layer L_1 , when the layer L_0 is correctly decoded.
SNR	Value of the signal to noise ratio (SNR).
$f_0(SNR, L_0 , L_1 , r_0, r_1)$	PER of layer L_0 , given the parameters $SNR, L_0 , L_1 , r_0, r_1$.
$f_1(SNR, L_1 , r_1)$	PER of layer L_1 , given the parameters SNR, L_1 , r_1 .
$d(L_0, L_1)$	The PSNR distortion jointly imposed by layer L_0 and layer L_1 .
$E[d(L_0, L_1)]$	The expected PSNR distortion jointly imposed by layer L_0 and layer L_1 .
$T_b(SNR)$	The MI value of the LLRs output by the BPSK demodulator at a given SNR .
I_s	The MI value between the <i>a-priori</i> LLRs of the systematic information and the corresponding information bits.
$T_e(SNR, I_s, r)$	The MI value of the extrinsic LLR output of the RSC decoder recorded for a given SNR.
$T_p(SNR, I_s, r)$	The PER value associated with the LLRs output by the RSC decoder at diverse channel SNRs.
$I[u; L(u)]$	The MI value between the LLRs $L(u)$ and the information bits u .
σ^2	Variance of the Gaussian random variable representing the virtual extrinsic AWGN channel.
$J(\sigma)$	Accumulated mutual information of the virtual extrinsic AWGN channel.
$J^{-1}(I)$	Inverse function of $J(\sigma)$.

ISCD Aided Video Transmission

x_i	The m -bit pattern of a pixel scanned from the original video pixels at time instant i .
m	The number of bits in each m -bit pattern x_i of pixels.
$x_i(j)$	The j^{th} bit of the pixel x_i .
X_m	The set of all possible values in an m -bit pattern x_i , namely $\{0, 1, \dots, 2^m - 1\}$.
x_0^t	The bit patterns of the 1 st frame of the original video consisting of $(t + 1)$ m -bit patterns during the time interval spanning from 0 to t , namely x_0, \dots, x_t .
y_0^t	Potentially error-infested bit pattern of the 1 st frame, namely y_0, \dots, y_t .
H	Number of horizontal scanlines.
V	Number of vertical scanlines.
R	Reordering operation of pixel blocks.
\hat{x}_i	Estimated version of pixel x_i .
$p(x_i y_0^t)$	APP of pixel x_i conditioned on the received pixels y_0^t .

$L[x_i y_0^t]$	LLR version of the APP $p(x_i y_0^t)$.
$L[x_i(k)]$	<i>A-priori</i> LLR of the bit $x_i(k)$.
$L[y_i(k) x_i(k)]$	Systematic LLR information of the bit $x_i(k)$.
$p[x_i(k) y_0^t]$	APP of bit $x_i(k)$ conditioned on the received pixels y_0^t .
$L_e[x_i(k)]$	Extrinsic LLR of the bit $x_i(k)$.
$\gamma_i(x_i)$	Occurrence probability of the pixel x_i .
$\gamma_i^{[ext]}[x_i(k)]$	Extrinsic occurrence probability of the bit $x_i(k)$.
$\alpha_{i-1}(x_{i-1})$	Occurrence probability of pixel x_{i-1} at time instant $i-1$.
$A_{i-1}(x_{i-1})$	Logarithm version of the probability $\alpha_{i-1}(x_{i-1})$.
$\beta_i(x_i)$	Occurrence probability of pixel x_i at time instant i .
$B_i(x_i)$	Logarithm version of the probability $\beta_i(x_i)$.
$p(x_{i+1} x_i)$	Markov transition probability of pixel x_i to pixel x_{i+1} .
$\chi_i(x_i)$	Occurrence probability of the pixel x_i .
$\Gamma_i(x_i)$	Logarithm version of the probability $\chi_i(x_i)$.
f	Number of consecutive frames.
π	A bit-interleaver.
v_i	A RSC coded symbol.
$u_{s,i}$	Systematic component of the symbol v_i .
$u_{p,i}$	Parity component of the symbol v_i .
N	Number of consecutive correlated pixels.
$y_{s,i}$	Received soft information of the systematic component $u_{s,i}$.
J_a	<i>A-priori</i> information.
J_e	Extrinsic information.
M_a	<i>A-priori</i> information.
M_e	Extrinsic information.
O_a	<i>A-priori</i> information.
O_e	Extrinsic information.
g_i	A generator polynomial.
S_r	The total uncompressed size of a video file.
S_e	The entropy of the video source file.
r	The “Natural” coding rate of the source video.
$I(\cdot)$	The MI value between the <i>a-priori</i> LLRs of the corresponding information bits.

ISCD Aided Distributed Video Coding (DVC)

X	A binary source.
-----	------------------

Y	Binary source, which is independently and identically distributed with X (i.i.d.).
$H(X)$	Entropy of source X .
$H(Y)$	Entropy of source Y .
$H(X, Y)$	Joint entropy of sources X and Y .
R_X	The Slepian-Wolf encoding rate for X .
R_Y	The Slepian-Wolf encoding rate for Y .
$H(X/Y)$	Conditional entropy of source X given source Y .
$H(Y/X)$	Conditional entropy of source Y given source X .
u	The key frame.
\hat{u}	Reconstructed key frame.
v	The Wyner-Ziv (WZ) frame.
\hat{v}	Reconstructed WZ frame.
q	2D signal quantized from WZ frame v .
q_p	Parity bits of signals q .
\tilde{v}	Side information (SI) of the WZ frame v .
\hat{q}	Recovered version of signals q .
m	Number of most significant bit (MSB) planes.
I	Number of camera views.
G_i	The original monoscopic frame of the camera view i .
F_i	The SI of the frame G_i .
$x_{i,j}$	The j^{th} pixel in G_i , namely the pixel at position (i, j) .
$y_{i,j}$	The SI of the pixel $x_{i,j}$, which is at position j of F_i .
m	The number of bits contained in the pixel $x_{i,j}$.
X_m	The set of legitimate values of an m -bit pattern $x_{i,j}$, namely $\{0, 1, \dots, 2^m - 1\}$.
$L_{i,j,k}$	The set of the original pixels linked with $x_{i,j}$ in camera view k .
$L_{i,j,1}^I$	Notation for the set $\bigcup_{t=1}^I L_{i,j,t}$.
$Y_{i,j,k}$	The corresponding SI of $L_{i,j,k}$.
$Y_{i,j,1}^I$	Notation for the set $\bigcup_{t=1}^I Y_{i,j,t}$.
$p(x_{i,j} F_1, \dots, F_I)$	APP of pixel $x_{i,j}$ conditioned on the SI of the current multi-view WZ frame F_1, \dots, F_I .
$p(x_{i,j} Y_{i,j,1}^I)$	APP of pixel $x_{i,j}$ conditioned on the SI of the pixels $L_{i,j,1}^I$.
$L[x_{i,j}(k) Y_{i,j,1}^I]$	APP of bit $x_{i,j}(k)$ conditioned on the SI of the pixels $L_{i,j,1}^I$.
$\alpha_{i,j}(x_{i,j})$	Occurrence probability of pixel $x_{i,j}$.
$\beta_{i,j}(x_{i,j})$	Occurrence probability of pixel $x_{i,j}$.
$\chi_{i,j}(x_{i,j})$	Occurrence probability of the pixel $x_{i,j}$.
$L[x_{i,j}(k)]$	LLR distribution probability of bit $x_{i,j}(k)$.

$\gamma_{i,j}(x_{i,j})$	Occurrence probability of the pixel $x_{i,j}$.
$L[y_{i,j}(k) x_{i,j}(k)]$	The channel information LLR.
$L_e[x_{i,j}(k)]$	The extrinsic LLR information of bit $x_{i,j}(k)$.
$\gamma_{i,j}^{[ext]}[x_{i,j}(k)]$	The extrinsic LLR information of bit $x_{i,j}(k)$.
g_i	A generator polynomial.
I_{iter}	Number of iterations in iterative decoding.
$C_{\chi_{i,j}}$	A normalization factor.

Introduction

1.1 Historical Overview of Video Compression

Uncompressed video sequences captured from a real-world scene exhibit a high correlation amongst pixels. Intuitively, the correlation residing in an uncompressed video sequence should be removed in order to represent the uncompressed video with the aid of less bits, yet without any substantial loss of the perceived visual quality. Video compression will reduce the size of storage in a hard-drive for example, or the transmission bandwidth and the transmission power required for distributing the video. Hence, a number of video compression standards [1] have been designed during the past decades, such as H.120, H.261, H.263 [2,3], H.264/AVC, MPEG-2 and MPEG-4 for the sake of achieving a high compression ratio.

1.1.1 Brief History of Video Compression

The timeline of the video coding standards is shown in Figure 1.1. The first video recorder was invented as early as in the 1950s.

H.120 (1984-1988)

The International Telecommunication Union (ITU) standardized the recommendation H.120 [4], which is the first digital video coding standard. The techniques employed in H.120 include differential pulse code modulation (DPCM), scalar quantization of pixels and variable-length coding (VLC) of the transform domain coefficients. Although H.120 video failed to achieve adequate quality for practical video compression, it provided important knowledge for its successors.

H.261 (1988-1993)

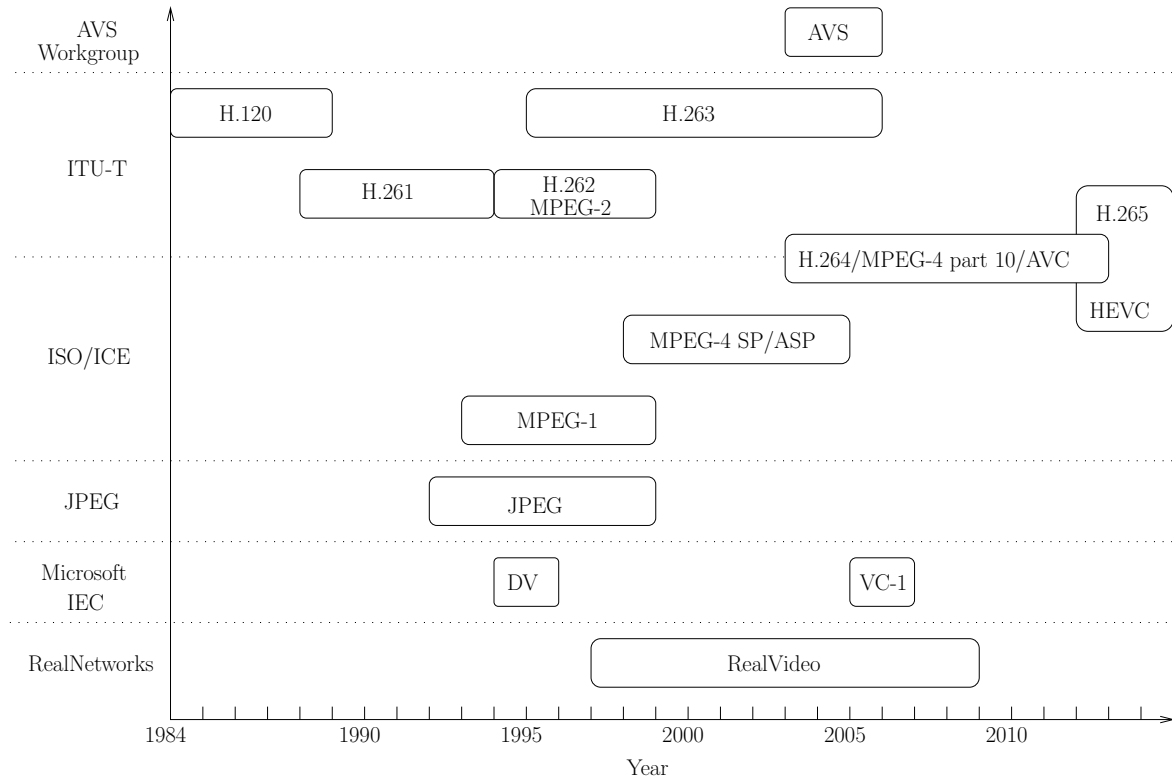


Figure 1.1: Timeline of the video compression standards.

The ITU H.261 standard [5] pioneered the era of practical digital video compression techniques during the 1990s, which was designed for transmitting 352×288 -pixel Common Intermediate Format [5] (CIF) and for 176×144 -pixel Quarter Common Intermediate Format [5] (QCIF) video clips over Integrated Services Digital Networks (ISDN). This standard employed a hybrid video coding scheme, which formed the basis of all state-of-the-art video coding standards.

Joint Photographic Experts Group (1992-1998+)

Motion-JPEG employs the image compression standard of the Joint Photographic Experts Group [6] (JPEG), which was standardized in 1992 for digital still images. Based on JPEG, Motion-JPEG treats a video clip as a sequence of independent images, which are encoded by JPEG separately. Since the temporal correlation is not removed, Motion-JPEG results in a higher bitrate at a lower computational complexity than that of its motion-compensated counterparts. Although Motion-JPEG may be used for digital cameras and video processing systems, it has not been standardized by any organization.

MPEG-1 (1993-1998+)

Then the Moving Picture Experts Group (MPEG) proposed the MPEG-1 standard in 1992 [7] for CIF- or 352×240 -pixel videos. MPEG-1 may be used by almost all

Personal Computers (PCs), Video Compact Disc (VCD) players and Digital Versatile Disc (DVD) players. However, MPEG-1 only supports progressively scanned images, which cannot be used for the interlaced video frames of the National Television System Committee's (NTSC) standard or for the so-called Phase Alternating Line (PAL) video formats [8]. The lack of this compatibility correspondingly prompted the development of the MPEG-2/H.262 standard [9].

MPEG-2/H.262 (1994-1998+)

In 1993, the International Organization for Standardization (ISO) and ITU jointly developed the MPEG-2/H.262 [9] standard for (720×576) - or (720×480) -pixel resolutions, and for high-definition (HD) video with a pixel resolution of 1920×1080 . MPEG-2 is widely used as the format of digital TeleVision (TV) signals, which are broadcast by terrestrial, cable and direct satellite TV systems.

DV (1994-1995+)

The Digital Video (DV) coding specification [10] was standardized by the International Electrotechnical Commission (IEC) in 1994, mainly targeting camera recorders, which encodes a video clip on a frame-by-frame basis, i.e. by dispensing with motion compensation. The DV efficiency is comparable to that of the Intra-frame (I) coding mode of MPEG-2/H.262 and it is better than that of Motion-JPEG.

H.263 (1995-2005)

The H.263 specification was standardized in 1995 by the ITU Telecommunication Standardization Sector [11] (ITU-T), which targeted video conferencing at low bitrates for mobile wireless communications and it is superior to all prior standards in terms of its compression efficiency. The H.263v2 standard, namely H.263+, was completed in 1998, which is the informal anonym for the second edition of the ITU-T H.263, which enhanced the H.263 capabilities by adding several annexes for substantially improving the encoding efficiency. Later, the definition of H.263v3, also known as H.263++, added three further annexes in 2000.

RealVideo (1997-2008+)

RealVideo [12] is a successful proprietary video compression format developed by the RealNetworks company, which was first released in 1997. RealVideo is supported by numerous computing platforms, including Windows, Mac, Linux, Solaris and several mobile phones.

MPEG-4 SP/ASP (1998-2004+)

MPEG-4 standardization [1] was initiated in 1995 and has been continually enhanced by a number of new profiles, including wavelet-based still image coding, scalable coding, 3D images etc. The MPEG-4 video standard strikes adjustable compression quality versus bitrate trade-off, where the so-called Simple Profile (SP) is very similar to H.263, while the Advanced Simple Profile (ASP) further increased the compression efficiency attained.

WMV9/VC-1/SMPTE 421M (2005-2006+)

The society of Motion Picture and Television Engineers (SMPTE) 421M [13] developed a scheme, which is, also known as Windows Media Video version 9 (WMV9) or VC-1. This was initially developed as a proprietary video format by Microsoft, but it was then released as a SMPTE video codec standard in 2006. VC-1 was designed as an alternative to the latest ITU-T and MPEG video codec H.264/MPEG-4 AVC. It was shown that the VC-1 compression efficiency is lower than that of H.264, while imposing a reduced computational complexity.

H.264/MPEG-4 part 10/AVC (2003-2012)

The H.264 standard [14], also known as MPEG-4 part 10 or Advanced Video Coding (AVC), was completed in May 2003, albeit its research continued by adding more extensions. H.264/MPEG-4 part 10/AVC was jointly developed by the ITU-T Video Coding Experts Group (VCEG) and the ISO/IEC JTC1 MPEG, which became one of the most commonly used formats of video recording, compression and distribution. The design goal was to halve the bitrate required by the previous video standards, while retaining the same video quality. A major recent extension of H.264 was the Scalable Video Coding [15] (SVC) scheme completed in 2007, which is specified in Annex G, allowing the construction of bitstreams that contain sub-bitstreams of the H.264 standard. Another major extension of H.264 was the Multiview Video Coding scheme (MVC) completed in 2009 [16], which is specified in Annex H, enabling the construction of bitstreams that represent more than one view of a video scene. The MVC extension contains two profiles, which are the Multiview High Profile [14] representing an arbitrary number of views and the Stereo High Profile [14] for stereoscopic video.

Audio and Video Coding Standard (2003-2005)

The Advanced Video Standard (AVS) [17] was initiated by the government of China for replacing MPEG-2, which has been standardized in 2005. The AVS coding efficiency is comparable to that of the H.264/AVC scheme, which is achieved at a lower computational complexity.

Year	Author(s)	Contribution
1999	Jiang and Ortega [19]	first decomposed the source input into two subsources using a polyphase transform.
2000	Kim <i>et al.</i> [20]	proposed the 3-D Set Partitioning relying on Hierarchical Trees (SPIHT) [21] algorithm for MDC.
2005	Franchi <i>et al.</i> [22]	proposed a spatial-domain down-sampling technique for MDC.
2008	Biswas <i>et al.</i> [23]	proposed a modified tree structure for 3D-SPIHT, where a branch-pruning complexity-reduction technique was employed.
2009	Karim <i>et al.</i> [24]	designed an odd-even frame separation MDC codec for stereoscopic video communications.
2010	Ramasubramanian <i>et al.</i> [25]	combined practical network coding and MDC for video multicast.
	Milani and Calvagno [26]	proposed a multiple description distributed video coder.
2011	Chunyu <i>et al.</i> [27]	proposed a MDC scheme for inserting and controlling the redundancy at MB level and determined the relative importance of each MB by analyzing the potential error propagation paths induced by channel errors.
2012	Kamnoonwattana <i>et al.</i> [28]	investigated a redundancy optimization strategy, equipping the encoder with the full flexibility of choosing the most appropriate quantization parameters (QPs).
	Tsai <i>et al.</i> [29]	proposed a MDC model based on hierarchical B-frames.

Table 1.1: Major contributions on multiple description coding.

HEVC/MPEG-H Part 2/H.265 (2012-2013+)

High Efficiency Video Coding (HEVC) [18] is undergoing development as a successor to H.264/AVC, which is being jointly developed by the ISO/IEC MPEG and ITU-T VCEG as ISO/IEC 23008-2 MPEG-H Part 2 and ITU-T H.265. HEVC aims for halving the bitrate of the H.264/AVC standard at the same video quality and it aims for supporting Ultra High Definition (UHD) videos at a resolution of 8192×4320 . The emerging H.265 will continue to support SVC, which has not been widely adopted by the industry at the time of writing.

1.1.2 Multiple Description Coding

Multiple description coding (MDC) [30, 31] was proposed for overcoming the effects of channel impairments inflicting packet loss events in applications, where the employment of classic forward error correction (FEC) becomes inefficient. Hence MDC is particularly beneficial in the Internet, since retransmissions cannot be readily used in real-time interactive telephony [32]. In MDC, the source is encoded into multiple descriptions, which may be transmitted to the receiver via multiple TCP-IP routes. When all the descriptions have been received, the receiver becomes capable of reconstructing a high quality replica of the source signal. When some of the descriptions are lost due to network congestion, the receiver still remains capable of reconstructing an acceptable quality of the source signal. Furthermore, MDC may also be potentially employed in wireless applications, such as video transmissions in cooperative networks [33] using relays for providing a diversity gain in order to combat channel errors.

Since multiview video communications facilitate the selection of several camera views of a given scene, it may be deemed to be a promising technique also for mobile multiview television. Multiview video coding has been developed for more than 20 years [34, 35] and a number of coding standards appeared, for example the multiview profile based on MPEG-2 [36]. Recently, the Joint Video Team (JVT) proposed multiview video coding as an amendment to H.264/AVC [14]. Similar to conventional single-view video, multiview video also suffers from packet loss events imposed by network congestion. Hence it is necessary to design techniques for multiview video communications in order to combat the packet loss events. Again, MDC [30, 31] has been introduced to overcome the deleterious effects of channel errors with the aid of source encoding diversity. More explicitly, the source may be encoded into several correlated representations/descriptions. Any subset of these descriptions may be independently decoded at the receiver. MDC may be deemed to be an attractive solution for multiview video streaming over unreliable networks, such as the Internet. However, there is a paucity of solutions on the topic. Before illustrating the proposed scheme, let us continue by reviewing the existing MDC solutions.

The authors of [19] proposed a MDC system, where the input source signal is first decomposed into two subsources using a polyphase transform. Each of the subsources is quantized independently by a quantizer. Then each of the descriptions is multiplexed with the coarsely quantized version of the other subsource. Hence each description carries information about the other one, which may be used to combat packet loss events at the receiver. However, this method may become excessively

complex, when more descriptions are required and it is not readily compatible with standardized video codecs, such as MPEG-4, H.264 etc. [1].

Set Partitioning relying on Hierarchical Trees (SPIHT) [21] constitutes an image coding technique that is based on the embedded zerotree wavelet (EZW) algorithm, which organizes the wavelet coefficients in a so-called impartial orientation tree structure. In [20], Kim *et al.* proposed a 3-D SPIHT (3D-SPIHT) algorithm for multiple description coding. Then in [23], the authors propose a modified tree structure for 3D-SPIHT that is more efficient for employment in MDC. They also designed a branch-pruning technique for generating multiple descriptions. However, this technique cannot be readily combined with multiview video codecs.

An odd-even frame separation based MDC codec designed for stereoscopic video communications was proposed in [24], where the odd and even indexed frames were encoded separately into two streams. At the receiver, each frame may be predicted by interpolation techniques applied in case of packet-loss events. Due to the time-lag amongst frames, the odd-even frame-index based methods may be viewed as employing temporal down-sampling. In contrast to the above-mentioned time-domain down-sampling, a spatial-domain down-sampling technique was proposed in [22]. The authors proposed two MDC schemes, namely a so-called drift-compensation based multiple description video codec (DC-MDVC) and independent flow-based multiple description video codec (IF-MDVC). The DC-MDVC was restricted to the employment of only two descriptions, while the IF-MDVC philosophy was more flexible.

The combination of practical network coding and multiple description coding was investigated in [25] for video multicast in lossless networks, where the users receive video streams corresponding to the specific video quality of their specific hardware/-software resources. A multiple description based distributed video codec (MDDVC) was proposed in [26], which generates a set of “lossy” syndromes for each pixel block and creates different descriptions, which is achieved by multiplexing the primary and the redundant video packets. As a further advance, a MDC scheme was proposed in [27] for controlling the amount of redundancy inserted at the macro block (MB) level, which determines the relative importance of each MB by analyzing the potential error propagation paths in the presence of channel-induced errors. A redundancy optimization strategy was investigated in [28], which equips the encoder with the full flexibility of choosing the most appropriate quantization parameters (QPs) without any intervention of the users. A sophisticated MDC model based on hierarchical Bidirectional-predicted frames (B-frame) was proposed in [29], which strikes a trade-off between the attainable coding efficiency and error resilience. Specifically, this model produces a pair of descriptors by applying different MDC techniques, such

as duplication, spatial splitting and temporal splitting of different frames of a video sequence. The major contributions on MDC techniques are listed in Table 1.1.

1.1.3 Standardized Layered Video Coding Techniques

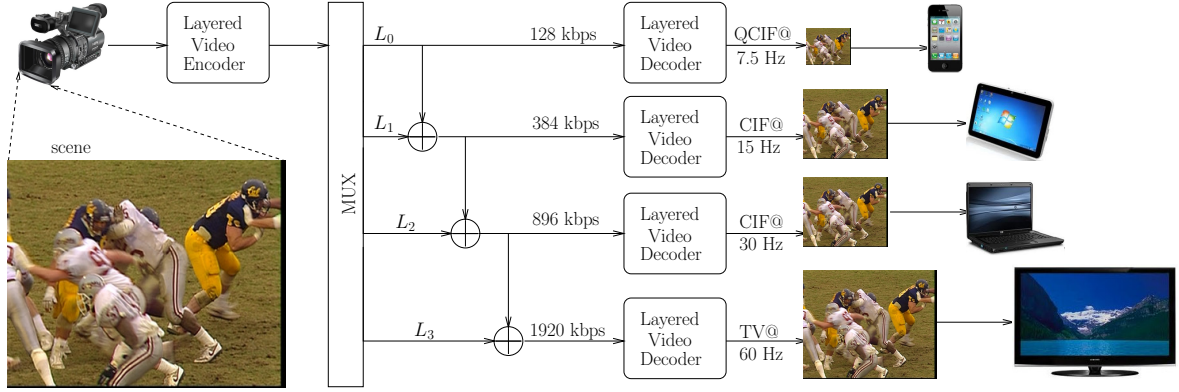


Figure 1.2: Architecture of a layered video scheme [37, 38], where the video quality is refined gradually.

Layered video compression [36, 15, 14, 39] encodes a video sequence into multiple layers, which enables us to progressively refine the reconstructed video quality at the receiver. Generally, the most important layer is referred to as the BL and the less important layers are termed as ELs, which rely on the BL. Furthermore, an EL may be further relied upon by less important ELs. Again, when the BL or an EL is lost or corrupted during its transmission, the dependent layers cannot be utilized by the decoder and must be dropped. A layered video scheme is displayed in Figure 1.2, where the video sequence captured from the scene is encoded into four layers by the layered video encoder, namely $L_0 \sim L_3$, where layer L_i ($0 < i \leq 3$) depends on layer L_{i-1} for decoding, while layer L_i improves the video quality of layer L_{i-1} . In other words, layer L_0 is the BL and layers $L_1 \sim L_3$ are ELs depending on the BL. Furthermore, as shown in Figure 1.2, the ELs L_2 and L_3 rely on the EL L_1 . In other words, if layer L_1 is corrupted, then layers L_2 and L_3 are dropped by the decoder. Given only the layer L_0 having a bitrate of 128 kbps per second (kbps), the corresponding layered video decoder of Figure 1.2 reconstructs the video with a resolution of quarter common intermediate format (QCIF) at 7.5 FPS, while a common intermediate format (CIF) based video sequence at 30 FPS can be reconstructed with the aid of layers L_0 , L_1 and L_2 , which require bitrates of 128 kbps, 256 kbps and 512 kbps, respectively. If the TV screen of Figure 1.2 is utilized by the user, all four layers $L_0 \sim L_3$ may also be streamed for achieving the highest video quality. In practice, the different video streaming scenarios of Figure 1.2 require different bandwidth and hence achieve different visual quality. The users may rely on different video screens,

such as those of mobile phones, tablet PCs, PC and TV screen, as seen in Figure 1.2 for example.

A multiview profile (MVP) [36] was developed by the moving picture expert group (MPEG)'s [1] video coding standard, where the left view and right view were encoded into a BL and an EL, respectively.

The subject of SVC [15] has been an active research field for over two decades. This terminology is also used in the Annex G extension of the H.264/AVC video compression standard [14]. Indeed, SVC is capable of generating several bitstreams that may be decoded at a similar quality and compression ratio to that of the existing H.264/AVC codec. When for example low-cost, low-quality streaming is required by the users, some of the ELs may be removed from the compressed video stream, which facilitates flexible bitrate-control based on the specific preferences of the users. A H.264 scalable video stream contains a sequence of network abstraction layer units (NALUs) [14], which consist of a header and a payload. The header contains the information about the type of NALU and its function in the video reconstruction process, while the payload carries the compressed signals of a video frame. The parameters *dependency_id* (DID), *temporal_id* (TID) and *quality_id* (QID) contained in the NALU header describe the scalability feature of the bitstream. Specifically, DID, TID and QID represent Coarse Grain Scalability (CGS), Temporal Scalability and Medium Grain Scalability (MGS) [15], respectively. The CGS feature facilitates the coarse adaption of video properties, such as the spatial resolution of the video, reconfiguring from QCIF to CIF, where the video can be encoded into a set of coarse enhanced sub-streams referred to as dependency-layers. The DID parameter represents the dependency-layer the current NALU belongs to. The decoding of a NALU with $DID > 0$ depends on the NALUs associated with $(DID - 1)$, but with the same TID and QID values. Based on this dependency rule, the video quality may be readily reduced by removing the NALUs with a DID larger than a specific DID parameter. Similar dependency rules exist for the temporal scalability and MGS features.

Recently, the Joint Video Team (JVT) proposed multi-view video coding (MVC) as an amendment to the H.264/AVC standard [14]. Apart from the classic techniques employed in single-view coding, multi-view video coding invokes the so-called inter-view correction technique by jointly processing the different views for the sake of reducing the bitrate. Hence, the first encoded view may be termed as the BL, while the remaining views may be treated as the ELs.

A number of layered video coding schemes have been developed and some of

them are adopted by recent video coding standards, for example the scalable video coding [15] and data partitioning (DP) [40, 14, 41]. In this treatise, we use the data partitioning based layered video coding in our simulations, where data partitioning is a beneficial feature of the H.264/AVC codec [14]. In the data partitioning mode, the data streams representing different semantic importance are categorized into a maximum of three bitstreams/partitions [42] per video slice, namely type A, type B and type C partition. The header information, such as macroblock (MB) types, quantization parameters and motion vectors are carried by the type A partition. The type B partition is also referred to as the intra-partition, which contains intra-frame-coded information, including the coded block patterns (CPBs) and intra-coded coefficients. The type B partition is capable of prohibiting error propagation in the scenario, when the reference frame of the current frame is corrupted. In contrast to the type B partition, the type C partition is the inter-partition, which carries the inter-CBPs and the inter-frame coded coefficients. The type C partition has to rely on the reference frame for reconstructing the current picture. Hence, if the reference picture is corrupted, errors may be propagated to the current frame. Amongst these three partitions, the type A partition may be deemed to be the most important one, which may be treated as the base layer. Correspondingly, the type B and C partitions may be interpreted as a pair of enhancement layers, since they are dependent on the type A partition for decoding. Albeit the information in partition B and C cannot be used in the absence of partition A, partition B and partition C can be used independently of each other, given the availability of partition A. In this treatise, we will employ the partitioning mode of H.264/AVC for benchmarking our system.

1.1.4 Preliminaries for the Evaluation of Video Transmission

This section introduces some preliminaries associated with our video simulations, including the video quality metric, the format of the H.264 bitstream and the evaluation framework of scalable video coding.

1.1.4.1 Video Quality Metric

Here we commence by introducing the most common video quality metric, namely the peak signal-to-noise ratio (PSNR). Let us commence by stipulating the following assumptions

- $(m \times n)$: the resolution of the luminance frame;
- $I(i, j)$: the original/reference video pixel value at position (i, j) of a specific video frame;
- $K(i, j)$: the reconstructed pixel value at position (i, j) of a specific video frame;

Then the PSNR is calculated as follows

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i, j) - K(i, j)]^2$$

$$PSNR_{dB} = 20 \cdot \log_{10} \frac{\max(I)}{\sqrt{MSE}}. \quad (1.1)$$

However, small mean square error (MSE) values may result in infinite PSNR values. Hence the following modified PSNR calculation is employed

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i, j) - K(i, j)]^2$$

$$PSNR_{dB} = 20 \cdot \log_{10} \frac{\max(I)}{\sqrt{MSE'}} \quad (1.2)$$

$$MSE' = \max(1, MSE),$$

which results in a maximum PSNR value of 48.13 dB.

1.1.4.2 Annex B Format of H.264/AVC

Let us continue by providing a brief introduction to the H.264/AVC bitstream format [14]. The basic element of the H.264/AVC bitstream is the Network Abstraction Layer Unit [14] (NALU), which provides a network-friendly video representation addressing both low-latency lip-synchronized conversational and non-conversational applications. In annex B of [14], the byte-stream syntax and the semantics of NALU is specified for the H.320 video telephony applications, which is displayed in Figure 1.3.

As shown in Figure 1.3, each NALU begins with a start prefix code, which is a fixed-sequence of 3 bytes, namely the bytes 0x000001 indicated by “start_code_prefix_one” in Figure 1.3. One or multiple leading zero bytes may precede the start prefix code of a NALU, which are indicated by the “leading_zero” and “zero_byte” in Figure 1.3. Following the start prefix code is the so-called NALU structure and the trailing-zeros bytes. The first byte of the NALU carries the “forbidden_zero_bit”, the “nal_ref_idc” and the “nal_unit_type” parameters, which are 1, 2 and 5 bits in length, respectively, as illustrated in Figure 1.3. The “nal_unit_type” contains 5 bits specifying the type of the Raw Byte Sequence Payload (RBSP) data structure, as detailed in Table 7-1 of [14]. For example, if the slice the partitioning mode is enabled in the H.264/AVC video encoder, then the “nal_unit_type” may have the

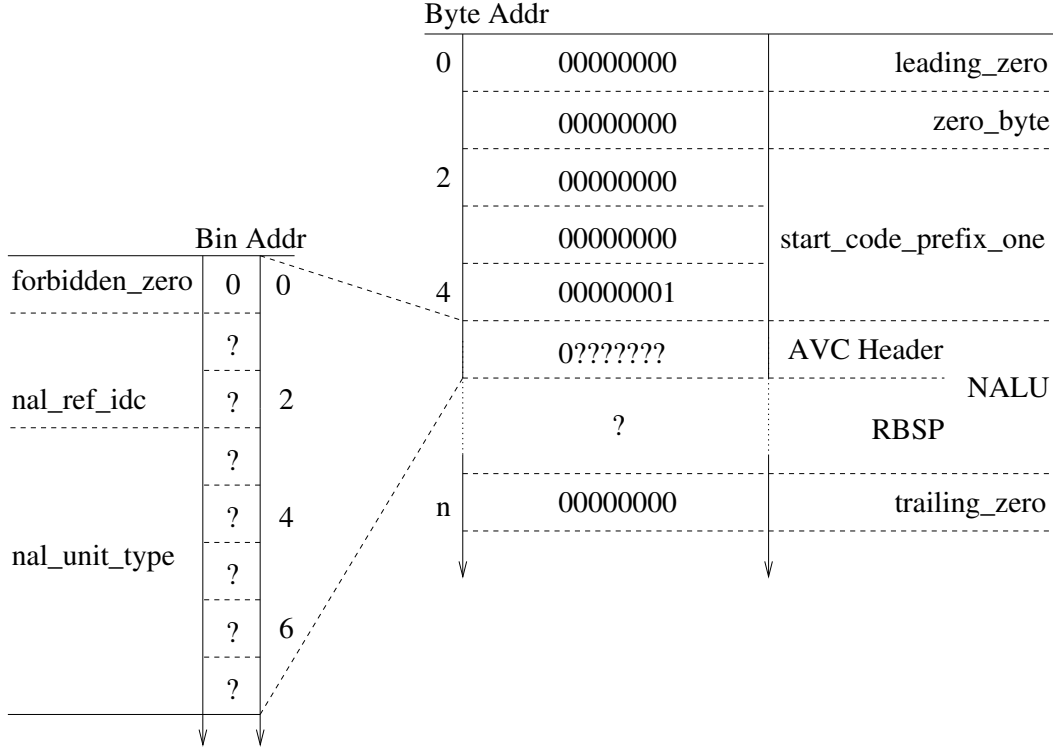


Figure 1.3: Byte stream syntax of the Network Abstraction Layer Unit (NALU) [14], where each NALU consists of an Advanced Video Coding (AVC) header and a Raw Byte Sequence Payload (RBSP).

values of 2, 3, 4, indicating the partition types of A, B, C, respectively. Furthermore, the "C" column of Table 7-1 in [14] indicates the importance of the NALUs.

1.1.4.3 Experimental Evaluation of Scalable Video Coding

1.1.4.3.1 Dependency Rule of Scalable Video Coding Since the scalable video coded (SVC) bitstreams are sensitive to channel impairments, which leads to avalanche-like error propagation, the SVC decoder provided by the Joint Scalable Video Model (JSVM) software does not support the decoding of any error-infested SVC bitstreams. Nonetheless, JSVM provides a number of tools, which may assist us in constructing an experimental evaluation framework for SVC-based video communications [43].

An H.264 SVC stream carries a sequence of NALUs, which is composed of a header and a payload carrying the encoded video frame. More specifically, each NALU may be split into the AVC header and RBSP, as illustrated in Figure 1.3, where RBSP may be further split into the SVC header and payload as detailed in Figure 1.4. Each NALU header, including both the AVC header and SVC header, carries information about the type of data and its specific role in the decoding process. In the information contained by the SVC header of Figure 1.4, we are specifically interested in three parameters, namely in the "dependency_id" (DID), the "temporal_id" (TID) and the

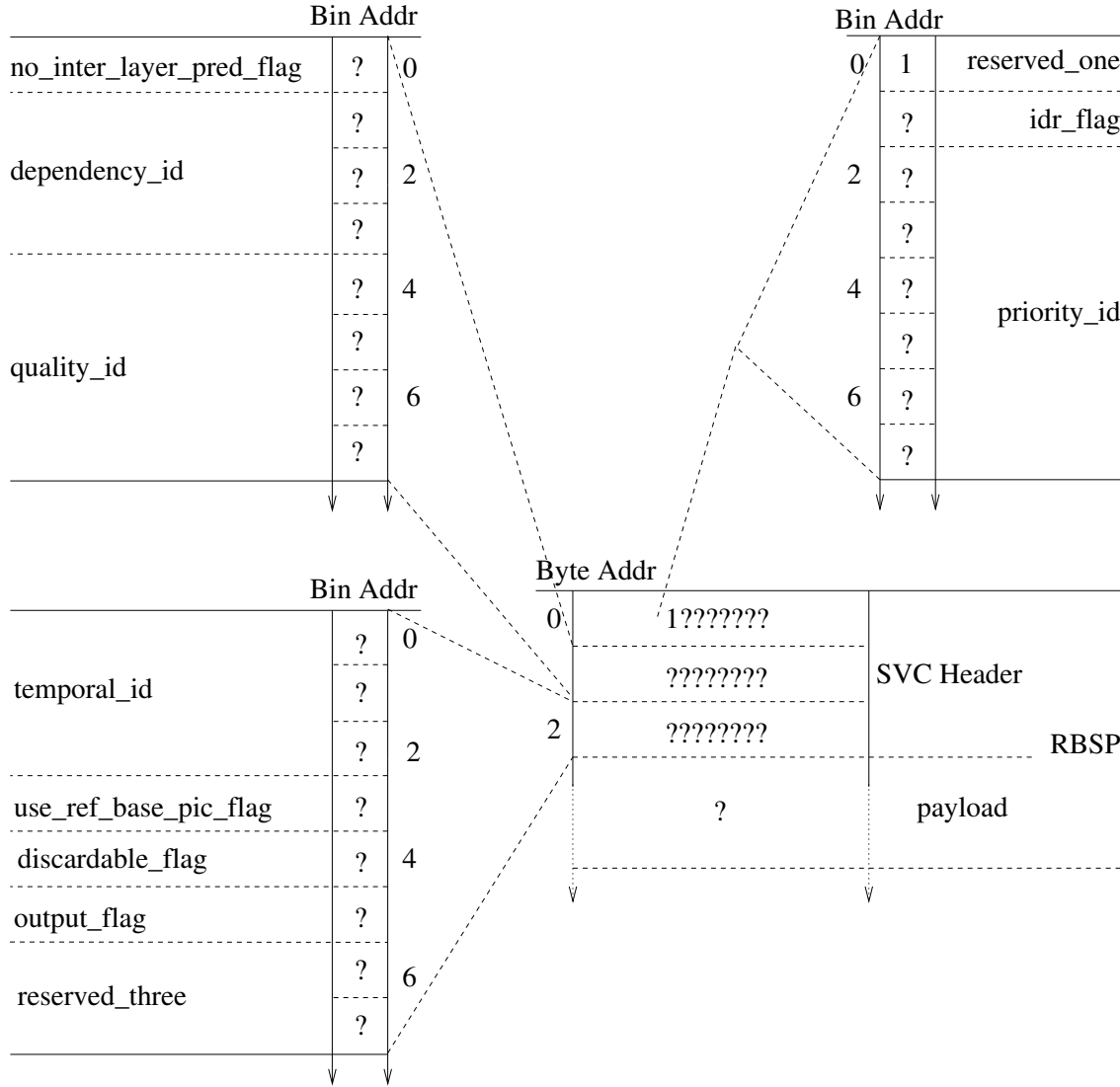


Figure 1.4: Byte stream syntax for the Raw Byte Sequence Payload (RBSP) of Figure 1.3, where the first three bytes constitute the header of Scalable Video Coded (SVC) H.264 video stream.

“quality_id” (QID) of Figure 1.4, each of which determines a particular scalability feature. Specifically, each of the DID, TID and QID parameters may indicate Coarse Grain Scalability (CGS), Temporal Scalability (TS) and Medium Grain Scalability (MGS), respectively.

The CGS feature indicated by the “dependency_id” of Figure 1.4 controls some of the most important coarse video properties, such as the spatial resolution of the video, potentially ranging from CIF to 4CIF. Correspondingly, the video may be encoded into the so-called dependency-layers. The DID parameter distinguishes the specific dependency-layer, which the current NALU belongs to. The decoding of a NALU with $DID > 0$ depends on the NALUs associated with $(DID - 1)$ and with the same TID and QID parameters. Based on this dependency rule, we can coarsely

control the video quality, for example by removing the NALUs associated with a DID higher than a specific value.

Temporal scalability indicated by the “temporal_id” of Figure 1.4 enables the adaption of the video frame-rate. The TID indicates the temporal-layer of the current NALU, i.e. the so-called “frame-rate sub-stream”. A NALU associated with the temporal-layer $TID > 0$ and with $QID = 0$ depends on NALUs of the temporal layer $(TID - 1)$, but it is associated with the same DID as well as QID parameters. Following this rule, a frame-rate scaling may be achieved for example by removing the NALUs having a TID higher than a specific value.

The MGS feature indicated by the “quality_id” of Figure 1.4 (also known as progressive refinement) enables the adaptation of the video quality. The video can be coded by incorporating a set of quality-enhancement sub-streams, also referred to in parlance as quality-layers. Naturally, including more quality-layers reduces the encoder’s quantization error and thus improves the reconstructed video quality. The QID parameter indicates the identifier of the specific quality-layer, which a particular NALU belongs to. A NALU having a quality-layer $QID > 0$ depends on the NALUs associated with the quality layer $(QID - 1)$, but associated with the same DID and TID parameters. Based on this dependency rule, video quality scaling may be readily achieved by removing the particular NALUs having a QID higher than a specific value.

With respect to the above-mentioned temporal and medium grain scalability, the dependency rules may be concluded as follows

$$\begin{aligned} (TID > 0, QID = 0) &\rightarrow (TID - 1, QID = 0) \\ (TID \geq 0, QID > 0) &\rightarrow (TID, QID - 1) \end{aligned} \quad , \quad (1.3)$$

where the arrows represent “depends on”.

1.1.4.3.2 Evaluation Framework The current JSVM software cannot decode the error-infested bitstream. In our experiments, we firstly process the received error-infested bitstream thereby obtaining a “decoder-friendly” bitstream. Then the “decoder-friendly” bitstream will be interpreted by the JSVM for the sake of reconstructing the transmitted video. The SVC evaluation framework is displayed in Figure 1.5. Generally, the tool “BitStreamExtractor¹” is used for generating a

¹This tool is provided by the JSVM software for the sake of extracting a corresponding bitstream given a so-called trace file.

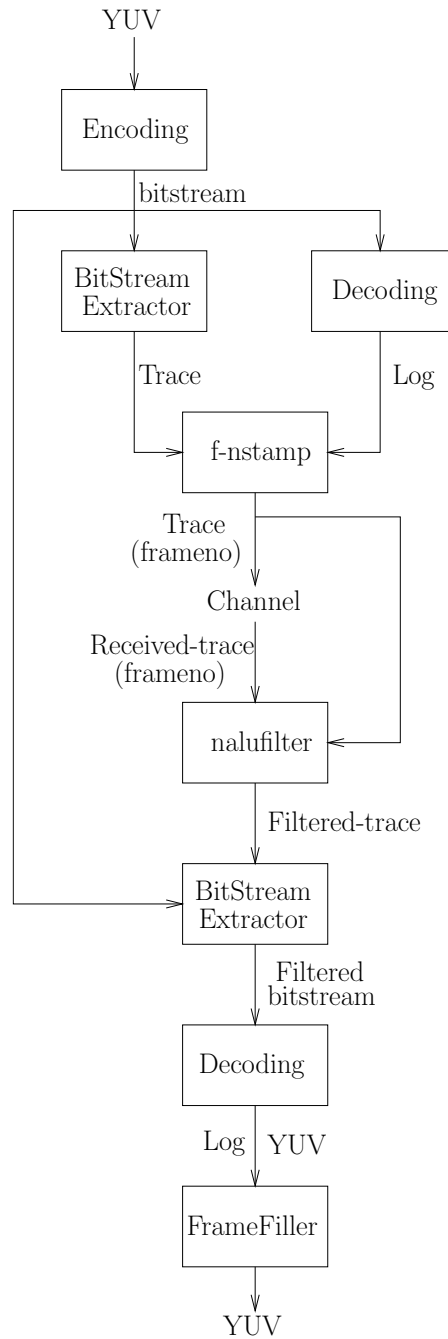


Figure 1.5: Scalable video-streaming evaluation framework (SVEF) [43].

Start-Pos. Truncatable	Length	LId	TId	QId	Packet-Type	Discardable	
0x00000000	156	0	0	0	StreamHeader	No	No
0x0000009c	13	0	0	0	ParameterSet	No	No
0x000000a9	15	0	0	0	ParameterSet	No	No
0x000000b8	16	0	0	0	ParameterSet	No	No
0x000000c8	16	0	0	0	ParameterSet	No	No
0x000000d8	16	0	0	0	ParameterSet	No	No
0x000000e8	16	0	0	0	ParameterSet	No	No
0x000000f8	9	0	0	0	ParameterSet	No	No
0x00000101	9	0	0	0	ParameterSet	No	No
0x0000010a	10	0	0	0	ParameterSet	No	No
0x00000114	10	0	0	0	ParameterSet	No	No
0x0000011e	9	0	0	0	ParameterSet	No	No
0x00000127	10	0	0	0	ParameterSet	No	No
0x00000131	9	0	0	0	SliceData	No	No
0x0000013a	5833	0	0	0	SliceData	No	No
0x00001803	6555	0	0	1	SliceData	Yes	No
0x0000319e	6411	0	0	2	SliceData	Yes	No

Figure 1.6: Exemplified trace file generated by the “BitStreamExtractor” tool, where each line records the parameters of a NALU. The left 5 columns indicate the byte address, size layer id, TID, QID. The column “Packet-Type” indicates the data type of this NALU. The column “Discardable” indicates whether the lack of this NALU corrupts the JSVM decoding process.

JSVM 9.19.15 Decoder

```

new ACCESS UNIT

Frame 0 ( LId 0, TL 0, QL 0, AVC-I, BId -1, AP 0, QP 38 )
Frame 0 ( LId 0, TL 0, QL 1, SVC-I, BId 0, AP 1, QP 34 )
Frame 0 ( LId 0, TL 0, QL 2, SVC-I, BId 1, AP 1, QP 30 )
Frame 0 ( LId 0, TL 0, QL 3, SVC-I, BId 2, AP 0, QP 26 )
Frame 0 ( LId 0, TL 0, QL 4, SVC-I, BId 3, AP 1, QP 22 )
Frame 0 ( LId 0, TL 0, QL 5, SVC-I, BId 4, AP 0, QP 18 )
new ACCESS UNIT
Frame 1 ( LId 0, TL 0, QL 0, AVC-P, BId -1, AP 0, QP 38 )
Frame 1 ( LId 0, TL 0, QL 1, SVC-P, BId 0, AP 1, QP 34 )
Frame 1 ( LId 0, TL 0, QL 2, SVC-P, BId 1, AP 1, QP 30 )
Frame 1 ( LId 0, TL 0, QL 3, SVC-P, BId 2, AP 1, QP 26 )
Frame 1 ( LId 0, TL 0, QL 4, SVC-P, BId 3, AP 1, QP 22 )
Frame 1 ( LId 0, TL 0, QL 5, SVC-P, BId 4, AP 1, QP 18 )
new ACCESS UNIT
Frame 2 ( LId 0, TL 0, QL 0, AVC-P, BId -1, AP 0, QP 38 )
Frame 2 ( LId 0, TL 0, QL 1, SVC-P, BId 0, AP 1, QP 34 )

```

Figure 1.7: Exemplified log file generated by the JSVM decoder, where each line records the parameters of a NALU. The number following the string “Frame” indicates the frame index of current NALU.

decoder-friendly bitstream from a processed trace file, which is exemplified in Figure 1.6. Let us commence by introducing the following files:

- Log: records the parameters of all NALUs within the corresponding bitstream.
- Trace: records the parameters of all NALUs within the original bitstream.
- Received-trace: records the parameters of all NALUs within the received error-free bitstream, where the error-infested NALUs are discarded. This file is used for generating the “Filtered-trace”.
- Filtered-trace: records the parameters of all NALUs within the decoder-friendly bitstream, where the error-infested NALUs and dependency-broken NALUs are discarded. This file may be input to the “BitStreamExtractor” tool for the sake of obtaining the decoder-friendly bitstream.

Below, we detail the generation of the “Filtered-trace” file. At the transmitter, the original video sequence is encoded into a scalable bitstream using the SVC video encoder. The corresponding trace-file may be created by the “BitStreamExtractor” tool. Moreover, the SVC decoder can output the decoder “Log” based on the SVC bitstream input, as exemplified in Figure 1.7. Then the “f-nstamp” block shown in Figure 1.5 may associate the trace-file with the frame index field of the “Log” file of Figure 1.7. At the receiver, the “Received-trace” file associated with the frame index of Figure 1.5 can be created according to the received error-free NALUs. However, some of the NALUs included in the “Received-trace” file of Figure 1.5 may not meet the dependency rules introduced in Section 1.1.4.3.1. Hence the NALUs with broken dependency should be removed from the “Received-trace” file, which is accomplished by the “nalufilter”² of Figure 1.5, resulting in the “Filtered-trace” file. Then the “Filtered-trace” file is used together with the original bitstream available at the receiver for extracting the “Filtered-bitstream” of Figure 1.5 by the “BitStreamExtractor” tool. The “Filtered-bitstream” will be input to the SVC decoder for reconstructing the YUV video file. Finally, the “framefiller” operation will fill in all the missing frames in the reconstructed video file.

1.2 Video Transmission Exploiting Redundancy

The video encoders cannot remove all the present redundancy in the original video signals, such as in layered video streams and in distributed video coding. Hence the residual redundancy should be exploited at the receiver for the sake of achieving an improved video quality similarly to the way we exploit the deliberately introduced

²The “nalufilter” check the dependency chain of each NALU within the input trace file. If the dependency rules of a NALU cannot be satisfied, its corresponding information will be removed from the trace file.

redundancy of FEC. Specifically, both unequal error protection (UEP) techniques [44] and joint source-channel decoding (JSCD) techniques [45] may be employed.

1.2.1 Unequal Error Protection

Table 1.2: Major contributions on unequal error protection for video communications.

Year	Author(s)	Contribution
1967	Masnick and Wolf [46]	first proposed UEP, which allocates stronger FEC to the more important data.
2004	Marx and Farah [47]	minimized the mean distortion by non-uniformly distributing redundancy between the succeeding video frames.
2005	Brüggen and Vary [48]	allocated different transmission power to individual bits according to their bit error sensitivity.
	Wang <i>et al.</i> [49]	conceived an UEP for object-based video communications for achieving the best attainable video quality under specific bitrate and delay constraints.
2006	Pavlushkov <i>et al.</i> [50]	studied the UEP capabilities of CC, while rate-compatible convolutional codes (RCPC) was proposed by Hagenauer [51].
	Kumar and Milenkovic [52]	proposed a new family of UEP codes, based on LDPC [53] component codes, where the order of decoding and the choice of the component codes jointly determine the level of error protection.
2007	Rahnavard <i>et al.</i> [54]	proposed the so-called UEP density evolution (UDE) technique for transmission of video streams over binary erasure channels (BEC).
2008	Aydinlik and Salehi [55]	investigated UEP based turbo coded modulation, where both the channel capacity and the cutoff rates of UEP levels were determined.
	Ha and Yim [56]	proposed the layer-weighted expected zone of error propagation (LW-EZEP), for quantifying the error propagation effects imposed by packet loss events.
	Chang <i>et al.</i> [57]	considered the unequal importance of the frames in a group of picture (GOP), as well as that of the macroblocks in a video frame.
2009	Chang <i>et al.</i> [58]	considered the different importance of the I-frame and of the P-frames within a GOP.
2010	Maani and Katsaggelos [59]	proposed cross-layer operation aided scalable video streaming, which estimated the expected video distortion according to the channel conditions.

	Ahmad <i>et al.</i> [60]	developed Luby Transform (LT) [61] coded UEP for combating the packet loss events imposed by the networks.
2011	Hellge <i>et al.</i> [62]	proposed the layer-aware FEC (LA-FEC) philosophy using a Raptor codec for video transmission over the BEC.
2012	Alajel <i>et al.</i> [63]	proposed an UEP scheme based on HQAM exploiting the unique characteristics of the color plus depth map stereoscopic video.
	Khalek <i>et al.</i> [64]	proposed an APP/MAC/PHY cross-layer architecture for optimizing the perceptual quality of delay-constrained scalable video transmission.
2013	Micallef <i>et al.</i> [65]	conceived UEP for side information (SI) values at different position of the Wyner-Ziv (WZ) frames for the sake of reducing the bitrate.

Unequal error protection (UEP) was firstly proposed by Masnick and Wolf in [66], which allocates stronger forward error correction (FEC) to the more important data, while dedicating weaker FEC to the less important video parameters. Since then numerous UEP techniques have proposed. A novel UEP modulation concept was investigated in [48] for the specific scenarios, where channel coding cannot be employed. Hence UEP was achieved by allocating different transmission power to individual bits according to their bit error sensitivity albeit in practice this remains a challenge. Additionally, the UEP capabilities of convolutional codes (CC) were studied in [50], while rate-compatible convolutional codes (RCPC) were proposed by Hagenauer [51]. Furthermore, as a benefit of the outstanding performance of low-density parity-check (LDPC) codes, a number of UEP design methodologies [67, 52, 68, 54] have been investigated using LDPC codes. The so-called UEP density evolution (UDE) technique of [67, 54] was proposed for transmission of video streams over binary erasure channels (BEC). The authors of [52] proposed a new family of UEP codes, based on LDPC component codes, where the component codes are decoded iteratively in multiple stages, while the order of decoding and the choice of the LDPC component codes jointly determine the level of error protection. A practical UEP scheme using LDPC codes was proposed in [68], where the high-significance bits were more strongly protected than low-significance bits.

As mentioned in Section 1.1.3, when the BL is corrupted or lost due to channel impairments, the ELs must also be dropped by the video decoder, even if they are perfectly received. Moreover, the less important layers have lower priority and hence may be dropped in the transmission scenario of network congestion or buffer

overflow [69]. Hence, it is intuitive to perform UEP for both the BL and the ELs for the sake of improving the system's performance. However, most of the above UEP studies considered artificially generated signals of unequal significance, rather than realistic video signals. Naturally, the significance differentiation of practical video signals is more challenging. In compressed video streams, as in layered video coding, different bits may have different significance. Therefore, again it is intuitive to employ UEP for protecting the more important bits by stronger FEC codecs than the less important bits, in order to achieve an improved reconstructed video quality. Nonetheless, a number of contributions have been made also in the field of UEP video communications relying on realistic video signals. Generally, these UEP techniques [47, 70, 49, 55, 56, 57, 58, 71, 72, 59, 60, 73, 62, 41, 74, 75] may be classified into two categories³, namely the packet-level schemes [49, 56, 71, 72, 59, 60, 73, 74] and bit-level schemes [47, 70, 55, 57, 58, 62, 41, 75]. Specifically, the packet-level contributions [49, 56, 71, 72, 59, 60, 73, 74] mitigate the packet loss events as exemplified by the packets lost in Internet-routers [76], while the bit-level ones are devoted to eliminating bit-errors of wireless scenarios [77]. Let us continue by considering the family of packet-level UEP schemes first [49, 56, 59, 60]. An UEP scheme was conceived for object-based video communications in [49] for achieving the best attainable system performance under specific bitrate and delay constraints in an error-prone network environment. A novel UEP method using Reed-Solomon codes was proposed in [56] for SVC video transmission over networks inflicting packet-loss events. Firstly, the layer-weighted expected zone of error propagation (LW-EZEP) was defined as an efficient performance metric for quantifying the error propagation effects imposed by packet loss events. Then, the corresponding Reed-Solomon (RS) coding rates were assigned based on LW-EZEP for minimizing the expected video distortion. Cross-layer operation aided scalable video streaming designed for error-prone channels was investigated in [59], where the RS coded UEP was optimized for robust video delivery. The expected video quality was evaluated based on both the available bandwidth and the packet loss ratio (PLR) encountered, which was then further improved by employing content-aware bitrate allocation. Finally, an error concealment method was invoked at the receiver. The authors of [60] studied an UEP scheme using Luby Transform (LT) codes for recovering the video packets lost owing to network congestions. The above-mentioned UEP methodologies tended to employ hard decoded FEC [78] codes and dispensed with soft-information exchange in the decoding process, since they are carried out at the application layer. Similarly, bit-level UEP techniques have also been widely investigated [47, 70, 55, 57, 58, 41]. The authors of [47] minimized the

³Thanks to an anonymous reviewer for this valuable suggestion.

mean distortion by non-uniformly distributing the redundancy imposed by the turbo code between the successive video frames, where the H.263 video codec was employed. In [70], an UEP scheme using a turbo transceiver was optimized for wireless video telephony. The UEP performance of data-partitioned [14] H.264/AVC video streaming systems using recursive systematic convolutional (RSC) codes was evaluated in [41], while turbo coded modulation [79] based UEP was investigated in [55], where both the cutoff rates and the channel capacity of each of the UEP levels was determined. The authors of [57] considered the unequal importance of both the video-frames in a group of pictures (GOP) and the significance of the diverse macroblocks (MB) in a video frame for transmission over wireless channels, where a prompt and efficient fast rate allocation scheme was also investigated. However, only three protection classes were discussed in [57], which limits the attainable system performance. The same authors also considered the different importance of the intra-coded frame (I-frame) and of the predicted frames (P-frame) within a GOP in [58], where the video bits of different importance were mapped to the different-integrity modulation constellation points of hierarchical quadrature amplitude modulation (QAM). An UEP scheme based on hierarchical quadrature amplitude modulation (HQAM) was proposed for 3D video transmission in [63], which exploits the unique characteristics of the color plus depth map based stereoscopic video representation. An APP/MAC/PHY cross-layer architecture was proposed in [64] that optimizes the perceptual quality of delay-constrained scalable video transmission. Specifically, an online Quality of Service-to-Quality of Experience (QoS-to-QoE) mapping technique was employed for quantifying the effects of packet loss events endured by each video layer using the ACK history and perceptual metrics. At the PHY layer, the authors developed a link adaptation technique relying on the QoS-to-QoE mapping for providing perceptually-optimized UEP. At the APP layer, the source rate was adapted by selecting the specific set of temporal- and quality-layers to be transmitted based on the prevalent channel statistics, on the source rates and on the playback buffer state. The authors of [65] show that the side information (SI) values within different positions of the Wyner-Ziv (WZ) frames have different error probability. Hence UEP of these non-uniformly distributed SI values was employed for the sake of reducing the required bitrate. The major contributions to UEP techniques conceived for video communications are summarized in Table 1.2. These bit-level schemes tended to employ physical layer FEC codes [80] and perform soft decoding [78] for wireless video communications.

In the traditional UEP scheme conceived for layered video communication, variable-rate FEC codes were assigned to the different-sensitivity layers for improving the

reconstructed video quality. However, when the BL is corrupted or lost, the ELs have to be discarded by the video decoder, regardless whether they are perfectly decoded or not, which implies that both the transmission power and the bandwidth assigned to the ELs is wasted. Hence it is beneficial improve the protection of the more important BL with the aid of the ELs. Hence, the authors of [72] proposed a packet-level solution based on fountain codes [71] for scalable video multicast over networks inflicting packet loss events, where ELs conveyed parity information protecting the more important BL. By contrast, hierarchical network coding [73] and generalized network coding [74] techniques were also proposed for combating packet-loss events, when transmitting information of unequal importance. Furthermore, a bit-level layer-aware FEC (LA-FEC) philosophy using a hard-decoded Raptor code was designed for scalable video transmission over the binary erasure channels (BEC) in [81, 62]. The Raptor encoder generated the parity bits right across the BL and the ELs at the transmitter. As a benefit, the parity bits of the ELs may be utilized for assisting in correcting the errors residing in the BL at the receiver.

1.2.2 Joint Source-Channel Decoding

Shannon's separation theorem [82] states that reliable transmission can be accomplished by separate source coding using lossless entropy codes and channel coding under the idealized assumption of Gaussian channels and potentially infinite encoding/decoding delay and complexity. However, a finite-delay, finite-complexity source encoder fails to remove all the redundancy residing in the source signals, such as audio and video. Hence, joint source-channel coding (JSCC) [95] was proposed for additionally exploiting the source statistics for the sake of improved performance. Furthermore, Görtz [45, 83] proposed the iterative source-channel decoding (ISCD) philosophy, which performed turbo-like [96, 97] iterative decoding by exchanging extrinsic information between the source encoder and decoder. Softbit source decoding (SBSD) [84] was proposed for error concealment of speech signals by Fingscheidt and Vary using softbit information, where the correlation of adjacent speech source samples were modeled by a first-order Markov process. Afterwards, Adrat and Vary [98, 87] developed SBSB for iterative source-channel decoding using both forward- and backward-oriented calculations.

Similar to speech, joint source-channel decoding of image and video signals also attracted substantial attention. For example, Kliewer, Görtz and Mertins [85, 86] proposed an ISCD scheme for images modelled by a Markov Random Field (MRF) by exploiting that any pixel can exchange extrinsic information with its eight neighbors for exploiting the residual spatial correlations residing in the encoded image. In [88],

Year	Author(s)	Contribution
1948	Shannon [82]	states the source-channel separation theory.
2001	Görtz [83]	proposed the ISCD using a turbo-like decoding philosophy.
	Fingscheidt and Vary [84]	proposed the Softbit Source Decoding (SBSD) for error concealment of speech signals.
2004	Kliewer <i>et al.</i> [85, 86]	proposed an ISCD scheme for images modelled by a Markov Random Field (MRF).
2005	Adrat and Vary [87]	developed SBSB for ISCD using both forward- and backward-oriented calculations.
	Kliewer [88]	proposed a novel symbol-based soft-input APosterior Probability (APP) decoder for Variable Length Coding (VLC) [89] coded source signals.
2008	Zhang <i>et al.</i> [90]	proposed an error-resilient JSCC scheme using an Slepian-Wolf (SW) [91] codec.
2009	Nasruminallah and Hanzo [40]	proposed an ISCD scheme for H.264 [14] video using SBC [92] and SBSB.
2010	Fresia <i>et al.</i> [93]	proposed a JSCC scheme using two concatenated LDPC codes, namely the source LDPC and channel LDPC, respectively.
2012	Ji <i>et al.</i> [94]	proposed a framework for broadcasting scalable video to mobile users having heterogeneous QoS.

Table 1.3: Major contributions on ISCD for video communications.

Kliewer proposed a novel symbol-based soft-input *a-posteriori* probability (APP) decoder for packetized variable length coded (VLC) [89] source signals transmitted over wireless channels, where the Markov-modelled residual redundancy generated after source encoding was exploited for achieving an improved error protection. In the context of distributed source coding, the authors of [90] proposed an error-resilient JSCC scheme using a Slepian-Wolf (SW) codec, which exploited the knowledge of both the channel statistics and the correlation between specific video frames and their reference frames. In [40], an H.264 [14] video telephone scheme was proposed using ISCD. Firstly, the H.264 bitstream was segmented into the partitions A, B and C [14]. Then the three partitions were encoded by variable code-rate short block codes (SBC), which artificially imposed redundancy on the H.264-encoded bitstreams. This artificial redundancy was then exploited by the softbit source decoder, which performs

iterative decoding by exchanging extrinsic information with the channel decoder. In [93], a double low-density parity-check (LDPC) code was proposed for joint source and channel coding, where two concatenated LDPC codes were employed as the source LDPC and channel LDPC, respectively. At the receiver, the source LDPC and channel LDPC performed joint decoding by exchanging extrinsic information. A framework was presented in [94] for broadcasting scalable video to mobile users having heterogeneous QoS requirements, which includes JSCC for the sake of achieving maximum overall quality. The major contributions on ISCD are listed in Table 1.3..

1.2.3 Distributed Video Coding

Below, we firstly detail the basics of Wyner-Ziv coding theory in Section 1.2.3.1. Then we review the distributed video coding principle, which was developed from the Wyner-Ziv theorem.

1.2.3.1 Wyner-Ziv Coding Theory

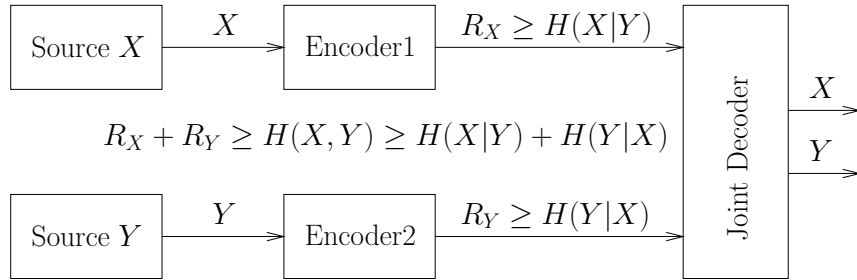


Figure 1.8: Slepian-Wolf coding of two correlated sources X and Y .

Let us assume that X and Y are two independently and identically distributed (i.i.d.) binary sources. In traditional lossless coding schemes, the sources X and Y may be jointly encoded at the transmitter at a rate of $H(X, Y)$, namely the joint entropy of X and Y , where the receiver can reconstruct both X and Y perfectly. As exemplified in Figure 1.9, the left and right views of a stereoscopic video scene are jointly encoded by the “Stereoscopic Encoder” resulting in a joint source s , which will be transmitted to the “Stereoscopic Decoder” block of Figure 1.9 at a bitrate of 1920 kbps. Then the left and view can be reconstructed and output to the 3D display of Figure 1.9 at the receiver. Generally, more computations are carried out by the “Stereoscopic Encoder” than by the “Stereoscopic Decoder” in the scenario of Figure 1.9. However, the question arises whether it is possible to encode X and Y independently, when they are at physically separate locations and still recover both X and Y even if we receive less than $H(X) + H(Y)$ amount of information at the destination? To answer this question, the Slepian-Wolf theorem [91] states that it is indeed possible to encode X and Y independently at the transmitter, while

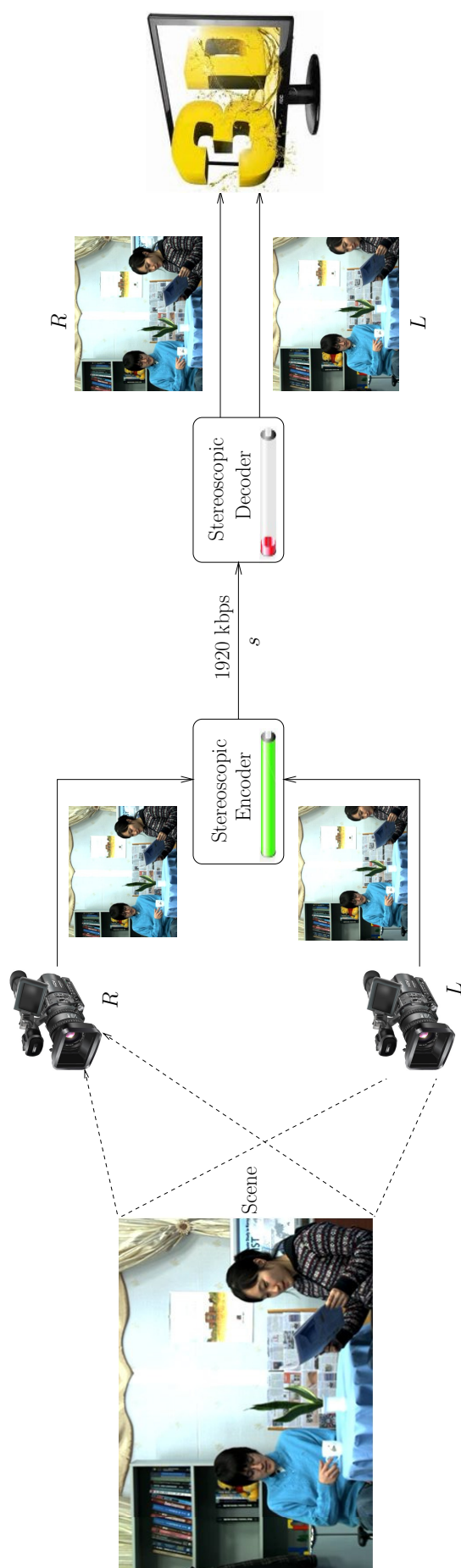


Figure 1.9: Example of traditional stereoscopic video coding, where more power/computation is consumed by the “Stereoscopic Encoder” than the “Stereoscopic Decoder”.

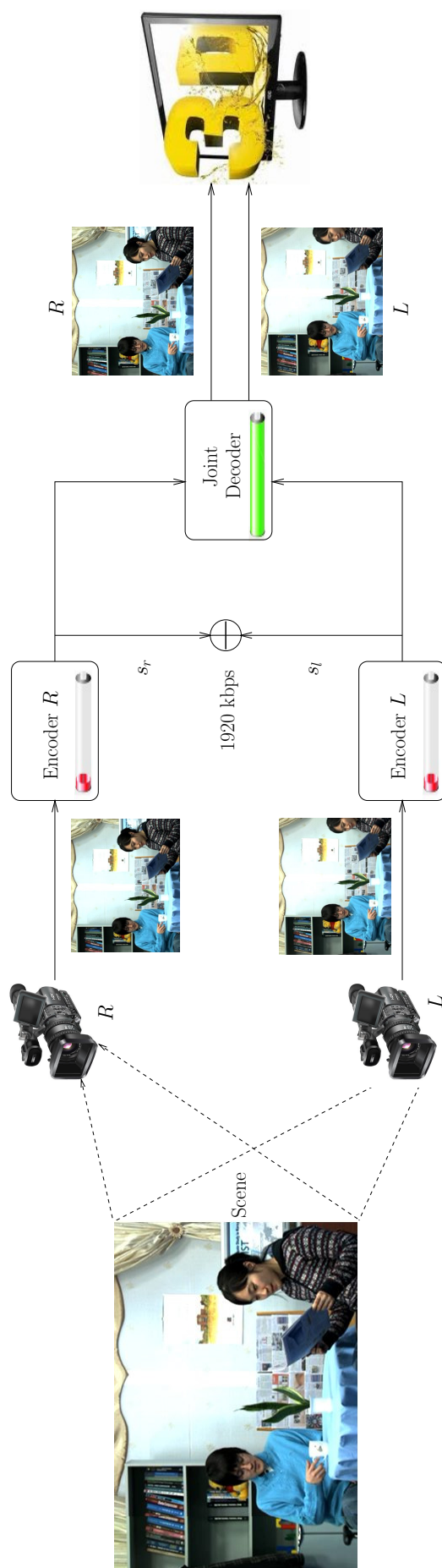


Figure 1.10: Example of Slepian-Wolf theorem based stereoscopic video coding, where more power/computation is consumed by the “Joint Decoder” than the “Encoder L ” and “Encoder R ”.

the receiver is capable of jointly recovering them at the receiver with an amount of $H(X, Y)$ information received from the transmitter. As exemplified in Figure 1.10, the left and right views of a stereoscopic video scene are independently encoded by the “Encoder L” and “Encoder R” blocks resulting in the source signals s_l and s_r , respectively, which will be transmitted to the “Joint Decoder” block of Figure 1.10 at a bitrate of 1920 kbps. Then the left and right view can be reconstructed and output to the 3D display of Figure 1.10 at the receiver. Generally, more computations are carried out by the the “Joint Decoder” than by the “Encoder L” and “Encoder R” in Figure 1.10. The distributed coding structure of the sources X and Y using SW theory is illustrated in Figure 1.8, where R_X and R_Y represents the SW encoding rates for X and Y , respectively. According to Shannon’s source coding theorem [82], we have the transmission rate limits of $R_X \geq H(X|Y)$ and $R_Y \geq H(Y|X)$ for the sources X and Y , respectively, which have to be satisfied for the sake of perfectly recovering them at the receiver. Again, the SW theorem proves that we can afford reducing the rate $(R_X + R_Y)$ to $H(X, Y)$ by jointly decoding X and Y at the receiver, where R_X and R_Y are the rates required for compressing the physically separated sources X and Y individually. Again, when considering the examples of Figures 1.9 and 1.10, the SW theorem states that the both mechanisms require the same bitrate for representing this stereoscopic video. The “Encoder L” and “Encoder R” of Figure 1.10 are referred to as a Slepian-Wolf encoder in the literature, while the “Joint Decoder” of Figure 1.10 is referred to as a Slepian-Wolf decoder.

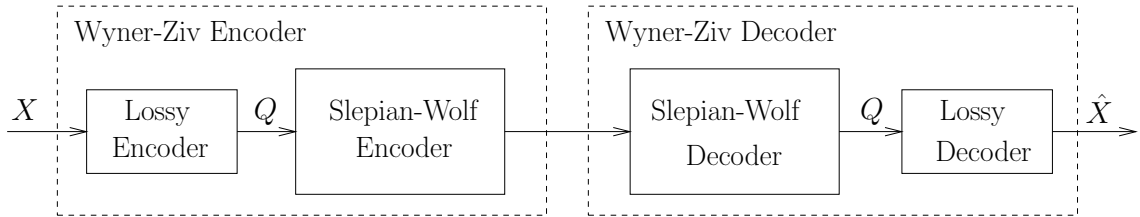


Figure 1.11: Wyner-Ziv coding structure

Then Wyner and Ziv (WZ) [99] further extended the SW theory to lossy - rather than lossless - coding scenarios. Again, when considering the examples of Figures 1.9 and 1.10, the WZ theorem proves that the both mechanisms require the same bitrate for maintaining the same stereoscopic video quality, when “Encoder L” and “Encoder R” of Figure 1.10 perform lossy video encoding. Correspondingly, the lossy version of “Encoder L” and “Encoder R” of Figure 1.10 are referred to as a WZ encoder in the literature, while the “Joint Decoder” of Figure 1.10 is referred to as WZ decoder. Generally, a WZ encoder may be thought of as a combination of a lossy encoder followed by the lossless SW encoder, as shown in Figure 1.11. Specifically, the “Lossy Encoder” followed by the “Slepian-Wolf Encoder” is constituted by

a Wyner-Ziv encoder, while the “Slepian-Wolf Decoder” followed by the “Lossy Decoder” is a Wyner-Ziv decoder. A number of lossy encoders have been proposed, such as for example the uniform quantizer of [100], which was designed for pixel-domain WZ (PDWZ) video coding. The authors of [101] proposed a quantizer matrix [102] based lossy decoder for transform-domain WZ (TDWZ) coding. Moreover, the corresponding PDWZ and TDWZ codecs were investigated for video coding in [101, 100]. Furthermore, the family of “request-and-decode” philosophy based turbo or LDPC codecs have also been investigated in [103, 104].

1.2.3.2 Advances in Distributed Video Coding

Table 1.4: Major contributions on distributed video coding.

Year	Author(s)	Contribution
1976	Wyner and Ziv [99]	states the Wyner-Ziv source coding theory.
2002	Aaron, Girod <i>et al.</i> [105, 106, 107]	substantially advanced the state-of-the-art in WZ video coding.
2005	Gehrig and Dragotti [108]	proposed a novel framework for the DMVC based on a tree-structured compression algorithm.
2006	Varodayan <i>et al.</i> [104]	invented the novel rate-adaptive LDPC accumulate (LDPC) and sum-LDPC-accumulate (SLDPCA) codes for distributed source coding.
2007	Weerakkody <i>et al.</i> [109]	proposed a unidirectional DVC architecture dispensing with feedback channel.
	Artigas <i>et al.</i> [101]	introduced the DVC techniques employed in the EU project DISCOVER.
2008	Varodayanand <i>et al.</i> [110]	proposed an unsupervised motion learning technique for MDC.
	Brites and Pereira [100]	proposed a more realistic WZ video coding approach, which performs online estimation of the channel-induced noise (CN) model parameters at the decoder.
	Guo <i>et al.</i> [111]	extended the WZ framework proposed in [107] into distributed MVC.
2010	Yeo and Ramchandran [112]	extended their previous PRISM framework of [113] to distributed MVC.
2011	Brites and Pereira [114]	proposed an efficient encoder rate control (ERC) solution for transform-domain Wyner-Ziv (TDWZ) video coding.

	Zhang <i>et al.</i> [115]	proposed a context-adaptive MRF reconstruction algorithm, which exploits the spatio-temporal correlation by modelling the WZ frames.
2012	Wang <i>et al.</i> [116]	integrated particle filtering (PF) with standard belief-propagation (BP) decoding to estimate the correlation between the source and SI.
2012	Skorupa <i>et al.</i> [117]	proposed motion-compensated extrapolation based SI estimation.
2013	Abou-Elailah <i>et al.</i> [118]	combined the global and local motion compensation at the decoder side.

Video coding has been developed for several decades [1], which tends to impose a high complexity at the encoder and low complexity at the decoder. A number of compression standards have been proposed, such as MPEG2, H.263 [119] and H.264, for example [1]. However, the traditional video coding schemes have to be further developed for emerging applications, such as wireless video sensor networks (WVSN) [120], where both the energy and the computational complexity of the transmitter are constrained. Hence, based on the SW and WZ theorems of Section 1.2.3.1, the concept of Wyner-Ziv (WZ) video coding, also known as distributed video coding (DVC) was proposed [99, 103].

More specifically, two basic types of WZ coding structures [103] have been proposed. In [105, 106, 107], the authors advocated a WZ codec, which is composed of an inner turbo-code-based [121] Slepian-Wolf (SW) codec [91] concatenated with an outer quantization-reconstruction component pair. More specifically, the odd-indexed video frames, namely the so-called key frames are intra-coded, while the even frames, namely the WZ frames are encoded by the WZ codec. At the receiver, the side information of the WZ frames will be estimated from their adjacent key frames for joint inter-frame decoding. However, a specific impediment of this structure is that the turbo decoder has to invoke a "request-and-decode" [105] process for the transmission of the WZ frames' parity bits, which precludes its application in delay-sensitive services. For the sake of avoiding the requirement of a feedback channel, a unidirectional distributed video coding architecture was proposed in [109], where the decoder exploited both the spatial and temporal correlation of the video signals for refining the side information (SI) for iterative decoding. Low density parity check (LDPC) codes were employed for distributed source coding in [104], which were the so-called rate-adaptive LDPC accumulate (LDPCA) codes and the sum-LDPC-accumulate (SLDPCA) codes. It was shown in [104] that the LDPCA codes

are capable of approaching the capacity of a variety of communication channels more closely - including that of the virtual channel in DVC - than the family of turbo codes. Based on the WZ video coding structure of [107], in the European project DIStributed COding for Video sERvices (DISCOVER) [101] the transform-domain of the WZ frames was encoded by the LDPCA code of [104] at the transmitter. A so-called unsupervised motion learning technique was proposed in [110], which estimates the Motion Vectors (MVMVs) of the next video frame during the decoding of the current frame with reference to the previous reconstructed frame. This technique may be readily applied for both pixel-domain and transform-domain coding. The authors of [100] proposed a more realistic WZ video coding approach, which performs online estimation of the channel-induced noise (CN) model parameters at the decoder, which can be invoked for both pixel-domain and for transform domain WZ video codecs. Moreover, three levels of granularity were proposed by Brites and Pereira for pixel-domain WZ (PDWZ) video coding in [100], namely frame-, block- and pixel-level granularity, while both DCT-band and DCT-coefficient level granularity was proposed for transform-domain WZ (TDWZ) video coding. Then in [114] the same authors proposed an efficient encoder rate control (ERC) solution for transform-domain WZ (TDWZ) video coding. As a further advance, a more accurate parity rate estimator (PRE) is employed for more closely estimating the parity rate necessitated. A context-adaptive Markov random field reconstruction algorithm was proposed in [115], which exploits the spatio-temporal correlation by modelling the WZ frames. Apart from the WZ coding architecture mentioned above, another DVC architecture was proposed in [113], which allows the flexible sharing of complexity between the encoder and decoder.

Multi-view Video Coding (MVC) [122] has recently attracted substantial attention in the context of both sophisticated 3D-TV and low-complexity wireless sensor network scenarios. A number of video coding techniques [123, 124, 39] have been developed for MVC, which typically rely on a high-complexity encoder and a low-complexity decoder. A comparative study of different MVC techniques was presented in [125] for quantifying their compression efficiency versus complexity, which demonstrated that most of the gain of MVC achieved compared to the simulcast transmission method is due to the inter-view prediction of anchor frames. Following these advances, the Joint Video Team (JVT) of MPEG and ITU-T selected the AVC-based MVC scheme as reference software, leading to the concept of the joint multi-view video model (JMVM) [126]. However, the constraint of the inter-view prediction based MVC is that all cameras of the MVC must exchange their monoscopic views with each other for inter-view prediction. What is even more

challenging is that the communication between cameras must have a low-latency in delay-sensitive applications. These requirements are unrealistic in many applications, such as WWSN [127] for example, where both the energy and the computational complexity are constrained [128]. Hence it is necessary to shift the computationally complex tasks from the sensor to the base station (BS) or the server of the network, in addition to limiting the data exchange among the sensors. In theory, the DVC-philosophy is capable of assisting the sensors in relocating the computational burden to the BS, whilst simultaneously limiting the data exchange among themselves. In [108], the authors proposed a novel framework for the distributed compression of multi-view images, which was based on a tree-structured compression algorithm that guaranteed an optimal rate-distortion performance for specific video signals. Yeo and Ramchandran extended their previous PRISM framework detailed in [113] into distributed MVC in [129, 112], where the achievable error-resilience was studied in wireless scenarios. Two alternative models were proposed for exploiting the inter-view correlation, namely the view-synthesis-based correlation model and the disparity-based correlation model. The view-synthesis-based correlation model requires at least two other camera views and relies on both disparity estimation and view interpolation, while the disparity-based correlation model requires only a single additional camera view. In [111], the authors extended the WZ framework proposed in [107] into distributed MVC. At the encoder side, a wavelet-based WZ scheme was proposed for compressing each camera's view independently, where all coefficients were organized as proposed in the SPIHT scheme of [130] on a bitplane by bitplane basis. At the decoder side, a flexible prediction technique was proposed for generating the required SI, which jointly exploited both the temporal and inter-view correlations. The common benefit of the frameworks advocated in [111, 112] is that inter-camera communication is completely avoided and the computational complexity of the encoder was shifted to the decoder. Particle filtering [116] (PF) combined with belief propagation (BP) [53] based decoding was integrated by the authors of [116] for creating a joint factor graph in order to estimate the correlation between the source information and the side information. A system based on motion-compensated extrapolation was proposed in [117], which estimates the motion vector field between the two most recently decoded video frames using the Lucas-Kanade algorithm [131]. The motion field obtained was then extrapolated to the current video frame using an extrapolation grid. The concept of global and local motion compensation was conceived in [118], where the parameters of the global motion field were estimated at the encoder and were signaled to the decoder in order to enhance the conventional motion-compensation with the aid of global motion compensation side information. The major contributions on DVC are listed in Table 1.4.

1.3 Extrinsic Information Transfer Charts

Extrinsic information transfer (EXIT) charts [132, 133] were conceived for characterizing the extrinsic information exchange between multiple component decoders within the iteratively decoded FEC codes. Specifically, the extrinsic information generated by the component decoders is quantified by the mutual information (MI) between itself and the related information bits. In the scenarios of two decoder components, the MI outputs of the two decoders are plotted on the horizontal axis and on the vertical axis, respectively. By contrast, the MI inputs of the two decoders are plotted on the vertical axis and on horizontal the axis, respectively. These inputs and outputs provide us with the two curves in the two-dimensional EXIT chart exemplified at the right of Figure 1.13, which are referred to as Inner/Upper or Outer/Lower decoder curves in the literature. The extrinsic information generated during the iterative decoding process is visualized by a stair-case-shaped trajectory, which bounces between the Inner/Upper or Outer/Lower EXIT curves exemplified at the right of Figure 1.13. An open tunnel must exist between the two curves for a successful decoding, which allows the extrinsic information to reach the point (1.0, 1.0), namely the top right corner of the chart.

$$\begin{aligned}
J(\sigma_{ch}) &= H(X) - H[X|L(Y)] \\
&= 1 - \int_{-\infty}^{+\infty} \frac{e^{-(\xi - \sigma_{ch}^2/2)^2 / 2\sigma_{ch}^2}}{\sqrt{2\pi\sigma_{ch}^2}} \cdot \log_2 [1 + e^{-\xi}] d\xi \\
&\approx \begin{cases} a_{J,1}\sigma_{ch}^3 + b_{J,1}\sigma_{ch}^2 + c_{J,1}\sigma_{ch}, & 0 \leq \sigma_{ch} \leq 1.6363 \\ 1 - e^{a_{J,2}\sigma_{ch}^3 + b_{J,2}\sigma_{ch}^2 + c_{J,2}\sigma_{ch} + d_{J,2}}, & 1.6363 < \sigma_{ch} < 10 \\ 1, & 10 \leq \sigma_{ch} \end{cases}
\end{aligned} \tag{1.4}$$

The calculation of the MI between the log-likelihood ratio (LLR) and the related information bits was detailed in [133], where the LLRs must be near-Gaussian distributed signals. Specifically, let us assume that the original bits X are transmitted over the Gaussian channel, resulting in the related received signal of $Y = X + N$, where N is the Gaussian noise associated with a zero mean and a variance of σ_n^2 . Then, the LLR representation of Y may be readily denoted by $L(Y)$, which is Gaussian distributed with a mean of $\mu_{ch} = \pm 2/\sigma_n^2$ and a variance of $\sigma_{ch}^2 = 4/\sigma_n^2$. Then, the MI $J(\sigma_{ch}) = I[X, L(Y)]$ between X and $L(Y)$ may be formulated as Eq. (1.4) [134],

where we have:

$$a_{J,1} = -0.0421061, \quad b_{J,1} = 0.209252, \quad c_{J,1} = -0.00640081$$

$$a_{J,2} = 0.00181491, \quad b_{J,2} = -0.142675, \quad c_{J,2} = -0.0822054, \quad d_{J,2} = 0.0549608.$$

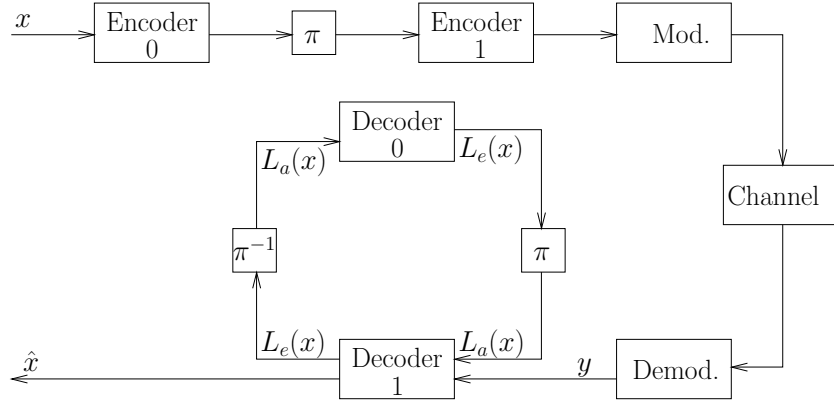


Figure 1.12: Example of an iterative decoding scheme.

System Parameters	Value	System Parameters	Value
Codec 0	SBC [2, 6]	Coding rate 0	3/4
Codec 1	RSC[1011, 1101, 1111]	Coding rate 1	1/3
Channel	Narrowband Rayleigh Fading Channel	Overall Coding Rate	1/4
		Modulation	QPSK

Table 1.5: Parameters employed in our systems, where “AA” indicates antenna array.

Given the above method of calculating the MI, the EXIT chart can be generated for arbitrary iterative decoding aided receivers. A general iterative decoding scheme is exemplified in Figure 1.12, where the original bitstream x is encoded by the serially concatenated Encoder 0 and Encoder 1. At the receiver, Decoder 0 and Decoder 1 iteratively exchange extrinsic information during the decoding process. More specifically, we may employ a short block code (SBC) as Encoder 0 and an RSC code as Encoder 1 for exemplifying the EXIT chart. In order to augment those concept more explicitly, we carried out the following investigation. The FEC encoded source bits were then mapped to a quadrature phase-shift keying (QPSK) modulator and then they were transmitted over Rayleigh fading channels. All the parameters are listed in Table 1.5. The BER curves of different iterations and the related EXIT chart are displayed in Figure 1.13. Observe from Figure 1.13 that the BER tends to become

vanishing low upon increasing the number of iterations at $E_b/N_0=2$ dB, while the related trajectory is converging to the top right corner at the (1,1) point of the EXIT chart.

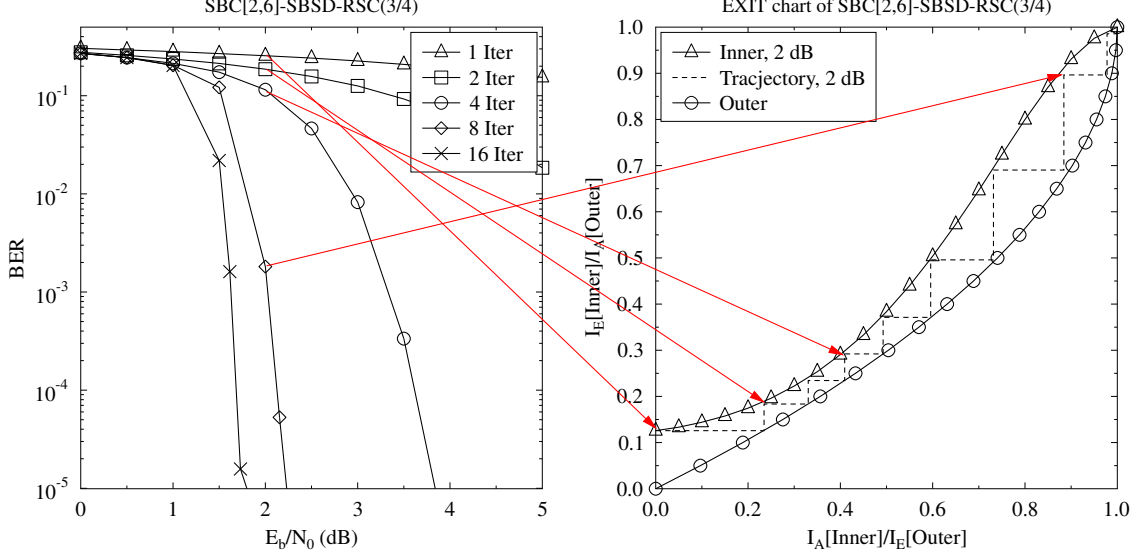


Figure 1.13: BER curves and the related EXIT chart for transmission over an uncorrelated Rayleigh channel using the parameters of Table 1.5 and the schematic of Figure 1.12.

1.4 Motivation and Methodology

A new tree-structure based MDC scheme will be proposed for the flexible generation of multiple video descriptions, which may be compressed by arbitrary video codecs. The advocated MDC technique artificially increases the redundancy within the video signals for the sake of combating the channel-induced errors.

On the other hand, realistic finite-complexity, finite-delay video encoders fail to remove all the redundancy in the source video signals. Hence it is intuitive to exploit the residual redundancy for improving the attainable system performance. For example, there is residual redundancy in the layered video coded bitstreams, which may be manifested by the fact that the ELs rely on the BL for decoding. In the traditional UEP schemes conceived for layered video communication, variable-rate FEC was invoked for the different layers. When the BL is corrupted or lost, the ELs also have to be dropped, regardless whether they are perfectly received or not, which implies that the transmission power assigned to the ELs was wasted. Motivated these background, we developed an inter-layer operation aided FEC (IL-FEC) scheme relying on a systematic FEC code in [75], where the systematic bits of the BL were implanted into the ELs. At the receiver, the above-mentioned implanted bit of the ELs may be utilized for assisting in decoding the BL. The IL-FEC technique

of was also combined with the UEP philosophy for the sake of further improving the attainable system performance.

However, the above-mentioned IL techniques cannot guarantee the best system performance for diverse video sequences. Hence we propose a technique for determining the coding rates “on-the-fly” for optimizing the IL-FEC coded system’s performance. Specifically, the optimized coding rates are capable of minimizing the video quality distortion.

Furthermore, a high amount of redundancy is present in the transmitted uncompressed video, which manifests itself in terms of a high adjacent-pixel correlation as well as inter-frame correlation, which may be exploited at the receiver for concealing the pixel errors in the reconstructed video. Motivated by the congenial principle of ISCD, in Chapter 4 we will design an ISCD technique for video communication, where the video-domain redundancy is exploited for iteratively correcting the channel-induced video impairments. Specifically, in Chapter 4 we will propose a sophisticated iterative horizontal-vertical scanline model (IHVSM) aided iterative source decoding model, which iteratively exchanges extrinsic information with the FEC codec for the sake of exploiting the intra-frame pixel-domain redundancy.

Similar to the above mentioned IHVSM, we also develop the three-dimensional ISCD concept relying on an RSC codec for uncompressed video transmissions, where both the intra-frame and the inter-frame correlations are exploited relying on first-order Markov processes. A single RSC codec is combined with three source decoders for forming three decoder-pairs, for three-stage decoding.

Furthermore, in Chapter 5 we will proposed an ISCD aided WZ video coding system, which employs the above-mentioned iterative horizontal-vertical source modelling (IHVSM) aided decoder of Chapter 5. Similarly the inter-view correlation is not removed by the WZ multi-view video encoder either. Hence it is beneficial to exploit the residual correlation at the receiver for the sake of reducing the required bitrate. Therefore, in Chapter 5 we develop a novel mesh-structured source model (MSSM) based decoder for exploiting the inter-view correlation among the pixels, which will be combined with a turbo codec for performing iterative source-channel decoding in the context of distributed multi-view video coding (DMVC) for the sake of achieving a reduced bitrate.

1.5 Outline of the Thesis

Having briefly reviewed the literature of video coding, as well as the unequal error protection [135] and iterative source-channel decoding philosophies, let us now outline

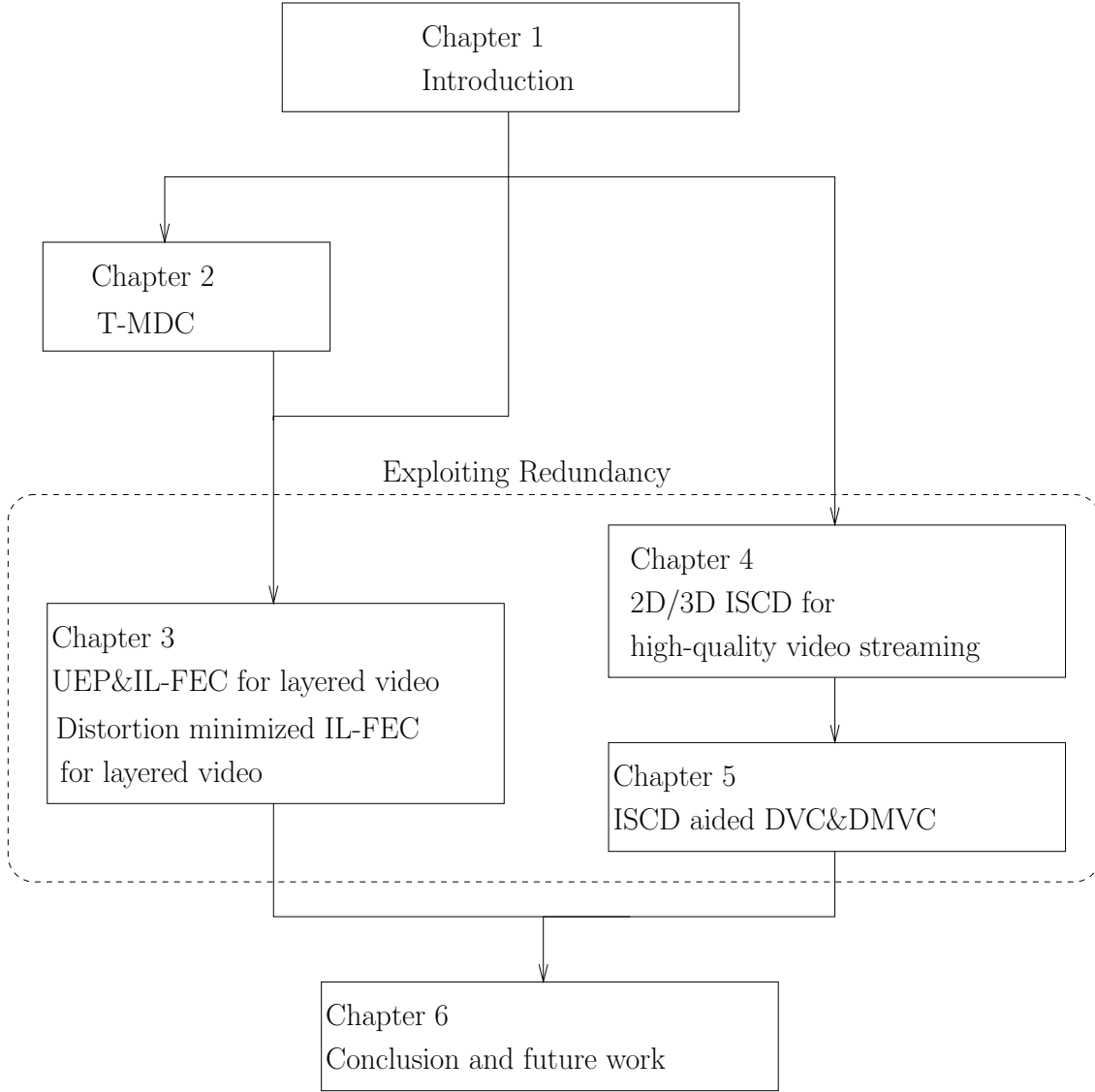


Figure 1.14: Outline of the thesis

the organization of the thesis, which is displayed in Figure 1.14. The rest chapters are organized as follows:

- **Chapter 2: Tree-Structured Multiple Description Coding**

As outlined in Figure 1.14, in this chapter a novel tree-structured multiple description codec (T-MDC) [136] is conceived, which may be beneficially combined with arbitrary video codecs. The technique advocated splits the original video signal into a number of correlated descriptions in the time-domain, while retaining the correlation among the consecutive video frames within each description. Furthermore, we will demonstrate that our proposed scheme is also capable of splitting the video stream into multiple descriptions of unequal importance.

- **Chapter 3: Inter-Layer Coded Unequal Error Protection**

As outlined in Figure 1.14, in this chapter we propose inter-layer FEC (IL-FEC) techniques for partitioned H.264 video coding, where the systematic information of the BL will be implanted into the ELs with the aid of an XOR operation for the sake of achieving an improved reconstructed video quality. Additionally, in order to meet the challenging performance requirements of bandwidth-constrained environments, a multiple input multiple output (MIMO) system, namely the sophisticated Layered Steered Space-Time Codes (LSSTC) [137, 138], will be employed. The LSSTCs are capable of providing both a diversity gain for the sake of achieving a high BER performance in mobile environments as well as attaining a multiplexing gain in order to maintain a high data rate. Then the system is analyzed using EXIT charts in Section 3.3. The performance of the IL-FEC techniques will be benchmarked using diverse FECs in various diverse layered video coding scenarios in Sections 3.5 and 3.6, where the results recorded for multiple video sequences will be presented. Moreover, this chapter will also optimize the IL-FEC system, as outlined in Figure 1.14. In Section 3.4, we propose a technique conceiving for finding the optimized coding rates of the coded bitstreams operating “on-the-fly” at the transmitter, which optimizes the IL-FEC coded system’s performance. Specifically, we find the coding rates achieving the minimum video quality distortion by maximizing the mutual information (MI) between the LLRs and the corresponding video bits, where the soft-decoding metric of the FEC codec and of the demodulator are characterized by lookup tables (LUTs), since these cannot be readily characterized theoretically. The widely employed SVC-H.264 codec will be utilized for the sake of benchmarking our proposed system. Finally, Section 3.7 characterizes the attainable system performance in the scenario of SVC using RSC codes.

- **Chapter 4: Iterative Source-Channel Decoding Aided Wireless Video**

As illustrated in Figure 1.14, a novel two-dimensional ISCD technique was proposed for reducing both the bitrate and power consumption in the context of high-quality uncompressed video transmission in Section 4.2. Section 4.3 details the first-order Markov process aided three-dimensional iterative source-channel decoding concept relying on a recursive systematic convolutional codec for uncompressed video transmissions, where both the horizontal and vertical intra-frame correlations as well as the inter-frame correlations were exploited by relying on first-order Markov processes.

- **Chapter 5: Iterative Source-Channel Decoding Aided Distributed Video Coding**

As shown in Figure 1.14, in Chapter 5, the two-dimensional ISCD technique of

Chapter 4 was firstly applied for reducing both the bitrate and power consumption of distributed video coding. Then, we conceive a novel mesh-structured pixel correlation model from the interview-motion vectors and derive its decoding rules, which may be utilized in distributed multiview video coding. Specifically, Section 5.4.1 will illustrate our system model. A novel trellis is designed for exploiting the inter-view correlation of the multi-view video in Section 5.4.2.1, followed by the corresponding trellis decoding algorithm in Section 5.4.2.2. The system's performance will be studied in Section 5.4.3.

- **Chapter 6: Conclusions and Future Work**

As outlined in Figure 1.14, in this concluding chapter, the major findings of our work are summarized along with our suggestions for future research.

1.6 Novel Contributions

The thesis is based on the publications and manuscript submissions of [136, 139, 140, 141, 75]. The novel contribution of this thesis includes the following:

- A novel tree-structured multiple description codec (T-MDC) [136] was proposed, which is flexible and may be combined with arbitrary video codecs. The technique advocated splits the original video signal into a pre-set number of correlated descriptions in the time-domain, while retaining the correlation among the video frames within each description. Furthermore, our proposed scheme is also capable of splitting the video stream into multiple descriptions of unequal importance.
- A novel inter-layer FEC (IL-FEC) coded video scheme [139, 141, 75] was proposed, where the information of the base layer was also incorporated into the systematic bits of the enhancement layers with the aid of an XOR operation. When the base layer can be successfully decoded in its own right, the systematic bits of the enhancement layers can be extracted by flipping the sign of the appropriately generated check information received. More explicitly, the check information is generated by performing inter-layer *XOR* operation on the base layer and the enhancement layers, which effectively embeds the related parity information into the EL without increasing its bitrate. However, when the base layer cannot be correctly decoded without the assistance of the enhancement layers, the inter-layer FEC decoding philosophy exchanging information between the base layer and the enhancement layers will be activated to assist in decoding the base layer. Furthermore, we propose a technique for finding the optimized coding rates of the individual coded bitstreams “on-the-fly” at the

transmitter, which optimizes the IL-FEC coded system's performance. Specifically, we find the coding rates achieving the minimum video quality distortion with the aid of the MI between the LLRs and the corresponding video bits. The soft-decoding metric of both the FEC codec and of the demodulator are characterized by appropriate lookup tables (LUTs), since these cannot be readily characterized theoretically. Finally, the widely employed SVC-H.264 codec were utilized for the sake of benchmarking our proposed system.

- We proposed a two-dimensional iterative source-channel decoding algorithm in [140], which may be combined with channel decoding in the scenarios of uncompressed video transmission and distributed video coding. Specifically, horizontal and vertical first-order Markov processes were designed for the sake of exploiting the intra-frame video correlation. Furthermore, a reduced-complexity first-order Markov model based source decoder was derived. Iterative decoding was performed by exchanging extrinsic information between two source decoders. Then we proposed the first-order Markov process aided three-dimensional iterative source-channel decoding concept of [140] relying on an RSC codec for uncompressed video transmissions. Both the horizontal and vertical intra-frame correlations as well as the inter-frame correlations were exploited by relying on first-order Markov processes. Furthermore, a single RSC codec was combined with three independent source decoders for the sake of forming three decoder-pairs for three-stage decoding, where the RSC was utilized for improving the source decoder's convergence behavior. Finally, we proposed a system architecture for the Wyner-Ziv coding of multiview video compression. We conceived a novel mesh-structured pixel correlation model from the interview motion vectors and derived its decoding rules. This technique was successfully employed in a distributed multiview video coding scenario for the sake of reducing the required bitrate.

Tree-Structured Multiple Description Coding

2.1 Introduction

Layered video coding encodes a video sequence into a BL and multiple ELs [75]. The BL is necessary for decoding the video, while the ELs may be utilized for further improving the reconstructed video quality. However, the ELs are useless, when the BL is corrupted or lost during video transmission. In general, this implies that the quality of a reconstructed video stream is not directly proportional to the amount of correctly received data, despite our expectations.

Multiple description coding (MDC) [30, 31] represents a video sequence in terms of a number of substreams referred to as descriptions. The different descriptions may be transmitted over different TCP-IP routes. Any subset of these description may be employed for the sake of recovering the original video sequence stream, and naturally the quality improves upon increasing the number of descriptions received. MDC is capable of providing an enhanced error resilience for video transmissions. Since a description corruption event imposed by network congestion or packet loss events will not corrupt the entire stream, it only imposes a video quality degradation. Hence, the quality of a reconstructed video may be expected to be proportional to the amount of data received in MDC.

Similar to conventional single-view video, multiview video also suffers from packet loss events imposed by network congestion. Hence it is necessary to design techniques for multiview video communications in order to combat the packet loss events. Again, MDC [30, 31] has been introduced to overcome the deleterious effects of channel errors with the aid of source encoding diversity. More explicitly, the source may be encoded

into several correlated representations/descriptions. Any subset of these descriptions may be independently decoded at the receiver. MDC may be deemed to be an attractive solution for multiview video streaming over unreliable networks [142], such as the Internet.

In this chapter, we detail a novel tree-structured multiple description codec (T-MDC) [136], which is flexible and may be combined with arbitrary video codecs. The technique advocated splits the original video signal into a pre-set number of correlated descriptions in the time-domain, while retaining the correlation among the video frames within each description. Our proposed scheme is also capable of splitting the video stream into multiple descriptions of unequal importance. Specifically, Section 2.2 will present the tree-structured creation of multiple video descriptions, followed by detailing the corresponding reconstruction algorithm designed for both lossless and lossy video compression in Section 2.3. The framework of applying the proposed T-MDC to MVC is described in more detail in Section 2.4. The performance of two T-MDC structures is characterized in Section 2.5 in the context of both conventional monoscopic and multiview video sequences. Finally, we conclude this chapter in Section 2.6.

2.2 Tree-Structured Multiple Description Creation

It is widely recognized that the addition of a constant luminance level to a video signal does not affect the correlation among video frames. In this study, we consider gray-scale video sequences, but the proposed technique may be readily extended to the color YUV or RGB video formats. Let us commence our discourse by illustrating a simple method of expressing a single pixel value (positive integer) with the aid of multiple integers using an example. Let us assume that the pixel value equals x , which may be expressed as

$$\{x|L, (x+1)|L, (x+2)|L, \dots, (x+L-1)|L\}, \quad (2.1)$$

where $|$ represents the aliquot part upon division by L , which may also be viewed as the action of quantization. For example, if we have $x = 50, L = 4$, x may be expressed as $\{12, 12, 13, 13\}$. Provided that all the integers $\{12, 12, 13, 13\}$ are known, the original value can be readily recovered.

Based on Eq. (2.1), let us now discuss the method of creating multiple video descriptions which may be of practical importance in numerous applications, including

both equal and unequal importance descriptions. For example, equal-importance descriptions may display different camera-angles in interactive TV [143, 144] or the left-eye/right-eye views of stereoscopic video [145]. By contrast, less important descriptions-containing for example a low-resolution version of a scene may be dropped during instances of network-congestion [30]. For a $(c \times r)$ -pixel luminance video frame, the pixel value matrix may be expressed as

$$\begin{bmatrix} x_{0,0} & \cdots & x_{0,c-1} \\ \vdots & \vdots & \vdots \\ x_{r-1,0} & \cdots & x_{r-1,c-1} \end{bmatrix}. \quad (2.2)$$

For example, a (4×4) -pixel block of a video frame may be exemplified as

$$\begin{bmatrix} x_{0,0} = 50 & x_{0,1} = 49 & x_{0,2} = 48 & x_{0,3} = 48 \\ x_{1,0} = 45 & x_{1,1} = 44 & x_{1,2} = 45 & x_{1,3} = 45 \\ x_{2,0} = 39 & x_{2,1} = 38 & x_{2,2} = 40 & x_{2,3} = 38 \\ x_{3,0} = 64 & x_{3,1} = 64 & x_{3,2} = 62 & x_{3,3} = 62 \end{bmatrix}. \quad (2.3)$$

2.2.1 Descriptions of Equal Importance

Let us assume that we want to describe the video clip using L descriptions. By applying Eq. (2.1) to each pixel of a frame, we can generate a specific description of the frame for each offset l ($0 \leq l < L$). The l^{th} description corresponding to L in the temporal domain is formulated as:

$$\begin{bmatrix} (x_{0,0} + l)|L & \cdots & (x_{0,c-1} + l)|L \\ \vdots & \vdots & \vdots \\ (x_{r-1,0} + l)|L & \cdots & (x_{r-1,c-1} + l)|L \end{bmatrix}. \quad (2.4)$$

Now we have L correlated descriptions of the video frame, and each of which carries a quantized but equal-importance version of the original video, while also containing independent high frequency information. We may now treat the original video signal as the root of the corresponding coding tree, where the different equal-importance descriptions are the leaves of the coding-tree. We may refer to the coding-tree as the height-one complete tree (HOCT) which is presented in Figure 2.1. The HOCT-MDC coding tree for $x = 50$ using $Q = 4$ is detailed in Figure 2.2, resulting in the descriptions of $\{12, 12, 13, 13\}$.

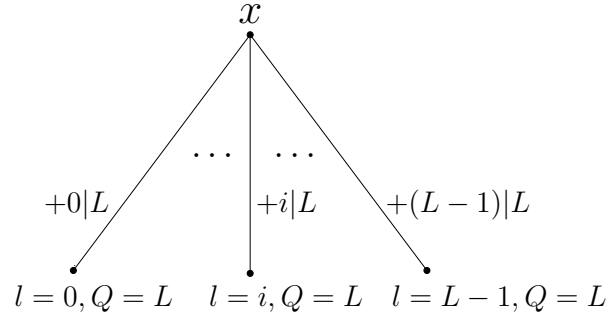


Figure 2.1: Proposed HOCT-MDC coding-tree, where $+l|Q$ means the addition of l to each video pixel and then its quantization with Q .

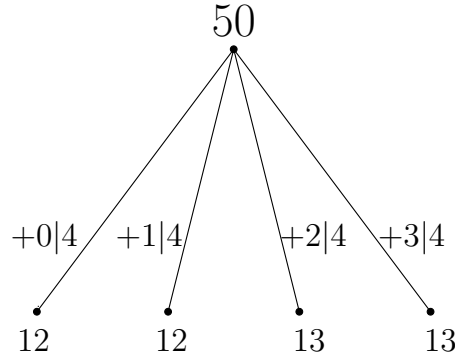


Figure 2.2: The pixel of $x = 50$ encoded by the HOCT-MDC coding tree using $Q = 4$.

$$\begin{aligned}
 & \begin{bmatrix} x_{0,0} = 50 & x_{0,1} = 49 & x_{0,2} = 48 & x_{0,3} = 48 \\ x_{1,0} = 45 & x_{1,1} = 44 & x_{1,2} = 45 & x_{1,3} = 45 \\ x_{2,0} = 39 & x_{2,1} = 38 & x_{2,2} = 40 & x_{2,3} = 38 \\ x_{3,0} = 64 & x_{3,1} = 64 & x_{3,2} = 62 & x_{3,3} = 62 \end{bmatrix} \\
 & \quad \downarrow Q = 4 \\
 & \overbrace{\begin{bmatrix} 12 & 12 & 12 & 12 \\ 11 & 11 & 11 & 11 \\ 9 & 9 & 10 & 9 \\ 16 & 16 & 15 & 15 \end{bmatrix} \quad \begin{bmatrix} 12 & 12 & 12 & 12 \\ 11 & 11 & 11 & 11 \\ 10 & 9 & 10 & 9 \\ 16 & 16 & 16 & 15 \end{bmatrix} \quad \begin{bmatrix} 13 & 12 & 12 & 12 \\ 11 & 11 & 11 & 11 \\ 10 & 10 & 10 & 10 \\ 16 & 16 & 15 & 16 \end{bmatrix} \quad \begin{bmatrix} 13 & 13 & 12 & 12 \\ 12 & 11 & 12 & 12 \\ 10 & 10 & 10 & 10 \\ 16 & 16 & 16 & 16 \end{bmatrix}} \\
 & \quad \begin{matrix} +0|4 & +1|4 & +2|4 & +3|4 \end{matrix}
 \end{aligned}$$

Figure 2.3: The 4×4 -pixel block of Eq. (2.3) encoded by the HOCT-MDC coding tree of Figure 2.1 using $Q = 4$.

In the coding-tree generation procedure, only two operations, namely the addition of a direct current (DC) component and the quantization are required. The higher the value of L , the more of the originally different pixels become identical after retaining the aliquot part, which results in a more correlated sequence than the original one, because the smaller pixel-differences representing the high-frequency components disappear after retaining the aliquot part. Hence the resultant aliquot part becomes more amenable to compression. The 4×4 -pixel block of Eq. (2.3) is encoded by the HOCT-MDC as exemplified in Figure 2.3 for the sake of providing further insights on how the high-frequency correlations are changed. More explicitly, this property facilitates the encoding of these correlated aliquot-based descriptions into bitstreams using any existing lossy or lossless compression schemes, including standardized video codecs. More specifically, L different video descriptions resulting in L compressed bit streams may be generated independently using for example lossy video compression schemes, such as the MPEG-2 and H.264 codecs [1]. The proposed codec architecture is displayed in Figure 2.4, where the offsets $l = 0, \dots, L - 1$ and quantizers resolutions $Q = L, \dots, Q = L$ may be associated with any specific coding-tree structure. After the aliquot encoding stage of Figure 2.4, the L bitstreams having indices of $l = 0, \dots, L - 1$ may be transmitted to the receiver via multiple wireless channels or TCP-IP routes. Provided that no packets are lost during transmission, L descriptions will be received at the receiver. By contrast, less than L descriptions may be received in the presence of packet erasures imposed by route-congestion or channel-errors. To simplify our discourse, let us assume that all the L bitstreams are received at the receiver. Then the receiver has to recover the original video, namely x in Figure 2.4, from the L bitstreams. Two reconstruction stages are involved in the recovery procedure of Figure 2.4, namely the reconstruction of the L aliquot descriptions from the L received bitstreams, followed by the reconstruction of the original video x from the reconstructed L aliquot descriptions. The aliquot reconstruction may be accomplished by the decoder pair of the aliquot encoder. The second reconstruction stage, namely the reconstruction of the original video clip will be illustrated in great detail in Section 2.3.

2.2.2 Descriptions of Unequal Importance

Again, the above method generates multiple descriptions of equal importance. However, in many practical multi-rate transceivers [146] or multi-route Internet based applications, it may be beneficial to transmit different-importance descriptions, so that the least important ones may be dropped in case of low instantaneous channel qualities [1] or network-congestion. Our scheme is also capable of generating descriptions of different importance by repeatedly and hierarchically splitting any specific

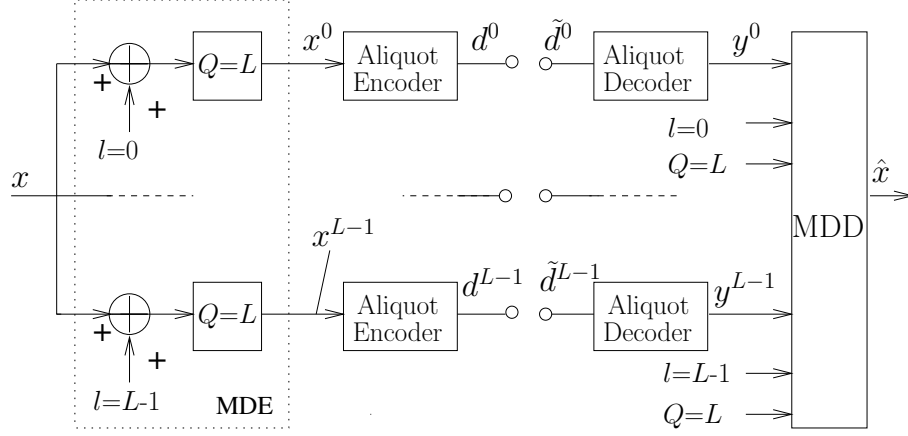


Figure 2.4: The proposed HOCT-MDC architecture. The *Aliquot Encoder* and *Aliquot Decoder* represent the codecs employed to encode and reconstruct the aliquot part descriptions, $d^0 \dots d^{L-1}$ represent the generated bitstreams, while *MDD* represents the multiple descriptions decoder for video reconstruction.

subset of the L descriptions. Specifically, if we only employ the quantizer value of $L = 2$, we generate a binary-tree based MDC (BT-MDC), which is exemplified in Figure 2.5. Theoretically any specific tree structure of descriptions may be generated. Using the proposed method, we can then remember the description offsets l and the quantizer resolution Q along the path from the root to any specific leaf of the coding tree. Finally, we can combine all $[\text{offset}, \text{quantizer}] = [l, Q]$ pairs into a single parameter, as exemplified in Figure 2.5. This implies that generating a more complex coding-tree structure only modestly increases the encoding complexity. Furthermore, by simply assigning $[l, Q]$ pairs of Figure 2.4 to the $[\text{offset}, \text{quantizer}]$ parameter pair of the leaves seen in Figure 2.5, we can readily generate the corresponding architecture for the coding-tree of Figure 2.5. The pixel value of $x = 50$ may be encoded by the BT-MDC coding tree of Figure 2.5 as detailed in Figure 2.6, resulting in the descriptions of $\{25, 12, 6, 3, 4\}$. The 4×4 -pixel block of Eq. (2.3) is encoded by the BT-MDC of Figure 2.5, as seen in Figure 2.7.

2.3 Video Reconstruction

This section outlines the reconstruction of the original video clip. For simplicity, we consider the reconstruction algorithm of the Multiple Description Encoder (MDE) of Figure 2.4. Other tree-structured MDC schemes may be readily reconstructed by simply modifying the reconstruction parameters of the MDE considered. Here we will commence by illustrating a simple reconstruction technique conceived for lossless coding. However, in practical video systems typically "lossy" compression is used. Hence Section 2.3.2 illustrates further "lossy" reconstruction techniques required for

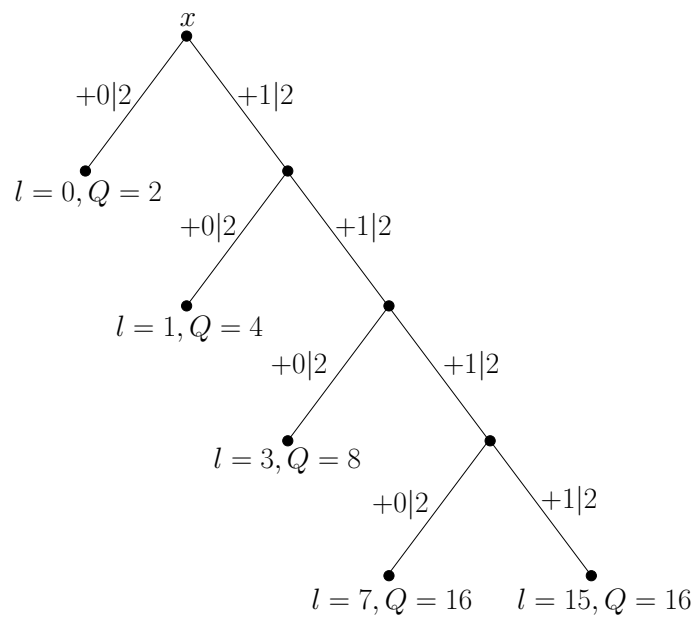


Figure 2.5: An example of BT-MDC coding-tree, where $+i|Q$ means the addition of i to each video pixel and then its quantization with Q .

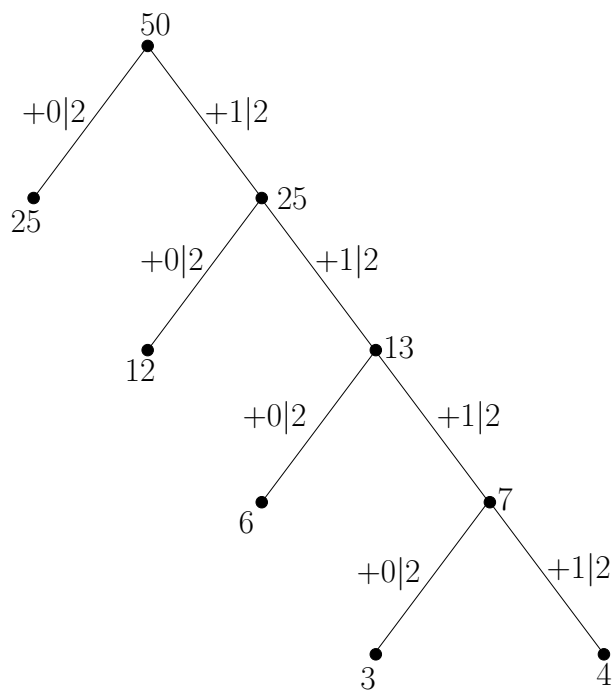


Figure 2.6: $x = 50$ encoded by BT-MDC coding tree of Figure 2.5.

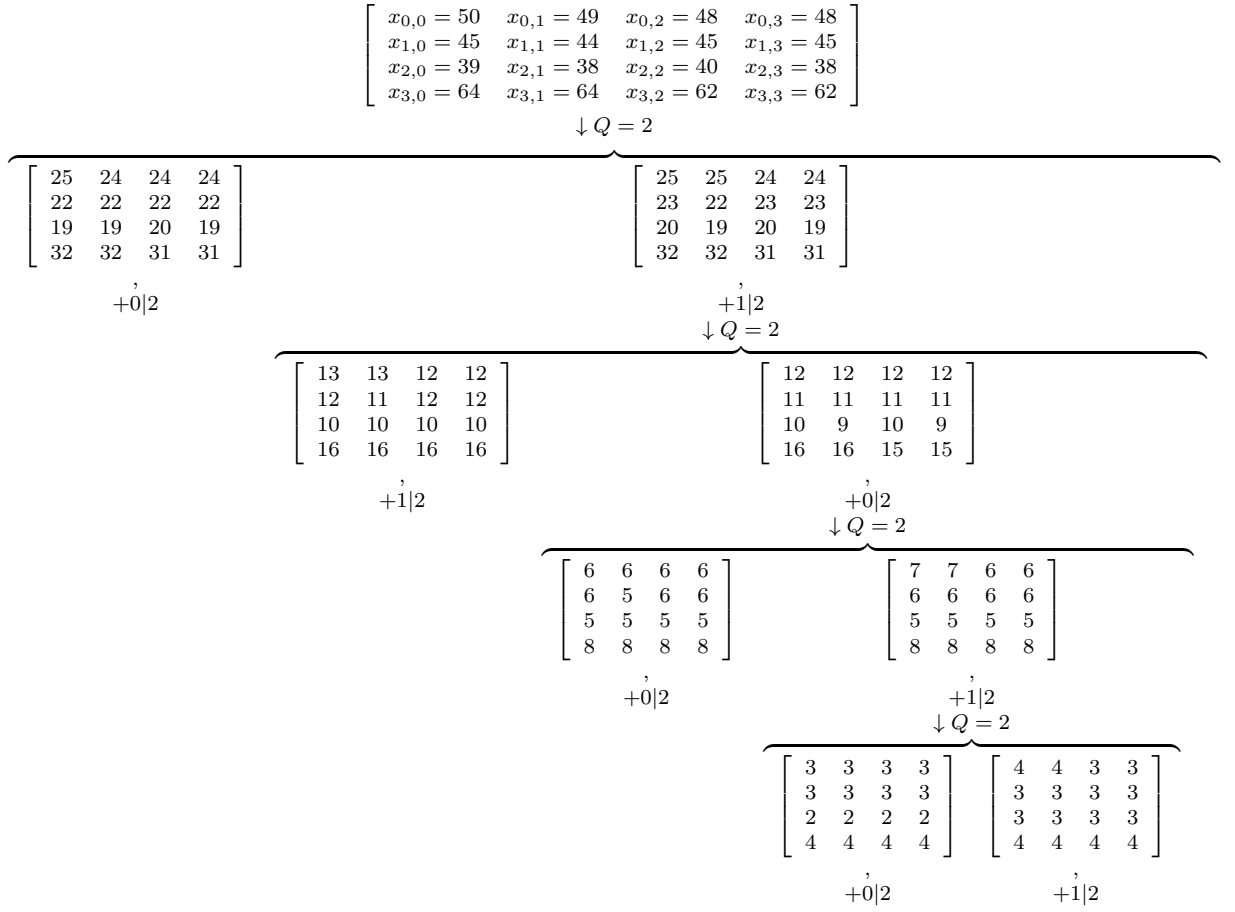


Figure 2.7: The 4×4 -pixel block of Eq. (2.3) encoded by BT-MDC coding tree of Figure 2.5.

practical applications. Before detailing the reconstruction methods, we stipulate the following assumptions:

- $x_{i,j}$: the pixel value at position (i, j) in a specific video frame;
- L : the quantizer resolution;
- $S_L = \{0, \dots, L - 1\}$: the set of descriptions received at the receiver, where each element corresponds to an offset invoked for creating a specific description;
- $X_m = \{0, \dots, 2^m - 1\}$: the set of m -bit luminance pixel values;
- $x_{i,j}^l$: the aliquot part of the pixel $x_{i,j}$ upon division by L in the l^{th} description, $l \in S_L$;
- d^l, \tilde{d}^l : the l^{th} bitstream at the transmitter and receiver respectively, $l \in S_L$;
- $y_{i,j}^l$: the l^{th} aliquot part of the pixel $x_{i,j}$ reconstructed by the aliquot decoder at the receiver, $l \in S_L$;
- $\hat{x}_{i,j}$: the estimated pixel value at position (i, j) ;

Based on the above notations, an entire $c \times r$ -pixel video frame of the l^{th} ($l \in S_L$) descriptions may be expressed in aliquot part form as follows:

$$\begin{bmatrix} x_{0,0}^l & \cdots & x_{0,c-1}^l \\ \vdots & \vdots & \vdots \\ x_{r-1,0}^l & \cdots & x_{r-1,c-1}^l \end{bmatrix}. \quad (2.5)$$

Recall from Figure 2.4, that the l^{th} description consisting of a sequence of video frames as described in Eq. (2.5) will be encoded using either lossy or lossless aliquot encoders and the corresponding bitstream $d^l, l = 0, \dots, L-1$ will be generated. Then, the L bitstreams d^0, \dots, d^{L-1} may be transmitted to the receiver through a number of wireless channels or TCP-IP routes. At the receiver, the L received bitstreams $\tilde{d}^0, \dots, \tilde{d}^{L-1}$ must be decoded using the aliquot decoder in order to reconstruct the L aliquot descriptions. An entire frame of the l^{th} ($l \in S_L$) aliquot description reconstructed by the aliquot decoder may be expressed as follows:

$$\begin{bmatrix} y_{0,0}^l & \cdots & y_{0,c-1}^l \\ \vdots & \vdots & \vdots \\ y_{r-1,0}^l & \cdots & y_{r-1,c-1}^l \end{bmatrix}. \quad (2.6)$$

Here, for each original pixel, we obtain L aliquot parts $y_{i,j}^l, l \in S_L$ at the receiver, which we will decode for recovering the original pixel $x_{i,j}$ in the following two sections.

Note that here we continue our discourse based on the assumption that all descriptions $S_L = \{0, \dots, L-1\}$ are received at the receiver, albeit the reconstruction procedure may rely on any subset of the full set S_L .

2.3.1 Reconstruction for Lossless Aliquot Encoding

Naturally, the original video sequence may be recovered from any of its L descriptions, but an improved video quality may be expected upon beneficially combining several descriptions. This will be further detailed below. Upon Eq. (2.1), for each received description $l \in S_L$, we arrive at:

$$y_{i,j}^l \cdot L - l \leq \hat{x}_{i,j} < (y_{i,j}^l + 1) \cdot L - l. \quad (2.7)$$

Provided that the transmitted aliquot part is perfectly received, namely $y_{i,j}^l = x_{i,j}^l$, then we have

$$\hat{x}_{i,j} \begin{cases} < \min \{ (y_{i,j}^l + 1)L - l \mid l \in S_L \} \\ \geq \max \{ y_{i,j}^l L - l \mid l \in S_L \}. \end{cases} \quad (2.8)$$

When some of the descriptions become unavailable owing to channel-induced packet loss events, we may simply choose the average of the available pixel values for the estimated value of $\hat{x}_{i,j}$. Considering the HOCT-MDC example of Figure 2.2, \hat{x} may be readily recovered as $12 + 12 + 13 + 13 = 50$. Similarly, given the descriptions $\{25, 12, 6, 3, 4\}$ of Figure 2.6, \hat{x} may be readily recovered as $25 + 12 + 6 + 3 + 4 = 50$.

2.3.2 Reconstruction for Lossy Aliquot Encoding

In case of lossy compression and error-infested channel decoded scenarios, the aliquot part obtained at the receiver after channel decoding and aliquot reconstruction may be expressed as $y_{i,j}^l = x_{i,j}^l + \delta$, where the reconstruction error δ is introduced by the aliquot decoder and channel decoder. Here we ignore the effects of transmission errors for simplicity. Then δ is solely the lossy aliquot decoder's reconstruction error. We now have to recover the original video pixel $x_{i,j}$ based on the received aliquot parts $y_{i,j}^0, \dots, y_{i,j}^{L-1}$. Below, we will now introduce the direct mathematical formulation of reconstructing the original pixel $x_{i,j}$, which depends on the probability of $x \in X_m$ conditioned on the aliquot parts $y_{i,j}^0, \dots, y_{i,j}^{L-1}$ reconstructed by the aliquot decoder of Figure 2.4. However, the direct mathematical rule requires a huge amount of side-information of the original video and the reconstruction error distribution of the aliquot codec. Hence we will then simplify the video clip reconstruction based on a few reasonable assumptions, which will assist us in reducing the amount of side-information required. Since we will focus our attention on a single pixel here, we simplify our notations by treating a pixel value without its position index, i.e. we use x^l instead of $x_{i,j}^l$. Furthermore, the notation y_0^{L-1} represents $y_{i,j}^0, \dots, y_{i,j}^{L-1}$.

Let us now assume that we have received all the aliquot values y_0^{L-1} from L different reconstructed aliquot descriptions at a given pixel position, although as noted above, even a single description is sufficient for adequately reconstructing the original video sequence. Naturally, having multiple descriptions is expected to improve the reconstructed video quality. Based on these received aliquot values, the original pixel value x can be recovered using for example either

- the MMSE estimation rule of

$$\hat{x} = \sum_{x \in X_m} x \cdot p(x|y_0^{L-1}), \quad (2.9)$$

- or the MAP estimation rule of

$$\hat{x} = \arg \max_{\forall x \in X_m} p(x|y_0^{L-1}). \quad (2.10)$$

Based on Bayes' theorem and on the chain rule of probability, the *a-posteriori* probability of occurrence $p(x|y_0^{L-1})$ in Eq. (2.9) and Eq. (2.10) may be formulated as follows

$$p(x|y_0^{L-1}) = \frac{p(y_0^{L-1}|x) \cdot p(x)}{\sum_{u \in X_m} p(y_0^{L-1}|u) \cdot p(u)}. \quad (2.11)$$

Furthermore, let us define the aliquot reconstruction error $\delta^l = y^l - x^l$ of the l^{th} description, which is solely introduced by the aliquot codec, since the channel effects are ignored. Then the PDF $p(y_0^{L-1}|x)$ in Eq. (2.11) may be formulated as follows

$$\begin{aligned} p(y_0^{L-1}|x) &= p(y^0, \dots, y^{L-1}|x) \\ &= p(\delta^0, \dots, \delta^{L-1}|x) = p(\delta_0^{L-1}|x). \end{aligned} \quad (2.12)$$

Let us now discuss the calculation of the joint probability $p(\delta_0^{L-1}|x)$ in Eq. (2.12). Let us consider a simple video codec comprised of a *quantizer* and a *bit mapper* as the aliquot codec. Then the aliquot reconstruction error $\delta^l, l \in S_L$ arises solely due to the *quantizer*. The aliquot reconstruction errors $\delta^0, \dots, \delta^{L-1}$ are independent of each other, when the original pixel x is given. When a more complex aliquot codec is employed, such as H.264, the aliquot reconstruction errors δ_0^{L-1} introduced by the aliquot codec may be deemed to be independent of each other. Hence their joint probability is given by the product of the individual probabilities, usually by

$$p(\delta_0^{L-1}|x) = p(\delta^{L-1}|\delta_0^{L-2}, x) \cdots p(\delta^0|x) = \prod_{l \in S_L} p(\delta^l|x). \quad (2.13)$$

Upon combining Eq. (2.11) , 2.12 and (2.13), the *a-posteriori* probability of pixel x conditioned on all the L reconstructed aliquot parts y_0^{L-1} may be expressed as

$$p(x|y_0^{L-1}) = \frac{\prod_{l \in S_L} p(\delta^l = y^l - x^l|x) \cdot p(x)}{\sum_{u \in X_m} p(u) \cdot \prod_{l \in S_L} p(\delta^l = u^l - x^l|u)}, \quad (2.14)$$

where $p(\delta^l|x)$ is the distribution of the reconstructed aliquot part error conditioned upon the pixel value x , while $p(x)$ is the distribution of the original pixels. Eq. (2.14) can then be used for video reconstruction from the L descriptions, provided that the two PDFs $p(\delta^l|x)$ and $p(x)$ are known at the receiver. In practice, these PDFs have to be evaluated for a representative video training sequence and stored at the receiver.

However, it is infeasible to pre-evaluate these two video feature-dependent distributions for every specific video. Hence the two video feature distributions need to be statistically evaluated at the transmitter and signaled to the receiver periodically in the video reconstruction procedure, which constitute the side-information required. Since the distribution $p(x)$ is solely about the distribution of the original pixels, hence it is identical in the permutation of the L descriptions in Eq. (2.14). However the component $p(\delta^l|x)$, which is a distribution conditioned upon x , is different upon different description $l, l \in S_L$. Hence item $p(\delta^l|x)$ constitute the main signaling of the side-information. The dynamic range of δ^l depends on the aliquot encoder and on the value of L . Moreover, the evaluated side-information may change upon different frames. However, the side-information may be signaled with a period of a number of frames with modest degradation of the reconstructed video quality. Apparently, the more side-information signaled the higher video reconstruction quality could be.

Since the video-feature dependent side-information is meaningless to users, below we will provide a further approximation of Eq. (2.14), which reduces the required side-information. According to probability theory, we have

$$p(\delta^l) = \sum_{x \in X_m} [p(\delta^l|x) \cdot p(x)]. \quad (2.15)$$

Since y^l solely relies on the l^{th} aliquot part, according to the reconstruction error definition $\delta^l = y^l - x^l$, we have $p(\delta^l|x) = p(\delta^l|x^l)$. Then the right side of Eq. (2.15) may be expressed as

$$\sum_{x \in X_m} [p(\delta^l|x) \cdot p(x)] = \sum_{x \in X_m} [p(\delta^l|x^l) \cdot p(x)]. \quad (2.16)$$

For two arbitrary pixels $x, z \in X_m$, we make the following assumptions

$$\begin{aligned} p(\delta^l|x^l) &= p(\delta^l|z^l) \\ p(x) &= p(z), \end{aligned} \quad (2.17)$$

where the first line implies that any aliquot part in the l^{th} description has the same probability of occurrence for the same reconstruction error, which corresponds to a uniformly distributed reconstruction error. Similarly, the second line implies that the pixel value of the video source obey a uniform distribution, which is a simplifying assumption. Based on these assumptions, the right side of Eq. (2.16)

$\sum_{x \in X_m} [p(\delta^l|x^l) \cdot p(x)]$ may be approximated as

$$\sum_{x \in X_m} [p(\delta^l|x^l) \cdot p(x)] = p(\delta^l|x^l). \quad (2.18)$$

With the aid of Eq. (2.15) to (2.18), the conditional probability of the reconstruction error $p(\delta^l|x^l)$ equals $p(\delta^l)$, then Eq. (2.14) can be further simplified as follows:

$$p(x|y_0^{L-1}) = \frac{\prod_{l \in S_L} p(\delta^l = y^l - x^l)}{\sum_{u \in X_m} \left[\prod_{l \in S_L} p(\delta^l = y^l - u^l) \right]}, \quad (2.19)$$

where the solely side-information, namely the reconstruction error $p(\delta^l)$ ($\delta^l \in [-R_l, R_l]$) represents a one-dimensional distribution and need less side-information signaling and may impose a reduced recovery complexity of the video than Eq. (2.14). Moreover, in practical applications, some mathematical tools such as Laplacian approximation may be employed to further reduce the side-information by approximating the distribution. Finally, by combining Eq. (2.9) or Eq. (2.10) with Eq. (2.19), \hat{x} may be estimated.

2.4 Multiple Description Coding of Multiview Video

In this section, we employ the proposed HOCT-MDC architecture of Figure 2.4 for multiview video transmission. As shown in Figure 2.8, K different camera-views are input to the system and L description streams are created. In order to encode each of the K video input streams considered using the MDE of Figure 2.4, K HOCT-MDC encoders are employed. Each of the K input views generates L aliquot descriptions, as seen in Figure 2.8. After the HOCT-MDC encoder stage, a total of $(K \times L)$ aliquot

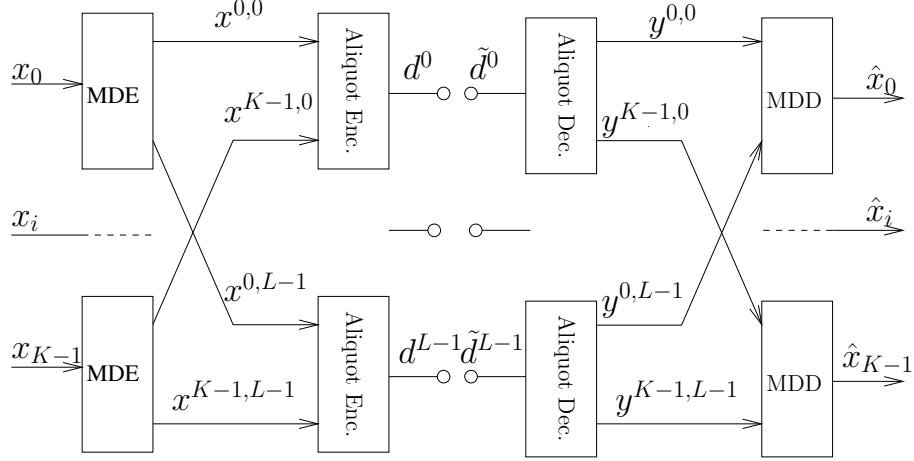


Figure 2.8: Framework of multiview coding with tree-structured-MDC. *MDE* represents a multiple description encoder, while *MDD* represents its multiple description decoder pair.

part descriptions are created, which may be grouped into L number of K -aliquot camera views based on their *offset* l and *quantizer* resolution L . The grouping seen in Figure 2.8 encodes the K correlated camera-view jointly and hence it is expected to achieve a certain compression. These L K -aliquot camera views are then input to L multiview video encoders, namely to the *Aliquot Enc.* of Figure 2.8, each of which may have different offsets and quantization parameters. Each of the L K -aliquot encoders will generate a bitstream independently, each of which may be transmitted via different wireless-channels or TCP-IP routes to the receiver. Consider that the pixels $\{50, 45, 39, 64\}$ originally from $K = 4$ views, respectively, are encoded by the HOCT-MDC coding tree of Figure 2.1. Then $\{50, 45, 39, 64\}$ may be encoded by the HOCT-MDC coding system of Figure 2.8, as seen in Figure 2.9 for generating $L = 4$ descriptions.

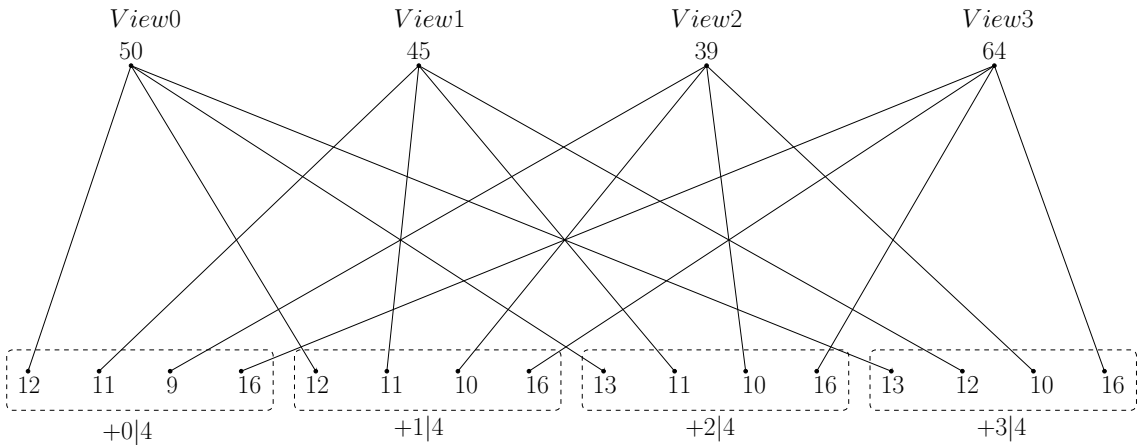


Figure 2.9: The pixels $\{50, 45, 39, 64\}$ originally from $K = 4$ views are encoded by the HOCT-MDC coding tree of Figure 2.1.

The receiver employs L multiview video decoders, namely the *Aliquot Dec.* blocks of Figure 2.8, each of which reconstructs the K -aliquot descriptions from the L received bitstreams $\tilde{d}_0, \dots, \tilde{d}_{L-1}$, as seen in Figure 2.8. After the aliquot reconstruction stage, a total of $(K \times L)$ aliquot part descriptions are reconstructed. These $(K \times L)$ descriptions may be grouped into K L -aliquot part groups, which represent the K original camera views, that are then input to K multiple description decoders (MDD), each of which will reconstruct one of the original camera views $\hat{x}_l, 0 \leq l < K$.

2.5 Simulation Results

This section evaluates the performance of both the proposed T-MDC (HOCT-MDC, BT-MDC) scheme as well as that of the entire system using MVC invoking the T-MDC. Simulcasting of duplicated single description (SD)¹ [147, 148] provides the lower bound of MDC and may always be employed for practical video streaming applications to transmit the same bitstream via different channels and routes in order to combat the packet loss events imposed by network congestion. In this section, we compare our system's performance to simulcast-SDC. Note that if a quantizer resolution of $Q = L = 1$ is used, then multiple description coding degenerates to conventional single description coding (SDC). We will employ the H.264 video codec for aliquot compression and the quantizer parameter (QP) values used in this section are those of the H.264 standard [14]. Furthermore, peak signal-to-noise ratio (PSNR) of luminance is used to qualify the video quality. In order to simplify our discourse, let us now stipulate the notation G denoting the total number of generated descriptions, as well as G_a denoting the number the descriptions available.

2.5.1 Multiple Description Codec Performance

In this section, we characterize the performance of our proposed MDC schemes. In all the simulations 45-frame *Akiyo* video sequence represented in (176×144) -pixel quarter common intermediate format (QCIF) is input to the HOCT-MDC codec. The JM/AVC 17.2 H.264 scheme was used for encoding the aliquot part descriptions into bitstreams and the rate distortion optimization (RDO) option of H.264 was enabled. Furthermore, the Intra-frame I refresh period was set to 15 and both the predicted frame (P-frame) and bidirectional frames (B-frame) were also enabled. The scanning rate expressed in frame per second (FPS) was set to 15. These parameters jointly determine the bitrate.

¹In simulcasting of SD, the original video is encoded into conventional single bitstream. Then multiple duplicated copies of the SD bitstream are transmitted to the receiver via multiple routes.

Firstly, we benchmark the proposed HOCT-MDC against the lower bound simulcast-SDC-H.264. The rate-distortion performance is qualified in Figure 2.10, where we observe that upon increasing the bitrate, HOCT-MDC increasingly outperforms simulcast-SDC-H.264, albeit naturally its performance saturates at the upper bound of Y-PSNR. Quantitatively HOCT-MDC using $G = 16$ descriptions outperforms simulcast-SDC-H.264 by 2 dB at 450 kbps, because the reconstruction error of the aliquot part descriptions becomes lower upon increasing the accuracy of high frequency information.

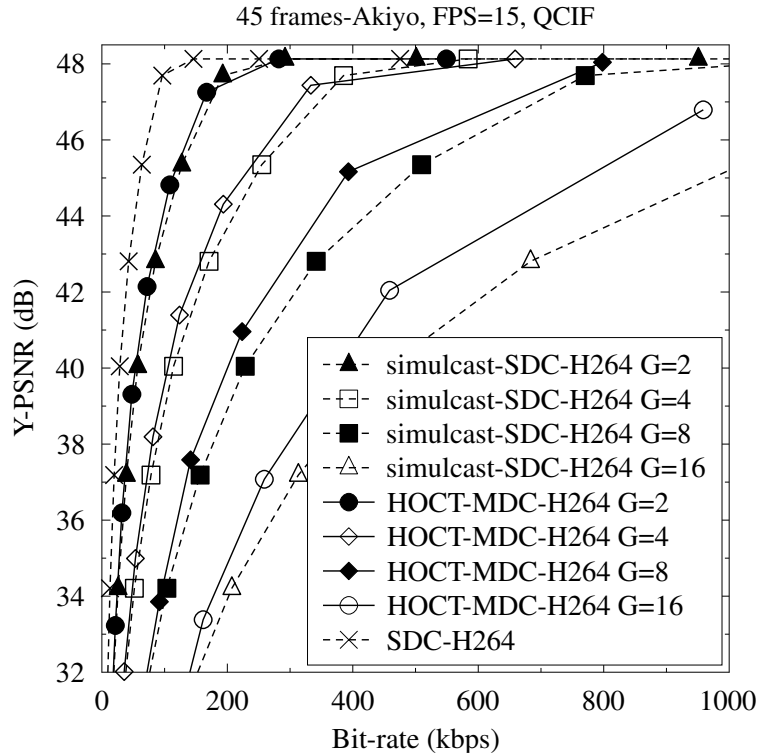


Figure 2.10: Comparison of HOCT-MDC-H.264 using $G = L = 2, 4, 8, 16$ and simulcast-SDC-H.264 using $G = L = 2, 4, 8, 16$. The schematic of Figure 2.4 was used.

Figure 2.11 characterizes the rate-distortion performance when different number of descriptions are available at the receiver. An important point to mention is that the aliquot part descriptions were chosen by maximizing the distance of description offsets. For example, we chose aliquot descriptions associated with the offsets of $l = 0, 2$ for $G = 4, G_a = 2$. As shown in Figure 2.11, Y-PSNR increases gradually with the number of available descriptions increasing. Furthermore, except for the curves recorded when all the G aliquot descriptions were available, there is a PSNR upper bound for the curves, which is jointly determined by the number of available descriptions G_a and the quantizer $Q = 4$. For example, when $G_a = 1$ aliquot description is available, Y-PSNR increases slower upon increasing the bitrate, because some

high frequency information is removed by the quantizer. One reconstructed frame of *Akiyo* is displayed in Figure 2.12.

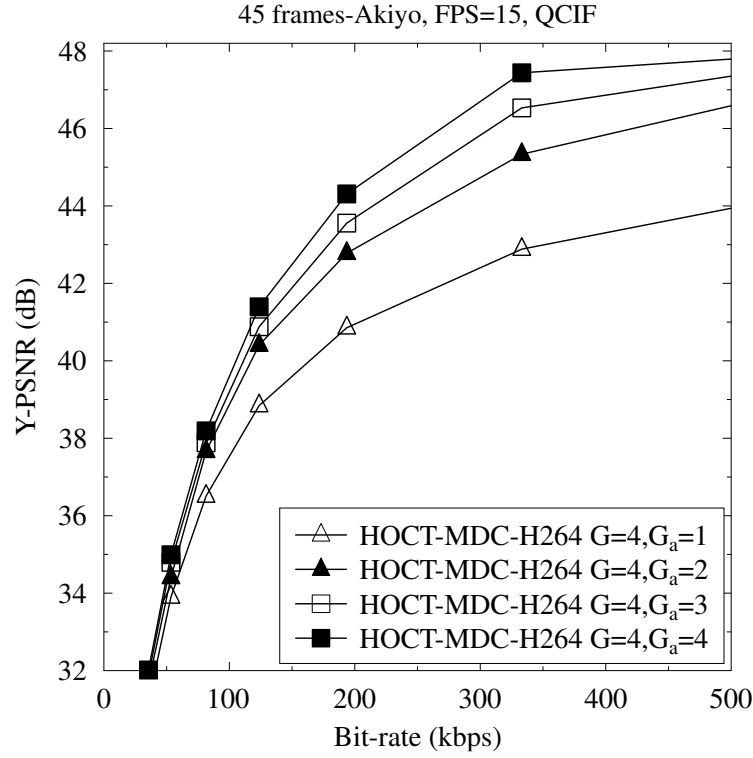


Figure 2.11: Rate-distortion performance of HOCT-MDC-H.264 for $L = 4, G = 4, G_a = 1, 2, 3, 4$. The schematic of Figure 2.4 was used.



Figure 2.12: Akiyo frame with $G = 8$ when $QP=20, L=12$ for HOCT-MDC-H.264. The schematic of Figure 2.4 was used.

Below, we will characterize the rate-distortion performance of BT-MDC, as portrayed in Figure 2.5 and Figure 2.13. The performance curves of BT-MDC with $G = 5$ and different G_a are displayed in Figure 2.13. Since the 5 aliquot descriptions are of unequal importance, the Y-PSNR is measured by averaging all descriptions combinations.

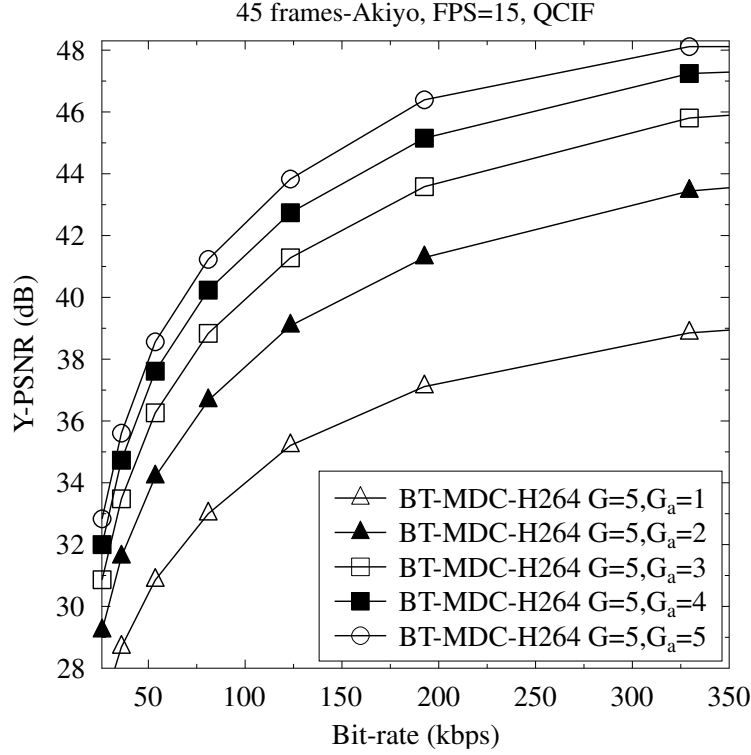


Figure 2.13: Rate-distortion performance of BT-MDC-H.264 with $G = 5$ when different descriptions are available. The coding tree of Figure 2.5 was used.

As shown in Figure 2.14, the scheme outperforms both simulcast-SDC-H.264 using $G = 5$ by about 4 dB at a bitrate of 200 kbps and HOCT-MDC using $L = 4$ by 2.5 dB at a bitrate of 125 kbps. The reason for the superiority of BT-MDC is that the BT-MDC scheme generates aliquot part descriptions of unequal importance, where the less important descriptions carry less high-frequency information. This property statistically decreases the average correlation amongst all the BT-MDC aliquot descriptions.

2.5.2 Performance of MVC with T-MDC

In this section, we characterize the performance of our proposed MDC-MVC scheme, employing the left 8 of 16 views of the 100-frame *Leaving-Laptop* sequence in (1024×768) -pixel resolution. The H.264 JMVC scheme is employed as the Aliquot codec in Figure 2.8 for encoding $K = 8$ -aliquot camera views into $K \times L$ bitstreams. Moreover, the Intra-frame I refresh period was set to 15 and the frame scanning rate per second (FPS) was set to 16.67. This facilitates the evaluation of the bitrate, which can be adjusted by modifying the quantization parameters (QP).

The comparison of the rate-distortion performances of the HOCT-MDC-H.264-MVC and simulcast-SDC-H.264-MVC recorded for $L = 2, L = 4$ is displayed in

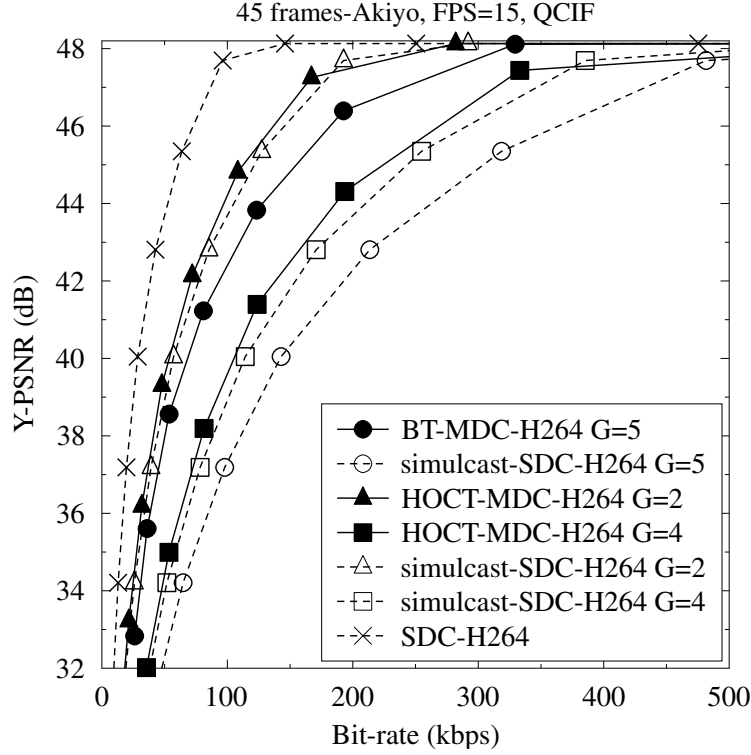


Figure 2.14: Comparison of BT-MDC-H.264 using $G = 5$, HOCT-MDC-H.264 using $G = 2, 4$ and simulcast-SDC-H.264 using $G = 5$. The schematic of Figure 2.4 was used.

Figure 2.15, where we observe that upon increasing the bitrate, HOCT-MDC-H.264-MVC increasingly outperforms simulcast-SDC-H.264-MVC. Quantitatively for $G = 4$ HOCT-MDC-H.264-MVC outperforms simulcast-SDC-H.264-MVC by 0.7 dB at 2×10^5 kbps, because the reconstruction error of the aliquot part descriptions becomes lower upon increasing the accuracy of high frequency information.

Below, we will investigate the BT-MDC-H.264-MVC scheme for $G = 5$ descriptions, which was outlined in Figure 2.5. The attractive rate-distortion performance is illustrated in Figure 2.16. The scheme outperforms simulcast-SDC-H.264-MVC using $G = 5$ by about 1.5 dB at a bitrate of 10^5 kbps. Alternatively we may argue based on Figure 2.16 that BT-MDC-H.264-MVC roughly halves the bitrate required for achieving 42 dB Y-PSNR.

2.6 Summary and Conclusions

A new tree-structure based MDC scheme was proposed for the flexible generation of multiple video descriptions, which may be compressed by arbitrary video codecs. Two specific structures were analyzed in detail, namely HOCT-MDC and BT-MDC. Since the BT-MDC scheme is capable of generating descriptions of unequal importance,

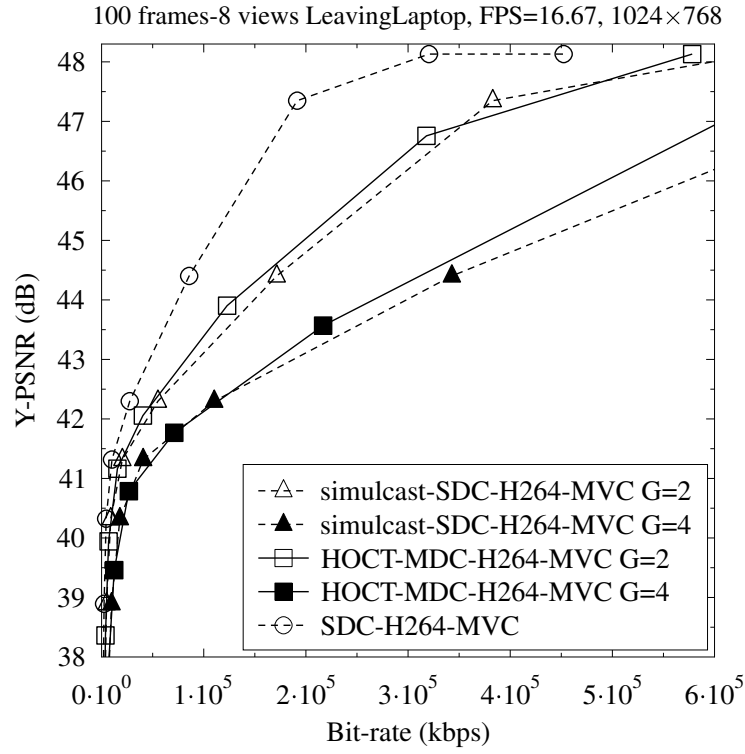


Figure 2.15: Comparison of HOCT-MDC-H.264-MVC using $G = 2, 4$ and simulcast-SDC-H.264-MVC using $G = 2, 4$. The schematic of Figure 2.8 was used.

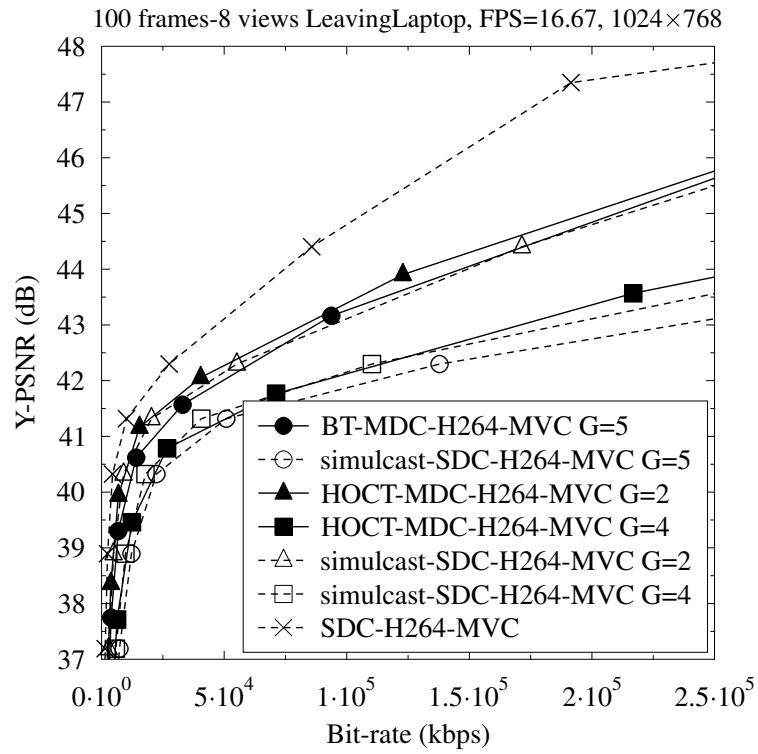


Figure 2.16: Comparison among BT-MDC-H.264-MVC using $G = 5$, HOCT-MDC-H.264-MVC using $G = 2, 4$ and simulcast-SDC-H.264-MVC using $G = 5$. The schematic of Figure 2.8 was used.

	$G = 2, 96 \text{ kbps}$	$G = 4, 181 \text{ kbps}$	$G = 8, 340 \text{ kbps}$	$G = 16, 660 \text{ kbps}$
HOCT-MDC	44 dB	44 dB	44 dB	44 dB
simulcast-SDC-H.264	43.6 dB	43.3 dB	42.8 dB	42.6 dB
gain	0.4 dB	0.7 dB	1.2 dB	1.4 dB

Table 2.1: Performance comparison of the HOCT-MDC scheme and the simulcast-SDC-H.264 scheme, *Akiyo* sequence. The schematic of Figure 2.4 was used.

	$G = 2, 1.96 \cdot 10^5 \text{ kbps}$	$G = 4, 3.75 \cdot 10^5 \text{ kbps}$
HOCT-MDC	45 dB	45 dB
simulcast-SDC-H.264	44.78 dB	44.66 dB
gain	0.22 dB	0.34 dB

Table 2.2: Performance comparison of the HOCT-MDC-H.264 scheme and the simulcast-SDC-H.264 scheme, *Leaving-Laptop* sequence. The schematic of Figure 2.4 was used.

which correspondingly reduces the correlation amongst the descriptions, it outperforms the HOCT-MDC scheme in the absence of packet loss events. Furthermore, diverse multiview schemes were detailed.

The proposed HOCT-MDC scheme is benchmarked against the simulcast-SDC-H.264 scheme in Figure 2.10 using the QCIF-sized *Akiyo* sequence. The HOCT-MDC scheme increasingly outperforms the simulcast-SDC-H.264 upon increasing the bitrate. Specifically, our performance comparison of the HOCT-MDC scheme and of the simulcast-SDC-H.264 scheme is summarized in Table 2.1 at various bitrates. The BT-MDC scheme is characterized in Figure 2.14, which outperforms the simulcast-SDC-H.264 scheme by about 4.8 dB in terms of its PSNR at a bitrate of 126 kbps, when generating $G = 5$ descriptions. Figure 2.14 demonstrates that the BT-MDC scheme significantly outperforms both the HOCT-MDC and the simulcast-SDC-H.264 scheme.

The proposed HOCT-MDC-MVC scheme was benchmarked against the simulcast-SDC-H.264-MVC scheme in Figure 2.15 using the *Leaving-Laptop* multiview sequence. Similar to the monoscopic video, scenario of the HOCT-MDC-MVC increasingly outperforms simulcast-SDC-H.264-MVC upon increasing the bitrate. Specifically, the performance comparison of the HOCT-MDC-MVC scheme and of the simulcast-SDC-H.264-MVC scheme is summarized in Table 2.2. The BT-MDC-MVC

scheme using the *Leaving-Laptop* multiview sequence is benchmarked in Figure 2.14, which outperforms the simulcast-SDC-H.264-MVC scheme by about 2.2 dB in terms of its PSNR at a bitrate of $2.1 \cdot 10^5$ kbps, when generating $G = 5$ descriptions. Figure 2.14 suggests that the BT-MDC-MVC scheme achieves the best performance among the benchmarked systems.

From Figure 2.10 to Figure 2.14, we may arrive at the conclusion that the HOCT-MDC scheme outperforms simulcast-SDC-H.264 in both the monoscopic and multiview video scenarios, while the BT-MDC scheme outperforms the HOCT-MDC scheme. This trend is as expected, since each of the simulcast-SDC descriptions carries the entire original video sequence, while each description in the HOCT-MDC carries a coarse version of the original video along with its unique high frequency information. It may be concluded that the BT-MDC scheme creates descriptions exhibiting the lowest correlations amongst the three schemes.

Finally, our proposed T-MDC is capable of generating numerous structured descriptions by appropriately configuring the generation tree. Deeper investigations may be carried out for the sake of finding the optimal configurations.

Inter-Layer Coded Unequal Error Protection for Layered Video Transmission

3.1 Introduction

Layered video compression techniques were reviewed in Section 1.1.3. In layered video transmission, such as the scheme SVC [15] for example, different layers may have different importance. When the BL is corrupted or lost due to channel impairments, the corresponding ELs cannot be utilized by the video decoder and must be discarded by the receiver even if they are correctly decoded.

Hence it is intuitive to employ UEP [66] for protecting the more important bits by stronger FEC codecs than the less important bits, in order to achieve an improved reconstructed video quality. A number of contributions, such as [49, 56, 59, 60], have been devoted to UEP aided video communications, where FECs, such as RSC codes, LT codes and Reed-Solomon codes may be employed. In the traditional UEP schemes conceived for layered video communication, variable-rate FEC was invoked for the different layers. When the BL is corrupted or lost, the ELs also have to be discarded, regardless whether they are perfectly received or not, which implies that the transmission power assigned to the ELs was wasted. In [81, 62] performs channel encoding was performed right across the BL and the ELs at the transmitter and the receiver invoked the parity bits of the ELs for assisting in correcting the errors within the BL, as we will detail in Section 3.2.

Motivated by these advances, we developed a bit-level inter-layer operation aided FEC (IL-FEC) scheme [139, 141, 75] relying on a systematic FEC code in Section 3.2,

where the systematic bits of the BL were implanted into the ELs. At the receiver, the above-mentioned implanted bit of the ELs may be utilized for assisting in decoding the BL. The IL-FEC technique of [139] was also combined with the UEP philosophy for the sake of further improving the attainable system performance. Our proposed technique is significantly different with the LA-FEC philosophy proposed in [81,62], as detailed below conceiving the following aspects. Firstly, our technique is proposed for layered video communication over wireless channels, while the LA-FEC of [81,62] is proposed for the BEC. Secondly, IL-FEC invokes the soft decoding aided channel codecs, such as an RSC code, while the LA-FEC of [81,62] considered a hard-decoding based Raptor codec. In this context, we note that Raptor codes are less suitable for low-delay lip-synchronized interactive multimedia communications, whilst our scheme is readily applicable. Furthermore, it is important to note that the LA-FEC cannot be readily applied in soft decoding aided channel codecs. Finally, IL-FEC implants the systematic bits of the BL into the ELs, while the LA-FEC [81,62] generates the parity bits across the BL and ELs.

At the time of writing, multimedia content is evolving from traditional content to a range of rich, heterogeneous media content, such as traditional TV, streaming audio and video as well as image and text messaging. Furthermore, in the current era of smart phones, mobile TV has become an appealing extension of terrestrial TV. Additionally, in order to meet the challenging performance requirements in bandwidth-constrained environments, multiple input multiple output (MIMO) systems constitute a promising transmission solution. Layered steered space-time codes (LSSTC) [137, 138] combine the benefits of the vertical Bell Labs space-time (VBLAST) scheme [149], of space-time block codes [150] and of beamforming [151]. Hence LSSTCs are invoked in Section 3.2 for providing both a diversity gain to achieve a high BER performance in mobile environments as well as for attaining a multiplexing gain in order to maintain a high data rate. In this treatise, we propose a system for transmitting an IL-FEC encoded compressed video bitstream with the aid of a LSSTC transceiver structure (IL-FEC-LSSTC) for mobile TV broadcasting. This scheme may be considered as an evolution of the traditional UEP schemes exemplified by [56, 59]. EXIT charts [133] are employed in Section 3.3 for analyzing the performance of our proposed system. Moreover, for providing a rich set of design guidelines multiple FECs, including an RSC codec, a turbo codec and an SECCC are employed for transmission of both partitioned H.264 video¹ and of SVC-compressed video in Sections 3.5 and 3.6. These FECs will be configured to operate at diverse coding rates for benchmarking the proposed system.

¹In the data partitioning mode (PM) of the H.264 video codec, the type B and type C partitions will be utilized for protecting the type A partition.

The code rates of the proposed IL arrangements will be determined in Sections 3.5 and 3.6 empirically. However, in practical scenarios [152], different configurations of video codecs and different video sequences may have different characteristics, which may require different channel coding rates for achieving the best system performance. Hence we will propose a technique for finding the optimized coding rates “on-the-fly” at the transmitter in Section 3.4, which optimizes the bit-level IL-FEC coded system performance [153]. Specifically, we find the coding rates achieving the minimum video quality distortion with the aid of the mutual information (MI) between the log-likelihood ratios (LLRs) and the corresponding video bits, where the soft-decoding metric of the FEC codec and of the demodulator wireless transceivers are characterized by lookup tables (LUTs), since these cannot be readily characterized theoretically. Finally, the widely employed SVC-H.264 codec will be utilized for the sake of benchmarking our proposed system in Section 3.7. Our proposed technique is significantly different from the existing contributions proposed in [47, 56, 57, 59, 60, 62] in the following aspects. Firstly, our technique is proposed for layered video communication over wireless channels, while the contributions in [56, 59, 60, 62] focus on networks, which inflict packet-loss events or on networks modeled by the BEC. Secondly, IL-FEC relies on soft-decoding channel aided decoders, such as an RSC code, while [56, 59, 60, 62] considered hard-decoded RS or fountain codes. Moreover, we study the coding rate optimization problem of an IL-FEC aided UEP system. Finally, we propose a solution employing near-continuous coding rates, while only three protection rates - namely low-order, medium-order and high-order protection - were utilized in [47, 57], which limits the system’s performance.

3.2 Inter-Layer FEC Coded Layered Video Streaming

In this section, we will briefly introduce the architecture of the inter-layer FEC scheme [139] conceived for layered video transmission over our LSSTC scheme in mobile TV applications. As seen in Figure 3.2, N_t transmit antennas and N_r receive antennas are used in our LSSTC structure. The system’s structure is displayed in Figure 3.1, where data-partitioning aided H.264 [14] encoding and LSSTC transmission are employed, while the structures of the variable node decoder (VND) and check node decoder (CND) [134] are further detailed in Figure 3.3. Both the VND and CND blocks may accept a maximum of three soft information inputs and generate a maximum of three soft information outputs with the goal of iteratively exploiting all IL dependencies amongst the FEC coded layers A, B and C. Specifically, assuming that u_1 , u_2 and $u_3 = u_1 \oplus u_2$ are random binary variables, the action of the VND of Figure 3.3a sums two LLR inputs for generating a more reliable LLR output, which may be formulated

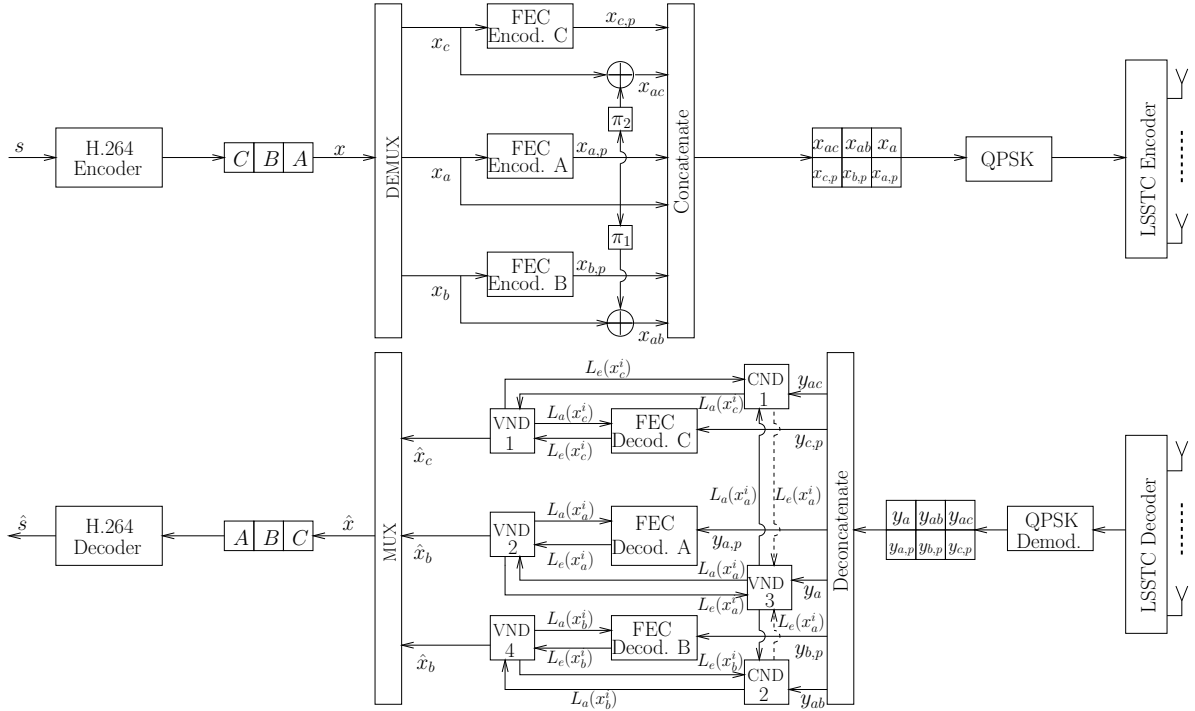


Figure 3.1: Bit-level IL-FEC encoding architecture of H.264 data partitioning mode coded video, where the LSSTC structure is shown in Figure 3.2.

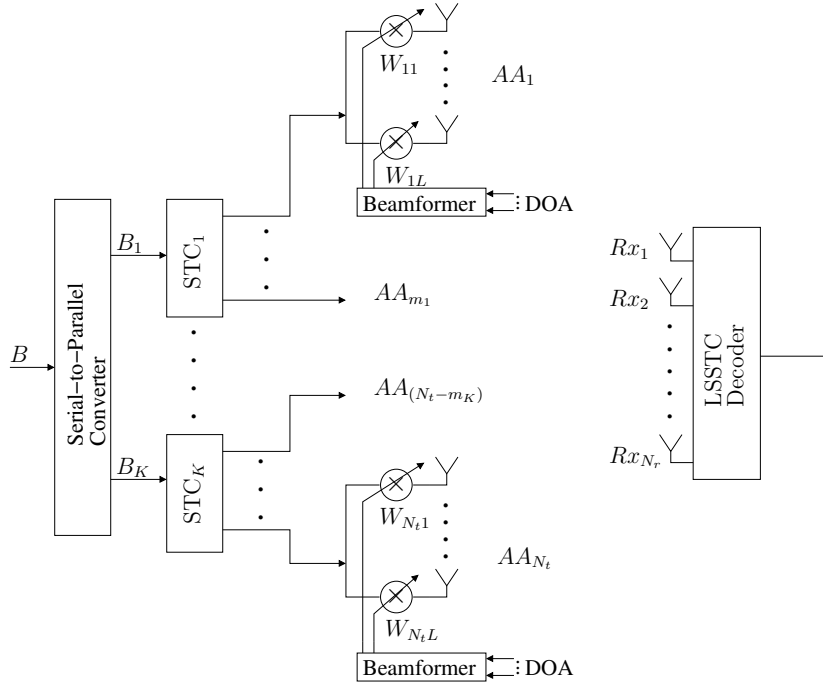


Figure 3.2: The structure of LSSTC [137], which combines the benefits of vertical Bell Labs space-time scheme, space-time block codes and beamforming. N_t transmit antennas and N_r receive antennas are used.

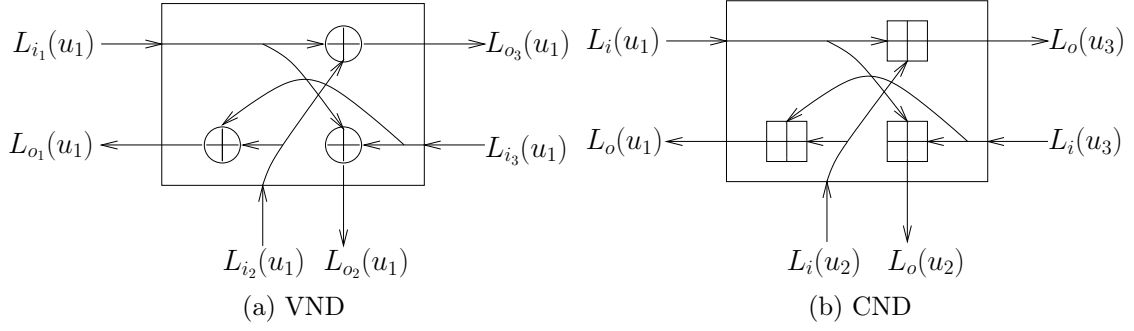


Figure 3.3: The structure of VND and CND, where \oplus and \boxplus indicate the addition and boxplus operation, respectively. $L_i(\cdot)$ and $L_o(\cdot)$ indicate the input and output LLR, respectively.

as $L_{o_3}(u_1) = L_{i_1}(u_1) + L_{i_2}(u_1)$. The boxplus operation of $L(u_3 = u_1 \oplus u_2) = L(u_1) \boxplus L(u_2)$ [154] may be utilized for deriving the confidence of the bit u_3 , given that the confidence of the bits u_1 and u_2 is known. Specifically, the boxplus operation \boxplus is defined as follows [155]

$$\begin{aligned}
 L(u_1) \boxplus L(u_2) &= \log \frac{1 + e^{L(u_1)} e^{L(u_2)}}{e^{L(u_1)} + e^{L(u_2)}} \\
 &= \text{sign}[L(u_1)] \cdot \text{sign}[L(u_2)] \cdot \min[|L(u_1)|, |L(u_2)|] \\
 &\quad + \log[1 + e^{-|L(u_1)+L(u_2)|}] - \log[1 + e^{-|L(u_1)-L(u_2)|}].
 \end{aligned} \tag{3.1}$$

In contrast to the above-mentioned VND function, the CND operation of Figure 3.3b may be formulated as $L_o(u_3) = L_i(u_1) \boxplus L_i(u_2)$ for extracting the confidence of the bit u_3 , given the LLR input of the bits u_1 and u_2 .

In Section 3.2.1, we first detail the techniques employed at the transmitter. Then, our inter-layer H.264 decoding techniques and the LSSTC receiver will be illustrated in Section 3.2.2, with special emphasis on how the VND and the CND exchange their inter-layer redundancy for improving the overall performance of the system. We assume that A is the BL and B, C are the corresponding dependent layers, but both partition B and C can be utilized for protecting the partition A . In Section 3.2.1 and 3.2.2, we assume that all the layers A, B and C contain n bits for the sake of convenient explanation, while in Section 3.2.4 we extend our algorithm to the more general scenario, where the layers have unequal length. Finally, Section 3.2.5 discusses the overheads imposed by our proposed IL technique, including its delay, complexity and its FEC-redundancy.

	Description
s	The source video signal
x	The H.264 encoded bitstream
x_a	The bit sequence containing partition A
x_b	The bit sequence containing partition B
x_c	The bit sequence containing partition C
$x_{a,p}$	The parity bits of partition A
x_{ab}	The check bits of partition A and B
$x_{b,p}$	The parity bits of partition B
x_{ac}	The check bits of partition A and C
$x_{c,p}$	The parity bits of partition C

Table 3.1: Symbol definitions for the transmitter of Figure 3.1.

3.2.1 Transmitter Model

The symbols characterizing the transmitter of Figure 3.1 are defined in Table 3.1. At the transmitter, the video source signal s is compressed using the data partitioning mode of the H.264 encoder, generating the bitstream x carrying the partitions A , B and C . Then the output bitstream is de-multiplexed into three bitstreams by the *DEMUX* block of Figure 3.1, namely into streams A , B and C , carrying the A , B and C partitions of all slices. The resultant binary sequences are x_a , x_b and x_c , representing the BL A and ELs B and C , respectively, as shown in Figure 3.1. Then the resultant three layers are encoded as follows:

1. The BL bit sequence x_a representing A will be encoded by the FEC encoder A of Figure 3.1, which results in the encoded bits containing the systematic bits x_a and parity bits $x_{a,p}$.
2. The bit sequence of the EL x_b representing B will firstly be encoded into the systematic bits x_b and the parity bits $x_{b,p}$ by the FEC encoder B. Then the XOR operation will be utilized for implanting the systematic information of x_a into the systematic information of x_b without changing the parity bits of the B partition $x_{b,p}$. Specifically, the implantation process results in the check bits $x_{ab}^i = x_a^i \oplus x_b^i$. After this procedure, both the check bits x_{ab}^i and the parity bits $x_{b,p}$ are output.

	x_a	$x_{a,p}$	x_{ab}	$x_{b,p}$	x_{ac}	$x_{c,p}$
A	✓	✓	✓		✓	
B			✓	✓		
C					✓	✓

Table 3.2: Information of partitions A, B, C are contained in different bit sequences, where ✓ indicates “is contained in”.

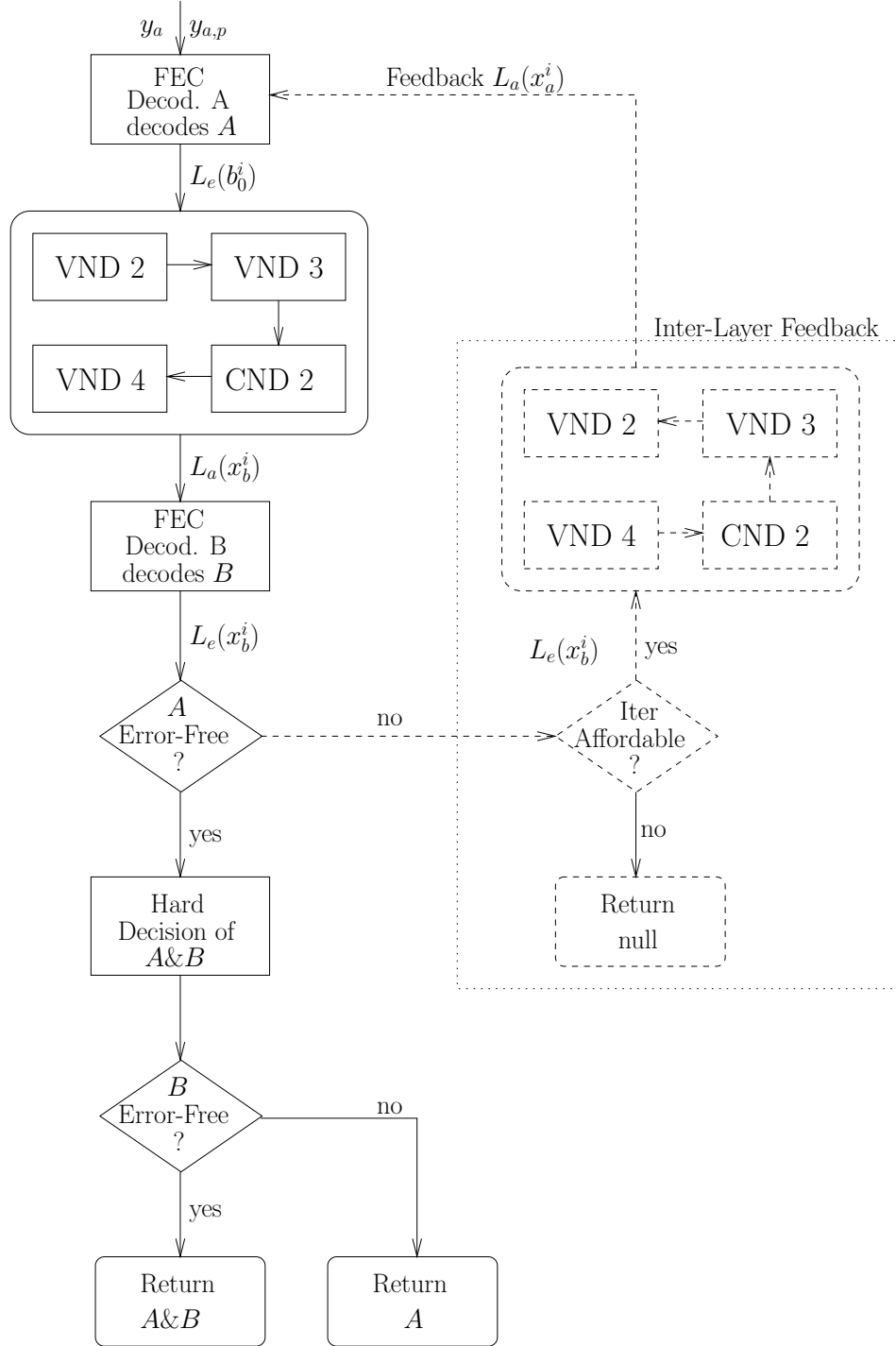
- Similar to the encoding process of the B partition, the bit sequence of the EL x_c representing C will firstly be encoded into the systematic bits x_c and the parity bits $x_{c,p}$ by the FEC encoder C of Figure 3.1. Then the XOR operation will be utilized for implanting the systematic information of x_a into the systematic information of x_c without changing the parity bits of the C partition $x_{c,p}$. Specifically, the implantation process results in the check bits $x_{ac}^i = x_a^i \oplus x_c^i$. After this procedure, both the check bits x_{ac}^i and the parity bits $x_{c,p}$ are transmitted.

Finally, the bit sequences x_a , $x_{a,p}$, x_{ab} , $x_{b,p}$, x_{ac} and $x_{c,p}$ are concatenated into a joint bitstream for transmission, which contain the information of the partitions A, B and C, as shown in Table 3.2. Note however that the layers x_a and x_b , x_c may contain a different number of bits. Again, the algorithm designed for this scenario will be detailed in Section 3.2.4. Additionally, the interleavers π_1 and π_2 are employed for interleaving the BL x_a , before its XOR-based implantation into the ELs x_b and x_c .

Following the IL-FEC encoding procedure, the resultant bits are modulated by the quadrature phase-shift keying (QPSK) modulator of Figure 3.1 and then transmitted over the LSSTC based MIMO transmitter architecture. Specifically, the transmission structure shown in Figure 3.1 has $N_t = 4$ transmit antennas, which are spaced sufficiently for apart in order to encounter independent fading. The receiver is also equipped with $N_r = 4$ receive antennas, where the LSSTC system used is characterized by a diversity order of 2 and multiplexing order of 2. Hence the LSSTC used is capable of providing twice the data rate of a single antenna system, while achieving a diversity order of two.

3.2.2 Receiver Model

In this section, we exemplify the IL decoding process using BL A and EL B, while the IL decoding process of BL A and EL C is similar. The symbols characterizing

Figure 3.4: The flow chart for inter-layer aided FEC decoding of BL A and EL B .

	Description
\hat{s}	The estimated video signal
\hat{x}	The estimated H.264 encoded bitstream
\hat{x}_a	The estimated bit sequence for partition A
\hat{x}_b	The estimated bit sequence for partition B
\hat{x}_c	The estimated bit sequence for partition C
y_a	The received soft systematic bits of partition A
$y_{a,p}$	The received soft parity bits of partition A
y_{ab}	The received soft check bits of partition A and B
$y_{b,p}$	The received soft parity bits of partition B
y_{ac}	The received soft check bits of partition A and C
$y_{c,p}$	The received soft parity bits of partition C
$L_a(x_a)$	<i>A-priori</i> LLR of bit sequence x_a
$L_e(x_a)$	Extrinsic LLR of bit sequence x_a
$L_a(x_b)$	<i>A-priori</i> LLR of bit sequence x_b
$L_e(x_b)$	Extrinsic LLR of bit sequence x_b
$L_a(x_c)$	<i>A-priori</i> LLR of bit sequence x_c
$L_e(x_c)$	Extrinsic LLR of bit sequence x_c

Table 3.3: Symbol definitions for the receiver of Figure 3.1.

the signals at the receiver of Figure 3.1 are defined in Table 3.3. At the receiver², the LSSTC decoding is performed [138]. Then the resultant soft signal will be demodulated by the QPSK demodulator, which generates the log-likelihood ratios (LLR). The LLR information contains the systematic information y_a , y_{ab} , y_{ac} and the parity information $y_{a,p}$, $y_{b,p}$ and $y_{c,p}$, for the A , B and C partitions, respectively. Following the demodulator, the IL-FEC decoder of Figure 3.1 is invoked for exchanging extrinsic information across the three layers. The IL aided FEC decoding process is illustrated by the flow-chart of Figure 3.4. Firstly, the FEC decoder A will decode the received information y_a and $y_{a,p}$ for estimating the LLRs of the bits x_a of the BL A. Then, the resultant extrinsic LLR information of BL A will be input to the

²The deinterleavers π^{-1} and π^{-2} are ignored at the receiver for the sake of simplifying the system architecture.

"VND2-VND3-CND2-VND4" block of Figure 3.4 for extracting the *a-priori* LLRs $L_a(x_b^i)$ ³ of EL B , which is carried out by following the processing of the LLRs in the VND2, VND 3, CND 2 and VND 4 components of Figure 3.3. Specifically, the "VND2-VND3-CND2-VND4" block of Figure 3.4 performs the following operations step-by-step:

1. VND 2 generates the information of BL A for VND 3. The inputs of the VND 2 block are constituted by the soft information $L_e(x_a^i)$ generated by the FEC decoder A and the soft information $L_a(x_a^i)$ generated by the VND 3 block. The output of the VND 2 block is the soft information of A . The output can be readily derived as detailed in Figure 3.3a. The extrinsic LLR $L_e(x_a^i)$ generated by the FEC decoder A is input to the VND 2 block of Figure 3.1, which extracts the extrinsic LLR information $L_e(x_a^i)$ and forwards it to the VND 3 block of Figure 3.1. Since VND 2⁴ has two input branches, it simply duplicates the soft information $L_e(x_a^i)$.
2. VND 3 generates the information of BL A for CND 2. The inputs to VND 3 block are constituted of the soft information $L_e(x_a^i)$ generated by the VND 2 block and the channel information y_a . The output of the VND 3 block is the soft information of A . The output can be readily derived as detailed in Figure 3.3a. The VND 3 block of Figure 3.1 extracts the extrinsic LLR information $L_e(x_a^i)$ and forwards it to the CND 2 block of Figure 3.1.
3. CND 2 generates the information of layer B for VND 4. The inputs of the CND 2 block are the soft check information y_{ab} received from the channel, the soft information $L_e(x_a^i)$ of BL A generated by VND 3 and the soft information $L_e(x_b^i)$ of EL B generated by the VND 4 of Figure 3.1. The output of CND 2 is the soft information of EL B $L_a(x_b^i)$. The outputs can be readily derived as detailed in Figure 3.3b. The LLR information $L_e(x_a^i)$ and the received check information y_{ab} is input to the CND 2 block of Figure 3.1 for extracting the LLR information of the systematic bit x_b^i , namely the soft input $L_a(x_b^i)$ of VND 4.
4. VND 4 generates the information of EL B for FEC decoder B. The inputs to the VND 4 block are the soft information $L_a(x_b^i)$ gleaned from CND 2 and the soft information $L_e(x_b^i)$ generated by FEC decoder B. The output of VND 4 is the soft information of layer B . The LLR information $L_a(x_b^i)$ extracted by

³As usual, the subscripts "a" and "e" in L_a and L_e stand for the apriori information and extrinsic information [121], respectively.

⁴All the VNDs of Figure 3.1 have two input branches and three output branches, resulting in a duplication process for two of the output branches. Note that two LLR inputs will be summed by each VND for the third output branch, which outputs the final *a-posteriori* LLR for the estimation of \hat{x}_a , \hat{x}_b and \hat{x}_c .

the CND 2 is input to the VND 4 block of Figure 3.1, which extracts the LLR information $L_a(x_b^i)$ input to the FEC decoder B of Figure 3.1.

Then, the FEC decoder B of Figure 3.4 will decode the EL B with the aid of the resultant *a-priori* LLR $L_a(x_b^i)$ and of the soft parity information received from the channel, namely $y_{b,p}$ of Figure 3.1. Afterwards, the classic cyclic redundancy check (CRC) is invoked for detecting, whether the recovered BL A is error-free or not, as shown in Figure 3.4. This check results in two possible decoding processes, as shown in Figure 3.4 and described as follows:

3.2.2.1 With Inter-Layer Feedback

When the bits x_a of the BL are not successfully decoded, the iterative IL technique will be activated for exploiting the extrinsic information of BL A fed back from the FEC decoder B. In this case, both the solid lines and the dashed lines shown in the decoder of Figures 3.1 and 3.4 will be activated. More explicitly, the "VND4-CND2-VND3-VND2" block of Figure 3.4 will be utilized for extracting the extra LLR information $L_e(x_a^i)$ for BL A based on both the extrinsic LLR $L_e(x_b^i)$ and the soft check information y_{ab} . Generally, the "VND4-CND2-VND3-VND2" block of Figure 3.4 represents a process similar to that of the "VND2-VND3-CND2-VND4" block of Figure 3.4. After this stage, improved *a-priori* information is generated for the BL A , which concludes the current IL decoding iteration. Afterwards, the receiver will return to the beginning of the flow chart shown in Figure 3.4. The iterative IL decoding process continues, until the affordable number of iterations is exhausted or the BL A is perfectly recovered, as shown in Figure 3.4.

3.2.2.2 Without Inter-Layer Feedback

When the BL A is successfully recovered, the layers A and B will be estimated by the hard decision block of Figure 3.4. Afterwards, the receiver may discard layer B , depending on whether it is deemed to be error-free or not by the CRC check. In this case, only the solid lines of Figures 3.1 and 3.4 will be activated.

Moreover, after decoding BL A , the recovered error-free hard bits x_a may be represented using infinite LLR values, indicating the hard bits 0/1, respectively. Then, the CND 2 process invoked for generating the LLR $L(x_b^i)$ shown in Figure 3.1

may be derived as follows using the boxplus operation

$$\begin{aligned}
L(x_b^i) &= L(x_a^i) \boxplus L(x_{ab}^i) \\
&= \text{sign}[L(x_a^i)] \cdot \text{sign}[L(x_{ab}^i)] \cdot \min[\infty, |L(x_{ab}^i)|] \\
&\quad + \log(1 + e^{-\infty}) - \log(1 + e^{-\infty}) \\
&= \text{sign}(\tilde{x}_a^i) \cdot L(x_{ab}^i),
\end{aligned} \tag{3.2}$$

where \tilde{x}_a^i is the modulated version of the bit x_a^i and the LLR input $L(x_{ab}^i)$ is obtained by soft demodulating the received signal y_{ab} .

Note that since the process of recovering y_b from y_{ab} expressed by Eq. (3.2) is essentially an LLR sign-flipping operation, it does not affect the absolute value of the LLR information of x_b . This implies that in this scenario our proposed IL technique is equivalent to the traditional UEP techniques, where layers A and B are encoded and decoded independently. Moreover, since BL A is decoded independently without feedback from EL B , the two layers are only decoded once, without any extra complexity imposed on the receiver. Additionally, in practical applications, BL A may be reconstructed immediately when it is received, without waiting for the arrival of the EL B .

In both of the above cases, if the decoded bit sequence \hat{x}_a of the BL is corrupted after the IL-FEC decoding stage of Figure 3.1, it will be dropped together with the ELs \hat{x}_b and \hat{x}_c . Otherwise they will all be forwarded to the H.264 decoder of Figure 3.1 for reconstructing the video signal \hat{s} .

Note that in the above description, we have considered decoding layers A and B only. The decoding of layer C is carried out in the same way but we have excluded it for the sake of simplifying our discussions.

3.2.3 Example of IL FEC Coding and Decoding

In this section, we further illustrate the IL-FEC encoding and decoding process through an example. The IL-FEC encoding process is exemplified in Figure 3.5, while the values of the related symbols are defined in Table 3.4. Specifically, the encoding process is listed as follows:

1. The source signals s are firstly encoded by the H.264 encoder generating the bitstream $x = 110101$, which is then split into $x_0 = 101$ and $x_1 = 110$, representing the BL L_0 and the EL L_1 , respectively.
2. The bitstream $x_0 = 101$ is encoded by FEC encoder 0 of Figure 3.5, which generates the parity bits $x_{0,p} = 111101$, while the bit sequence $x_0 = 101$ is

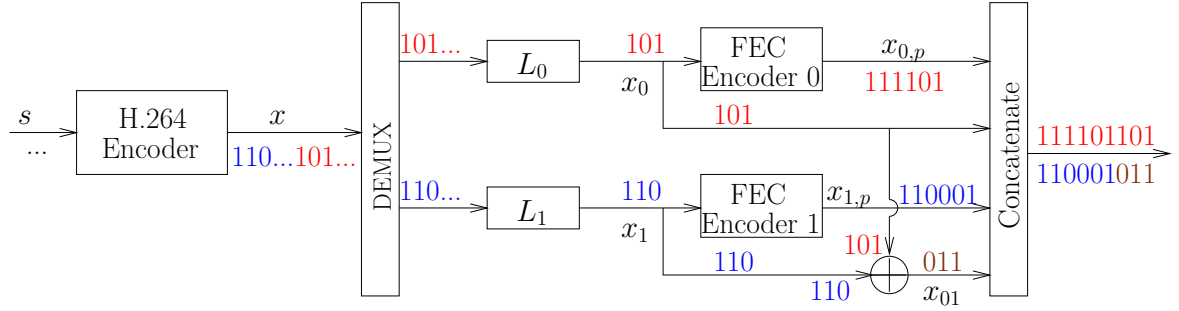


Figure 3.5: An encoding example of IL-FEC.

Symbol	Value	Symbol	Value
x	110101	$x_{0,p}$	111101
x_0	101	x_{01}	011
x_1	110	$x_{1,p}$	110001

Table 3.4: Symbols and related values for the transmitter of Figure 3.5.

directly forwarded as the systematic bits of the BL L_0 .

3. The bitstream $x_1 = 110$ is encoded by FEC encoder 1 of Figure 3.5, which generates the parity bits $x_{1,p} = 110001$.
4. Then the XOR operation is performed on the systematic bits of the BL $x_0 = 101$ and of the EL $x_1 = 110$ to generate the check bits $x_{01} = x_0 \oplus x_1 = 101 \oplus 110 = 011$.
5. Finally, the bit sequences $x_0 = 101$, $x_{0,p} = 111101$, $x_{01} = 011$, $x_{1,p} = 110001$ will be concatenated and transmitted as the IL-FEC encoding results of BL $x_0 = 101$ and EL $x_1 = 110$.

At the receiver, the soft information y_0 , $y_{0,p}$, y_{01} and $y_{1,p}$ may be received corresponding to the original transmitted signals x_0 , $x_{0,p}$, x_{01} , $x_{1,p}$, respectively. In the IL-FEC decoding process, FEC decoder 0 will attempt to recover x_0 by decoding $y_{0,p}$ and y_0 . Then, two decoding processes may be applied, depending on whether \hat{x} is successfully decoded or not. When x_0 is error-free, the decoding process operating without IL feedback will be applied, as detailed in Section 3.2.3.1. When x_0 is error-infested, the decoding process relying on IL feedback will be applied, as detailed in Section 3.2.3.2.

3.2.3.1 Operation Without IL Feedback

The IL-FEC decoding operating without IL feedback is exemplified in this section. In the scenario of Figure 3.6, the soft information $y_0 = -0.4, 0.6, -0.2$, $y_{0,p} =$

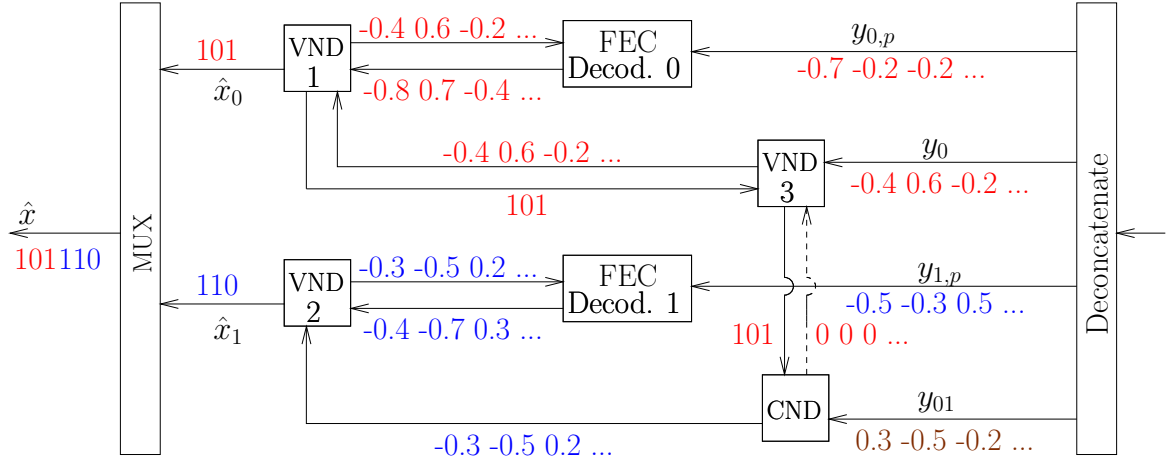


Figure 3.6: Decoding example of IL-FEC without IL feedback.

$-0.7, -0.2, -0.2, y_{01} = 0.3, -0.5, -0.2$ and $y_{1,p} = -0.5, -0.3, 0.5$ is received at the receiver. Since the VND of Figure 3.6 does not have any information related to the BL L_0 before the commencement of the IL-FEC decoding process, the information fed back from VND to VND 3 will be initialized as $L_a(x_0) = 0, 0, 0$. Then the decoding is performed as follows:

1. The *a-priori* information $L_a(x_0) = 0, 0, 0$ gleaned from the VND block of Figure 3.6 and $y_0 = -0.4, 0.6, -0.2$ will be input to VND 3 for the sake of generating the *a-priori* information $L_a(x_0) = -0.4, 0.6, -0.2$ for VND 1.
2. The *a-priori* information $L_a(x_0) = -0.4, 0.6, -0.2$ will be input to the FEC decoder 0 of Figure 3.6.
3. The soft information $L_e(x_0) = -0.8, 0.7, -0.4$ is then generated by the FEC decoder 0, which will be forwarded to VND 1 of Figure 3.6.
4. Based on the extrinsic LLR input $L_e(x_0) = -0.8, 0.7, -0.4$ gleaned from FEC decoder 0 and the *a-priori* LLR input $L_a(x_0) = -0.4, 0.6, -0.2$ received from VND 3, the VND 1 of Figure 3.6 becomes capable of generating the *a-posteriori* probability (APP) LLR $L(x_0) = -1.2, 1.3, -0.6$ of the BL, namely $x_0 = 101$.
5. At this stage, the BL can be estimated as $\hat{x}_0 = 101$ based on the APP LLR $L(x_0) = -1.2, 1.3, -0.6$, and the receiver becomes aware that the BL was successfully decoded using the CRC check. The resultant bits $\hat{x}_0 = 101$ will be output as the final estimates of the BL L_0 . Note that here we use the three-bits BL as an example, while in reality each layer may have thousands of bits.
6. At this stage, the BL $\hat{x}_0 = 101$ will be input to the VND 3 block of Figure 3.6, which will forward the hard bits $x_0 = 101$ to the VND block. Note that in practice, $\hat{x}_0 = 101$ will be expressed as $-\infty, \infty, -\infty$.

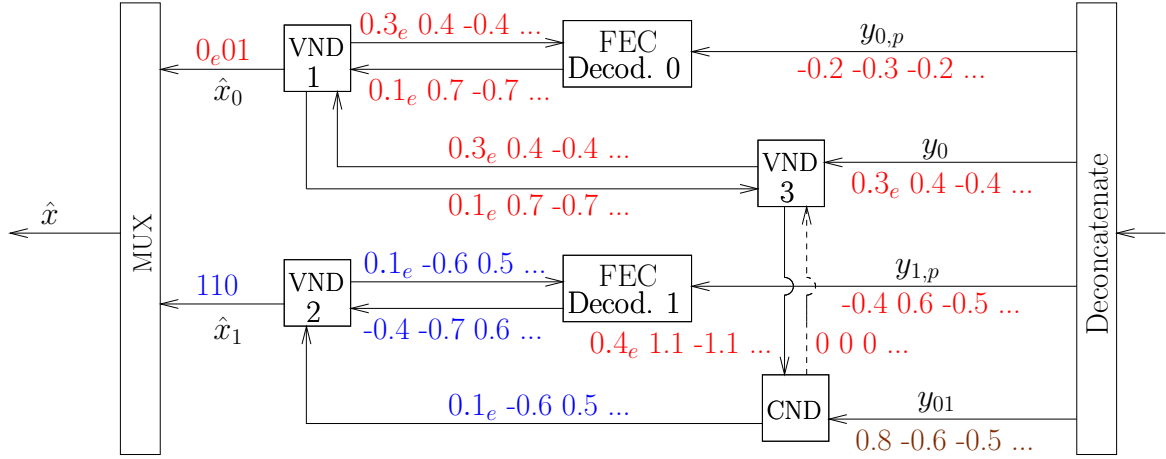


Figure 3.7: First round of IL-FEC decoding with IL feedback, where the subscript e indicates soft bit error.

7. Then, the CND block of Figure 3.6 may readily recover the soft-information of the EL x_1 as $y_1 = -0.3, -0.5, 0.2$ based on the inputs $x_0 = 101$ and $y_{01} = 0.3, -0.5, -0.2$. Note that $y_1 = -0.3, -0.5, 0.2$ is derived from $y_{01} = 0.3, -0.5, -0.2$ by a simple sign-flipping operations, hence no information is lost during this CND process.
8. The soft information $y_1 = -0.3, -0.5, 0.2$ generated by the CND is then forwarded to the FEC decoder 1 through VND 2 of Figure 3.6.
9. The soft information $L_e(x_1) = -0.4, -0.7, 0.3$ is then generated by the FEC decoder 1, which will be forwarded to VND 2 of Figure 3.6.
10. Based on the extrinsic LLR input $L_e(x_1) = -0.4, -0.7, 0.3$ gleaned from FEC decoder 1 and from the *a-priori* LLR input $y_1 = -0.3, -0.5, 0.2$ extracted from the CND, the VND 2 of Figure 3.6 can now generate the *a-posteriori* probability (APP) LLR $L(x_1) = -0.7, -1.2, 0.5$ of the EL $x_1 = 110$.
11. The EL can be estimated as $\hat{x}_1 = 110$ based on the APP LLR $L(x_1) = -0.7, -1.2, 0.5$, while the receiver is informed by the CRC check that the EL is successfully decoded.
12. Finally, the sequence $\hat{x}_0 = 101$ and $\hat{x}_1 = 110$ will be output as estimate of the BL and EL, respectively, which will be combined into $\hat{x} = 101110$ as the estimated bitstream of the H.264 decoder.

3.2.3.2 Operation With IL Feedback

The IL-FEC decoding relying on IL feedback is exemplified in this section. We consider the scenario that the soft information of $y_0 = 0.3_e, 0.4, -0.4$, $y_{0,p} = -0.2, -0.3, -0.2$, $y_{01} = 0.8, -0.6, -0.5$ and $y_{1,p} = -0.4, 0.6, -0.5$ is received at the receiver, where the

subscript e means soft bit error. Let us now detail the IL decoding process based on these soft information sequences.

At the beginning of the first IL-FEC decoding round, the CND of Figure 3.6 does not have any information concerning the BL L_0 . Hence the information fed back from the CND to VND 3 of Figure 3.7 will be initialized as $L_a(x_0) = 0, 0, 0$. Then, the first IL-FEC decoding round is performed as follows:

1. The *a-priori* information $L_a(x_0) = 0, 0, 0$ extracted from the CND of Figure 3.7 and $y_0 = 0.3_e, 0.4, -0.4$ will be input to VND 3 for generating the *a-priori* information $L_a(x_0) = 0.3_e, 0.4, -0.4$ for VND 1.
2. The *a-priori* information $L_a(x_0) = 0.3_e, 0.4, -0.4$ will be input to the FEC decoder 0 of Figure 3.7.
3. The soft information $L_e(x_0) = 0.1_e, 0.7, -0.7$ is then generated by the FEC decoder 0, which will be forwarded to VND 1 of Figure 3.7.
4. Based on the extrinsic LLR input $L_e(x_0) = 0.1_e, 0.7, -0.7$ extracted from FEC decoder 0 and the *a-priori* LLR input $L_a(x_0) = 0.3_e, 0.4, -0.4$ gleaned from VND 3, the VND 1 of Figure 3.7 becomes capable of generating the APP LLR $L(x_0) = 0.4_e, 1.1, -1.1$ of the BL $x_0 = 101$.
5. At this stage, the BL can be estimated as $\hat{x}_0 = 0_e01$ based on the APP LLR $L(x_0) = 0.4_e, 1.1, -1.1$, while the receiver is informed by the CRC check that the BL is corrupted. Hence the IL feedback will be activated for correcting the bit error of $\hat{x}_0 = 0_e01$.
6. Since the estimated $\hat{x}_0 = 0_e01$ is error-infested, the extrinsic LLR $L_e(x_0) = 0.1_e, 0.7, -0.7$ will be input to the VND 3 block.
7. Based on the soft input $L_e(x_0) = 0.1_e, 0.7, -0.7$ received from VND 1 and the soft information $y_0 = 0.3_e, 0.4, -0.4$, the extrinsic LLR $L_e(x_0) = 0.4_e, 1.1, -1.1$ will be generated by the VND 3 block as the input of the CND of Figure 3.7.
8. Then, the CND block is in the position generating soft information for the EL as $y_1 = 0.1_e, -0.6, 0.5$ based on the inputs $L_e(x_0) = 0.4_e, 1.1, -1.1$ and $y_{01} = 0.8, -0.6, -0.5$.
9. The soft information $y_1 = 0.1_e, -0.6, 0.5$ generated by the CND is then forwarded to the FEC decoder 1 through VND 2 of Figure 3.7.
10. The soft information $L_e(x_1) = -0.4, -0.7, 0.6$ is then generated by the FEC decoder 1, which will be forwarded to VND 2 of Figure 3.7.

Given the above-mentioned decoding process, the second round of IL-FEC decoding will be performed as detailed in Figure 3.8, which is listed as follows:

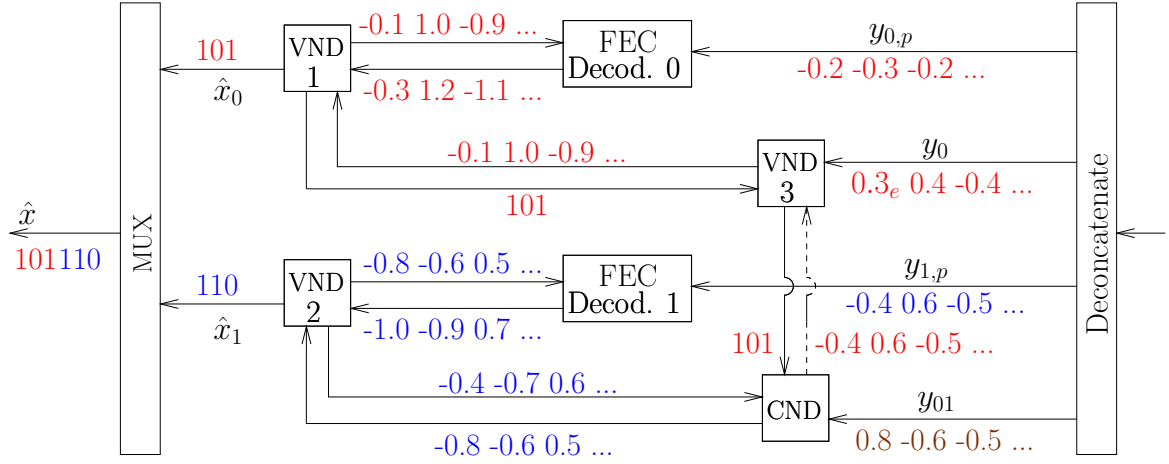


Figure 3.8: Second round of IL-FEC decoding with IL feedback, where the subscript e indicates soft bit error.

1. The soft information $L_e(x_1) = -0.4, -0.7, 0.6$ generated by the FEC decoder 1 in Figure 3.7 will be forwarded to the CND through VND 2.
2. Based on the soft information $L_e(x_1) = -0.4, -0.7, 0.6$ provided by VND 2 and the channel information $y_{01} = 0.8, -0.6, -0.5$, the *a-priori* information $L_a(x_0) = -0.4, 0.6, -0.5$ will be generated by CND of Figure 3.8, which will be input to VND 3.
3. The VND 3 of Figure 3.8 will then generate the *a-priori* information $L_a(x_0) = -0.1, 1.0, -0.9$ for VND 1 based on the soft information $L_a(x_0) = -0.4, 0.6, -0.5$ provided by CND and on the channel information of $y_0 = 0.3_e, 0.4, -0.4$. Note that the bit error in $y_0 = 0.3_e, 0.4, -0.4$ is corrected using the feedback information $L_a(x_0) = -0.4, 0.6, -0.5$ extracted from the EL L_1 .
4. The *a-priori* information $L_a(x_0) = -0.1, 1.0, -0.9$ will be input to the FEC decoder 0 of Figure 3.8.
5. The soft information $L_e(x_0) = -0.3, 1.2, -1.1$ is then generated by the FEC decoder 0, which will be forwarded to VND 1 of Figure 3.8.
6. Based on the extrinsic LLR input $L_e(x_0) = -0.3, 1.2, -1.1$ gleaned from FEC decoder 0 and from *a-priori* LLR input $L_a(x_0) = -0.1, 1.0, -0.9$ provided by VND 3, the VND 1 of Figure 3.6 can generate the APP LLR $L(x_0) = -0.4, 2.2, -2.0$ of BL $x_0 = 101$.
7. At this stage, the BL can be estimated as $\hat{x}_0 = 101$ based on the APP LLR $L(x_0) = -0.4, 2.2, -2.0$, while the receiver becomes aware that the BL was successfully decoded using the CRC check. Hence the sequence of $\hat{x}_0 = 101$ will be output as the final estimate of the BL L_0 .
8. The sequence of $\hat{x}_0 = 101$ will be input to the VND 3 block of Figure 3.8, which will forward the hard bits $x_0 = 101$ to the CND block.

9. Then, the CND block may readily recover the soft-information of $y_1 = -0.8, -0.6, 0.5$ based on the inputs $x_0 = 101$ and $y_{01} = 0.8, -0.6, -0.5$. Note that $y_1 = -0.8, -0.6, 0.5$ is derived from $y_{01} = 0.8, -0.6, -0.5$ by a simple sign-flipping operation, hence no information is lost during this CND process.
10. The soft information $y_1 = -0.8, -0.6, 0.5$ generated by the CND is then forwarded to the FEC decoder 1 through VND 2 of Figure 3.8.
11. The soft information $L_e(x_1) = -1.0, -0.9, 0.7$ is then generated by the FEC decoder 1, which will be forwarded to VND 2 of Figure 3.8.
12. Based on the extrinsic LLR input $L_e(x_1) = -1.0, -0.9, 0.7$ received from FEC decoder 1 and the *a-priori* LLR input $y_1 = -0.8, -0.6, 0.5$ gleaned from the CND, the VND 2 of Figure 3.8 can now generate the APP LLR $L(x_1) = -1.8, -1.5, 1.2$ of the EL $x_1 = 110$.
13. The EL can be estimated as $\hat{x}_1 = 110$ based on the APP LLR $L(x_1) = -1.8, -1.5, 1.2$, while the receiver can readily be informed by the CRC check that the EL is successfully decoded.
14. Finally, the sequence $\hat{x}_0 = 101$ and $\hat{x}_1 = 110$ will be output as estimate of the BL and EL, respectively, which will be combined into $\hat{x} = 101110$ as the estimated bitstream of the H.264 decoder.

In the above process, only two decoding rounds were considered. However, in practice more decoding rounds may be utilized based on a trade off between the complexity imposed and the performance attained.

3.2.4 Inter-Layer FEC Coding for Layers Having Unequal Length

In the above discussions, we assumed that the A, B and C partitions have an identical length. However, in practice they may carry an unequal number of bits. Here we detail the technique of applying our algorithm in the scenario, when the three partitions have an unequal length. Let us commence by assuming that the A, B, C partitions have the length of n_a, n_b, n_c bits, respectively.

For the case of implanting x_a into the systematic bits of x_b , the basic philosophy of the algorithm is to map/encode x_a into a new bit sequence t_b , which has the same number of bits as the bitstream x_b and will be implanted into the systematic bits of x_b using the algorithm discussed in Section 3.2.1. In other words, the bits x_a will be replaced by the newly generated bits t_b for the implantation process. Specifically, we introduce the sets $T_b^1, \dots, T_b^{n_b}$ to assist in generating the stream t_b , where the relationship between $T_b^1, \dots, T_b^{n_b}$ and the sequence x_b is displayed in Figure 3.9. For $n_a > n_b$, we split x_a into n_b number of groups on average as in Figure 3.9 (a), each constituting one of the sets $T_b^1, \dots, T_b^{n_b}$. By contrast, for $n_a < n_b$, we split

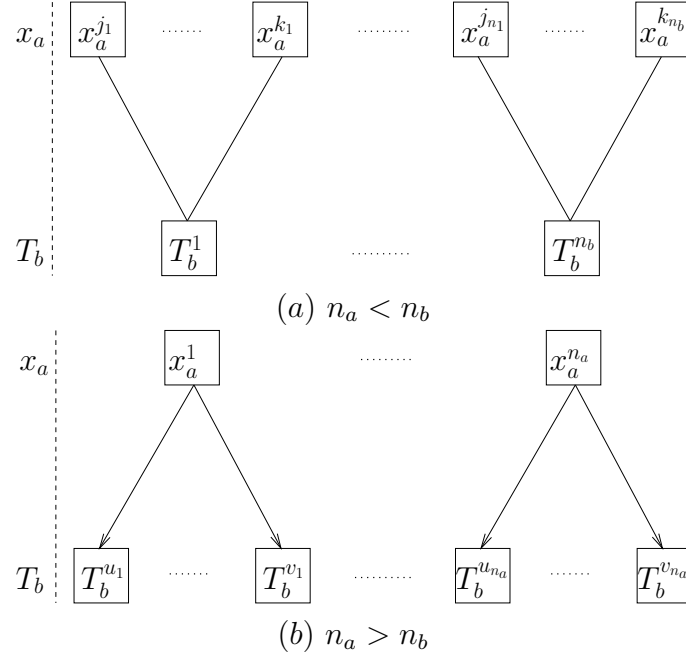


Figure 3.9: Definition of $T_b^1, \dots, T_b^{n_b}$ when the BL sequence x_a and the EL sequence x_b carry unequal length of bits.

$T_b^1, \dots, T_b^{n_b}$ into n_a number of groups on average as in Figure 3.9 (b), where the sets $T_b^1, \dots, T_b^{n_b}$ within the same group contain the same single bit of x_a . So far the sets $T_b^1, \dots, T_b^{n_b}$ have been created from the bit sequence x_a . Then, each bit of the sequences t_b will be generated from one of the sets $T_b^1, \dots, T_b^{n_b}$ as follows:

$$t_b^i = \sum_{x_a^r \in T_b^i} \oplus x_a^r, 0 < i \leq n_b. \quad (3.3)$$

Given the sequence t_b , we simply replace x_a by t_b , when implanting the x_a into the systematic bits of x_b . Therefore, x_{ab} may be generated correspondingly using $x_{ab}^i = t_b^i \oplus x_b^i$. Similarly, the stream x_a can be readily implanted into x_c by introducing the bit sequence t_c and the sets $T_c^1, \dots, T_c^{n_c}$.

At the receiver, based on the technique detailed in Section 3.2.2, decoder A is able to generate the extrinsic information of x_a . Decoder B is able to generate the extrinsic information of t_b with the assistance of CND 2 of Figure 3.1. Hence we design the technique to convert the extrinsic information between the sequence x_a and t_b for the sake of exchanging extrinsic information among the decoder A, CND 2 and decoder B of Figure 3.1. Provided the LLR of x_a and Eq. (3.3), the extrinsic

LLR of t_b may be readily derived using the boxplus operation as follows

$$L_e(t_b^i) = L \left(\sum_{x_a^r \in T_b^i} \oplus x_a^r \right) = \sum_{x_a^r \in T_b^i} \boxplus L(x_a^r). \quad (3.4)$$

Similarly, provided the *a-priori* LLR of x_a and the LLR of t_b^i , the extrinsic LLR of x_a may be derived as follows.

1. When $n_a > n_b$, the extrinsic information of x_a may be readily derived as

$$\begin{aligned} L_e(x_a^i) &= L \left(\sum_{x_a^r \in T_b^i \setminus x_a^i} \oplus x_a^r \oplus t_b^i \right) \\ &= \sum_{x_a^r \in T_b^i \setminus x_a^i} \boxplus L_e(x_a^r) \boxplus L(t_b^i). \end{aligned} \quad (3.5)$$

2. When $n_a < n_b$, the extrinsic information of x_a can be expressed as

$$L_e(x_a^i) = \sum_{\forall T_b^r, x_a^i \in T_b^r} L_e(x_a^r). \quad (3.6)$$

Note that the basic idea of the above algorithm is to map the bits x_a into a new bit sequence t_b , which is basically an encoder having a variable coding rate encoder. Hence, a number of codecs, such as low-density parity-check (LDPC) codes [53] and Luby transform (LT) [61] codes may be employed for the mapping of x_a to the stream t_b . However, they may impose error-propagation in this specific scenario. Hence, in this treatise we employ the method detailed in this section to prevent error-propagation.

3.2.5 IL-FEC Overheads

The possible overheads imposed by our proposed technique are listed as follows:

1. Delay: Our technique is implemented using the partitioning mode of H.264, where each video frame may be encoded into a number of slices. These slices may be encoded into at most three partitions. Since the IL encoding and decoding process is performed within each slice, no extra delay is imposed by our proposed technique. Specifically, we assume that each frame is encoded into a single slice consisting of three layers, each of which carries 1000 bits. In the traditional UEP systems, BL A is encoded and transmitted, followed by the encoding and transmission of ELs B, C. Hence a 1000-bit delay is imposed at

the transmitter, but no frame-level delay is imposed. At the receiver, BL A may be firstly decoded without waiting for the arrival of ELs B and C, where B and C are independent of each other. Hence a bit-level delay of 1000 bits is imposed in these systems. In video communications, each video frame can only be reconstructed with the aid of the decoded layers A, B and C. Hence no frame-delay is imposed by these systems. On the other hand, in the IL-FEC coded systems BL A is encoded and transmitted independently of ELs B and C. Then IL-FEC encoding is performed on the layer pairs A, B and A, C, respectively. Note that the processing of ELs B and C is performed independently. Hence a 1000-bit delay is imposed on the IL-FEC coded systems, while no frame-level delay is imposed. At the receiver, BL A may be firstly decoded without waiting for the arrival of ELs B and C. In the scenario that the BL A is successfully decoded, the estimated BL A can be directly forwarded to the H.264 decoder, which results in a delay of 1000 bits. Then the ELs B and C may be independently decoded based on the recovered BL A. In the case that the BL A is not successfully decoded, the IL-FEC decoder has to wait for ELs B and C for correcting the errors, which results in a delay of 3000 bits. Hence our IL-FEC may impose a maximum bit-level delay of 3000 bits. However, since each frame can only be reconstructed with the aid of the decoded layers A, B and C, no frame level-delay is imposed by these systems. We may conclude that no frame level delay is imposed by the IL-FEC techniques.

2. Complexity: As detailed in Section 3.2.2, the signal-flows are based on low-complexity operations compared to the FEC decoding. When the BL A can be recovered in its own right, only sign-flipping is necessitated for extracting the systematic LLR information of the ELs B and C . Specifically, we impose modest extra complexity⁵, as it will be detailed in Section 3.5.4.
3. FEC-redundancy: The BL A does not rely on the ELs for its decoding operations and the systematic LLR information of the ELs B and C can be extracted from the received check information y_{ab} and y_{ac} without any loss, provided that the BL is perfectly decoded. Furthermore, since the transmitted bit sequences x_{ab} and x_{ac} have the same length as that of the bit sequence x_b and x_c , respectively, we do not impose any extra protection bits. Hence the IL-FEC does not impose extra FEC redundancy.

⁵According to our experiments, it is sufficient to use a single iteration, which results in a low complexity.

3.3 EXIT Analysis of IL-FEC Coded Systems

EXIT charts [133, 156] are widely used in the state-of-art for analyzing the performance of FEC codes or systems. In this section, we utilize EXIT charts for illustrating why our proposed IL-FEC coded systems are capable of improving the performance of the system. Moreover, an RSC and a turbo code [157] will be employed in this section for providing insights into the benefits of IL-FEC techniques.

3.3.1 Mutual Information Analysis Using RSC

System Parameters	Value	System Parameters	Value
FEC	RSC[1011, 1101, 1111]	Number of Tx antennas	4
Modulation	QPSK	Elements Per AA	4
Channel	Narrowband Rayleigh Fading Channel	Number of Rx antennas	4
		Overall Coding Rate	1/2

Table 3.5: Parameters employed for characterization of the system in Figure 3.1, where “AA” indicates antenna array.

Error Protection Arrangements	Code Rates		
	L_0	L_1	Average
EEP	0.5	0.5	0.5
UEP1	0.54	0.46	0.5
UEP2	0.47	0.53	0.5

Table 3.6: Coding rates of the RSC coding error protection arrangements in the system of Figure 3.1 for the BL L_0 and the EL L_1 . The code-rates were adjusted by variable-rate puncturers.

In this section, we analyze our proposed system using MI⁶. For the sake of simplifying the analysis, we assume that there are two layers: a BL L_0 and an EL L_1 . Furthermore, we employed a 1/3 RSC having the generator polynomials [1011, 1101, 1111]⁷. The system parameters used in our simulations are summarized

⁶MI is known as a metric to represent the confidence of a signal sequence. Generally bigger MI indicates lower BER value of the measured signal sequence, while lower BER normally indicates lower PLR.

⁷The first polynomial indicates the feedback parameter, while the other two polynomials represent the feed-forward parameters. The code rates were adjusted by variable-rate puncturers.

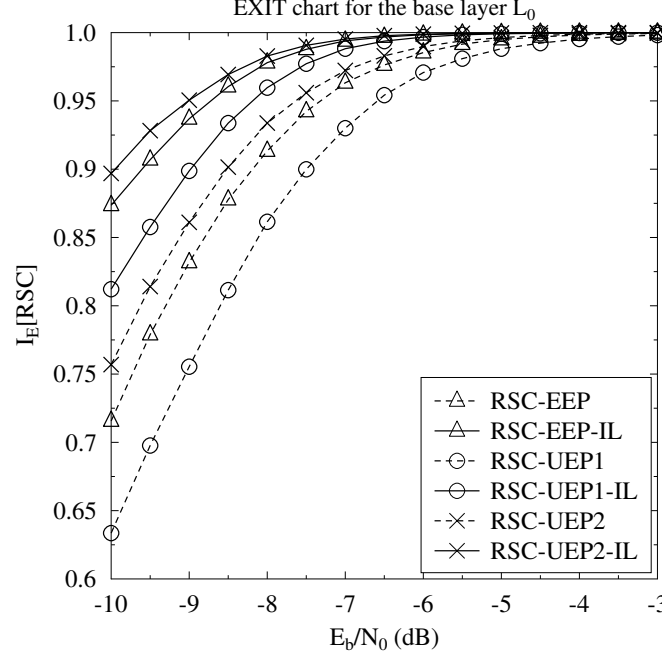


Figure 3.10: Extrinsic information generated by the RSC decoders for all error protection arrangements of Table 3.6. The schematic of Figure 3.1 was used.

in Table 3.5. In the following analysis, where two layers are considered, the BL is protected by the IL-FEC codec. Hence, we consider the convergence behavior of the BL. For the sake of analyzing our IL-FEC codec, different error protection arrangements were considered, as shown in Table 3.6.

In Figure 3.10, we plot the extrinsic MI at the output of the RSC decoder for different E_b/N_0 values for all the codes in Table 3.6. Observe from Figure 3.10 that the schemes employing our iterative inter-layer technique always acquire a higher MI value than those dispensing with the IL-FEC technique. For example, the RSC-EEP scheme and RSC-EEP-IL scheme generate 0.91 and 0.975⁸ extrinsic information at -8 dB. This improvement is attained by our proposed scheme due to the fact that extra MI is fed back to the BL from the EL. Similar trends can be observed for the system pairs RSC-UEP1 and RSC-UEP1-IL, RSC-UEP2 and RSC-UEP2-IL, as displayed in Figure 3.10.

3.3.2 EXIT Analysis Using Turbo

In this section, we analyze our proposed system using EXIT charts [159]. For the sake of simplifying the analysis, we assume that there are two layers: a BL L_0 and an EL L_1 . Furthermore, we employed a 1/3 rate turbo codec consisting of two identical RSC codec with the generator of [111, 101]. The system parameters used in our simulations are summarized in Table 3.7. In the following analysis, where two layers

⁸Larger amount of extrinsic information indicates a lower BER [158].

Parameters	Value	Parameters	Value
FEC	turbo-[111, 101]	Number of Tx antennas	4
Modulation	QPSK	Elements Per AA	4
Channel	Narrowband Rayleigh Fading Channel	Number of Rx antennas	4
		Overall Coding Rate	1/2

Table 3.7: Parameters employed in our systems of Figure 3.1, where “AA” indicates antenna array.

Error Protection Arrangements	Code Rates		
	L_0	L_1	Average
EEP	0.5	0.5	0.5
UEP1	0.54	0.46	0.5
UEP2	0.47	0.53	0.5

Table 3.8: Coding rates of turbo coding error protection arrangements of Figure 3.1 for the BL L_0 and the EL L_1 . The code-rates were adjusted by variable-rate puncturers.

are considered, the BL L_0 is protected by the IL-turbo codec. Hence, we consider the convergence behavior of the BL. For the sake of analyzing our IL-turbo codec, different error protection arrangements were considered, as shown in Table 3.8, where EEP stands for equal error protection and UEP is unequal error protection.

In Figure 3.11, we plot the EXIT chart for the turbo coded systems of Figure 3.1 using the coding arrangements of Table 3.8. The EEP, UEP1, UEP2 related schemes of Table 3.8 are characterized in Figures 3.12, 3.13 and 3.14, respectively, where the Lower-UEP1-IL displayed in Figure 3.13 for example characterizes the lower RSC decoder of the IL-turbo-LSSTC system utilizing the UEP1 arrangement of Table 3.8. Observe from Table 3.8, the error correction capability of the BL L_0 increases in the order of UEP1, EEP and UEP2, while the width of the open EXIT tunnel increases in the same order, as observed from Figure 3.11. Observe in Figure 3.11 that at $E_b/N_0 = -8.5$ dB, the IL-turbo coded system has a wider open EXIT tunnel than the system dispensing with the IL-turbo. More explicitly, if we consider the UEP2 aided system, then it is clear from Figure 3.14 that the IL-turbo coded system has a wider open EXIT tunnel. In other words, the IL-turbo coded system requires a lower E_b/N_0 than its counterpart dispensing with IL-turbo in order to attain an open

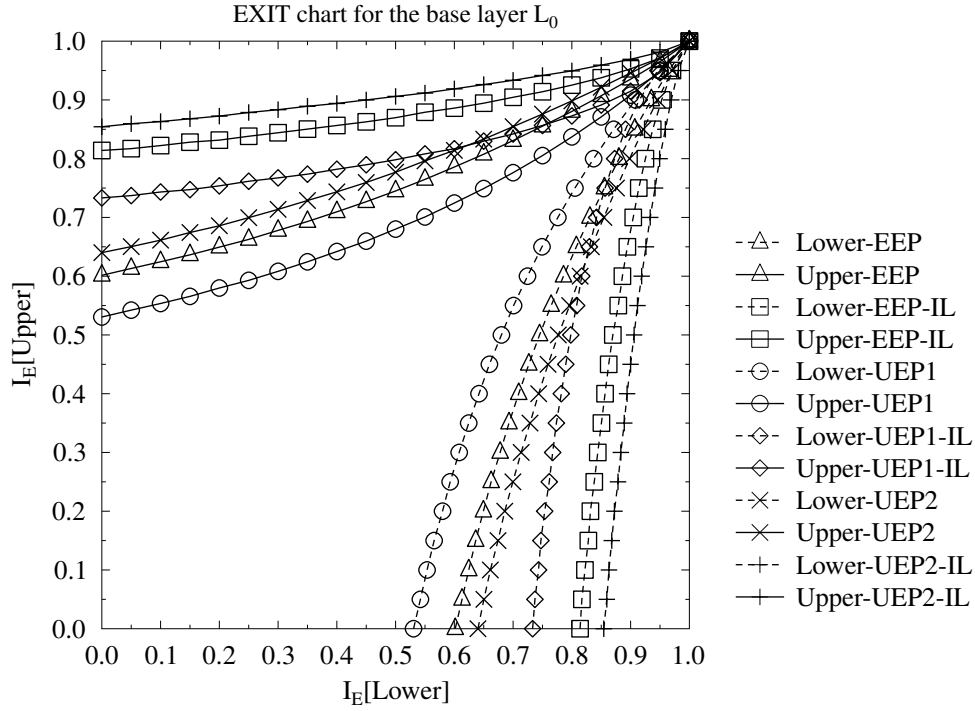


Figure 3.11: Comparison of the upper RSC and lower RSC EXIT curves at $E_b/N_0 = -8.5$ dB for all error protection arrangements of Table 3.8. The schematic of Figure 3.1 was used.

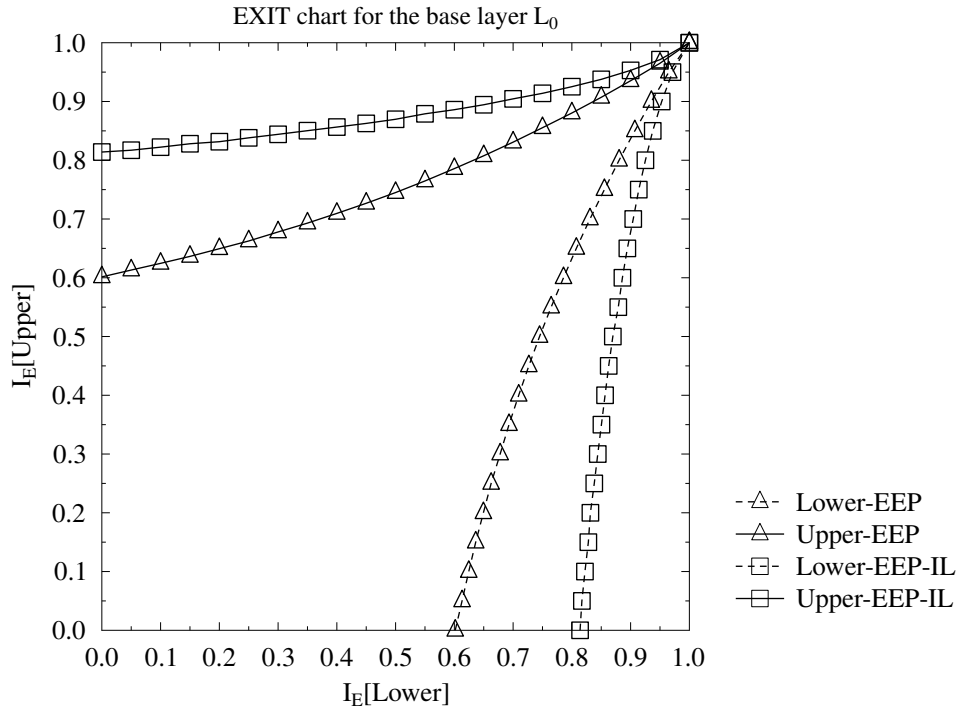


Figure 3.12: Comparison of the upper RSC and lower RSC EXIT curves at $E_b/N_0 = -8.5$ dB for EEP error protection arrangements of Table 3.8. The schematic of Figure 3.1 was used.

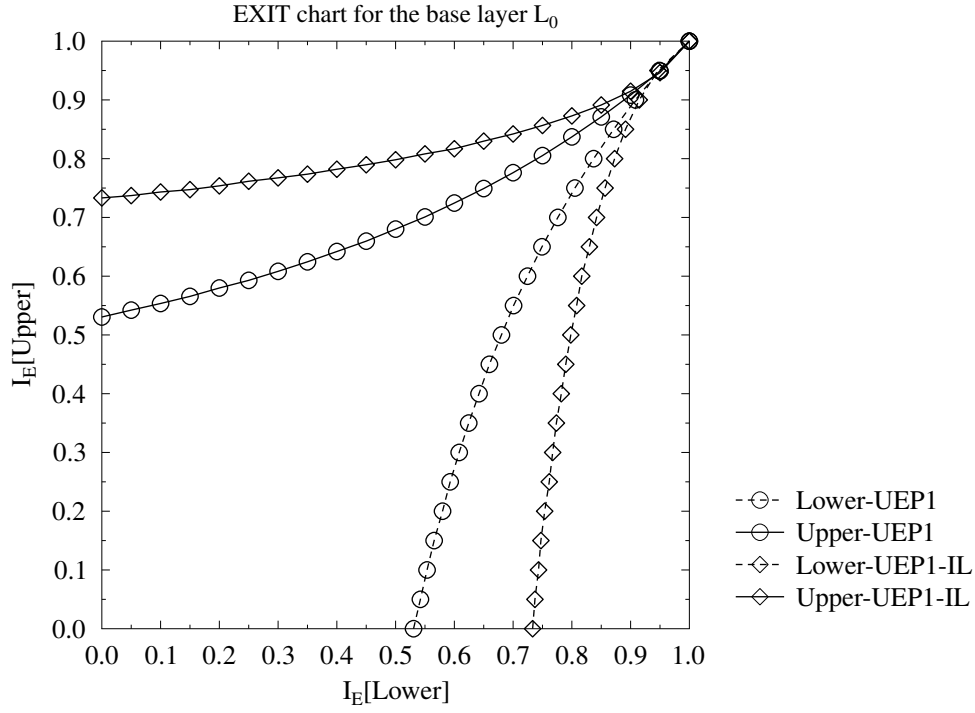


Figure 3.13: Comparison of the upper RSC and lower RSC EXIT curves at $E_b/N_0 = -8.5$ dB for UEP1 error protection arrangements of Table 3.8. The schematic of Figure 3.1 was used.

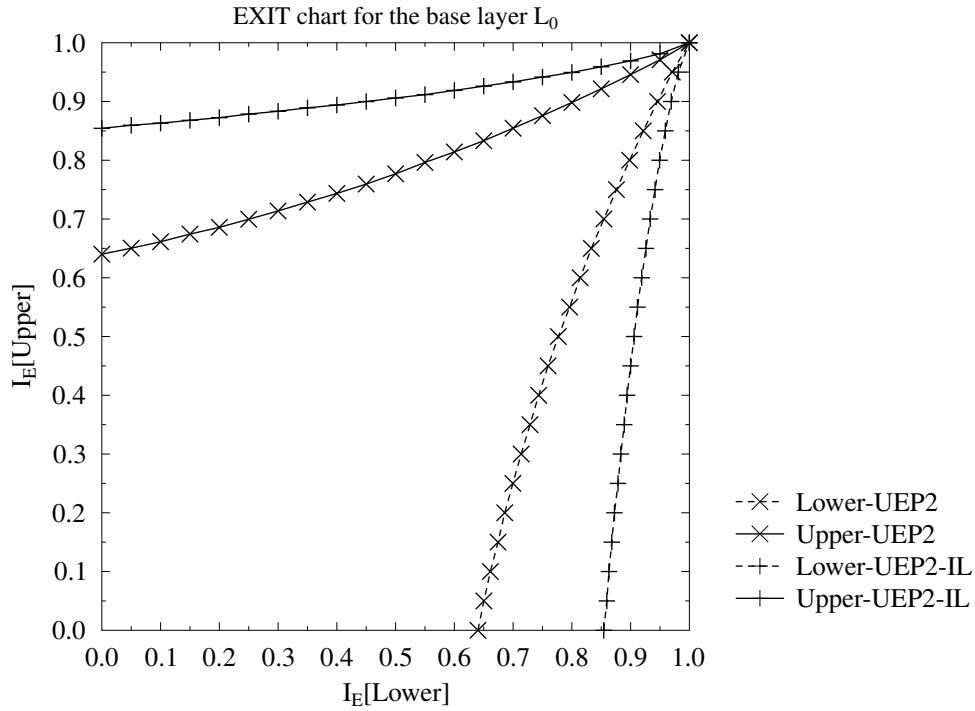


Figure 3.14: Comparison of the upper RSC and lower RSC EXIT curves at $E_b/N_0 = -8.5$ dB for UEP2 error protection arrangements of Table 3.8. The schematic of Figure 3.1 was used.

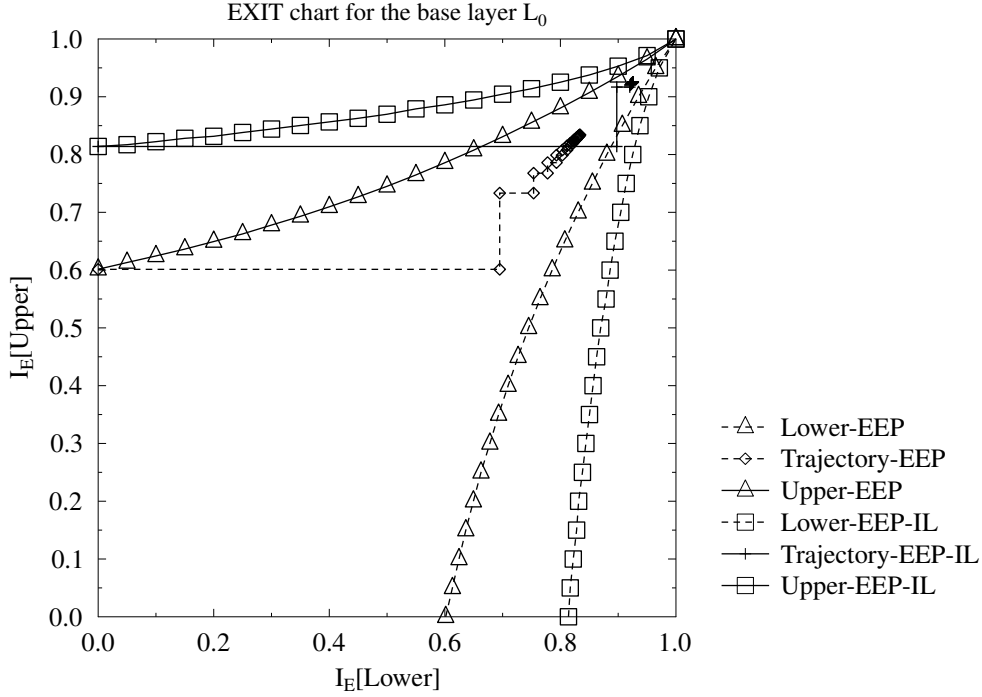


Figure 3.15: Trajectory comparison of the upper RSC and lower RSC EXIT curves at $E_b/N_0 = -8.5$ dB for the systems employing EEP. The schematic of Figure 3.1 was used.

tunnel. This implies that the IL-turbo system is capable of attaining a better BER performance for the BL than its counterpart dispensing with IL-turbo coding. The reason for attaining a wider EXIT tunnel by our proposed scheme is due to the fact that extra MI is fed back to the BL from the EL.

An EXIT trajectory comparison of the EEP-turbo-LSSTC system and of the EEP-IL-turbo-LSSTC system is displayed in Figure 3.15, which is based on Monte-Carlo simulations. Observe from Figure 3.15 that the EEP-IL-turbo-LSSTC system has a wider open tunnel than the EEP-turbo-LSSTC system, as discussed in the previous paragraph. The stair-case-shaped decoding trajectory of the EEP-IL-turbo-LSSTC system reaches the point (0.93, 0.93), while that of the EEP-turbo-LSSTC system is curtailed around (0.83, 0.83) point. Hence our proposed system has a better convergence behavior than the EEP technique, which results in a better BER performance [138]. Observe in Figure 3.15 that, although there is an open EXIT tunnel between the curves “Lower-EEP” and “Upper-EEP”, the trajectory fails to converge to the (1, 1) point of perfect convergence to a vanishingly low BER due to the fact that we employ short interleavers. The length of the interleaver is constrained in real-time video streaming application for the sake of delay control. Therefore, it can be inferred from Figures 3.12 and 3.15 that employing the IL-turbo coding results in a better BER performance, which is demonstrated by the wider open EXIT tunnel

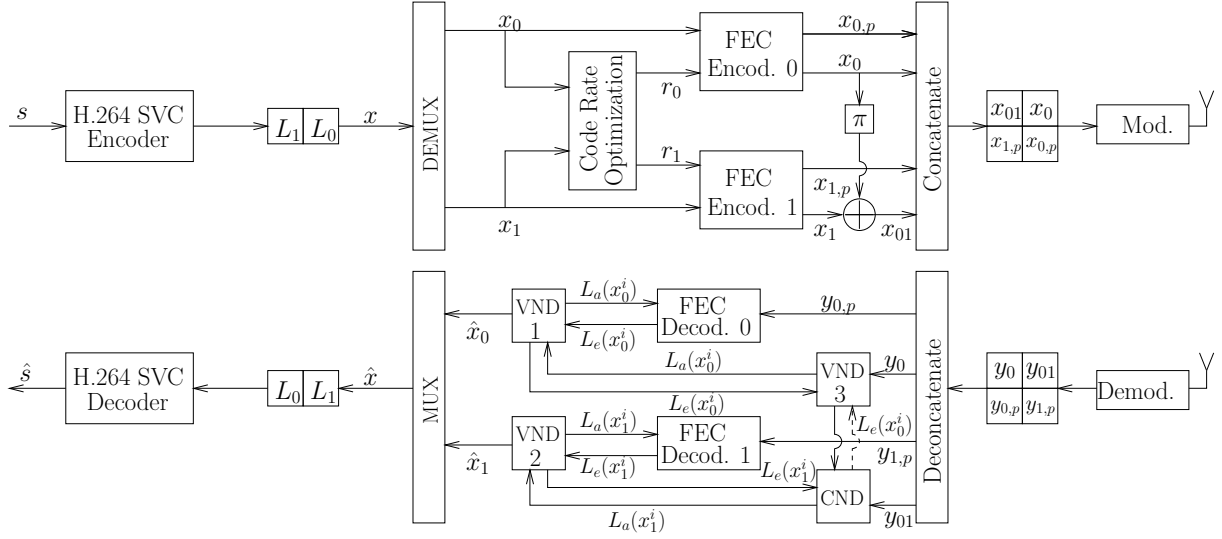


Figure 3.16: Block diagram of the proposed IL-FEC coded SVC H.264/AVC coded video system. In contrast to Figure 3.1, this schematic employs the “Code Rate Optimization” block for generating the optimized code rates for FEC encoder 0 and 1.

shown in Figures 3.12 and 3.15.

3.4 Distortion Minimization by Adjusting the FEC Coding Rate

In Section 3.2, we detailed our IL-FEC coded system conceived for partitioned video transmission. In Sections 3.5 and 3.6, the IL-FEC coded techniques of Section 3.2 will be investigated in conjunction with channel coding, where the code rates are determined empirically. However, in practical scenarios, we may require different channel coding rates for achieving the best system performance. Against this background, here we set out to conceive a technique for finding the optimized coding rates for channel-coded bitstreams “on-the-fly” at the transmitter for the sake of optimizing the IL-FEC coded system’s performance.

The general system structure of Figure 3.1 was further developed in Figure 3.16, where the BL L_0 and the EL L_1 are generated by the H.264 SVC codec [14]. In contrast to Figure 3.1, the schematic of Figure 3.16 employs two layers L_0 , L_1 and the blocks “Mod.” and “Demod.” representing a generalized transceiver, where the “Code Rate Optimization” block is capable of finding the best code rates. The technique detailed in Section 3.2 may also be readily applied for the system displayed in Figure 3.16.

Below, we further improve the scalable video transmission system of Figure 3.16, where IL-FEC coded UEP is employed for the sake of achieving an improved video

quality. Specifically, we seek specific FEC coding rates for the layers L_0 and L_1 for achieving the best system performance. Additionally, although only two layers are assumed in Figure 3.16, more complex coding scenarios may be readily applied, as evidenced by the results of Section 2.5. Below, we will use an RSC codec for describing the proposed solution. However, our proposed techniques are not limited to the RSC codec. Firstly, we define the following notations that will aid our analysis:

- $d(L_0)$: video distortion, namely the PSNR reduction, induced by the corruption of the BL L_0 , which is measured using the peak signal-to-noise ratio (PSNR);
- $d(L_1)$: the video distortion, when the BL L_0 is correct while the EL L_1 is corrupted;
- $|L_i|$: the length of the bitstream of layer L_i , $0 \leq i \leq 1$;
- R : the overall coding rate of the system shown in Figure 3.16;
- r_i : the coding rate of layer L_i , $0 \leq i \leq 1$;
- $p(L_0)$: the packet error ratio (PER) of layer L_0 ;
- $p(L_1)$: the PER of layer L_1 , when the layer L_0 is correctly decoded.

According to the IL-FEC decoding process detailed in Section 3.2, both the FEC decoders 0 and 1 of Figure 3.16 affect the PER $p(L_0)$, where FEC decoder 1 feeds back *a-priori* information of BL L_0 through the VND and CND blocks of Figure 3.16. Specifically, the performance of the FEC decoder 0 depends on SNR , r_0 , $|L_0|$, while the performance of the FEC decoder 1 depends on SNR , r_1 , $|L_1|$. Hence $p(L_0)$ depends on the parameters SNR , $|L_0|$, $|L_1|$, r_0 and r_1 , which can be expressed as

$$p(L_0) = f_0(SNR, |L_0|, |L_1|, r_0, r_1). \quad (3.7)$$

On the other hand, $p(L_1)$ purely depends on the FEC decoder 1, hence it is determined by the parameters SNR , $|L_1|$ and r_1 , which may be expressed as

$$p(L_1) = f_1(SNR, |L_1|, r_1). \quad (3.8)$$

Given the specific layers L_0 and L_1 , $d(L_i)$ may be calculated experimentally as the PSNR degrades due to the erasure of layer L_i [56]. Then the expected PSNR reduction induced by BL L_0 may be estimated as $p(L_0) \cdot d(L_0)$. Additionally, when BL L_0 is correctly decoded, the expected PSNR reduction induced by EL L_1 may be estimated as $p(L_1) \cdot d(L_1)$. Hence the expected distortion jointly induced by the

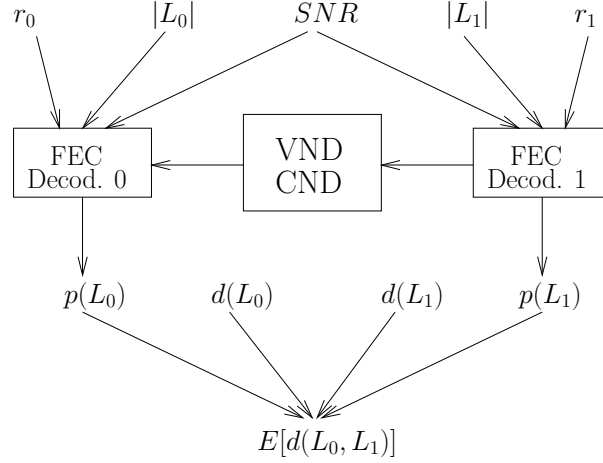


Figure 3.17: Dependency chains for calculating the expected distortion formulated in Eq. (3.9).

decoded layers L_0 and L_1 , denoted as $E[d(L_0, L_1)]$, may be estimated as

$$E[d(L_0, L_1)] = p(L_0) \cdot d(L_0) + [1 - p(L_0)] \cdot p(L_1) \cdot d(L_1). \quad (3.9)$$

Based on the above discussions, the expected distortion $E[d(L_0, L_1)]$ may be calculated by following the dependency chains displayed in Figure 3.17, which relies on the parameters SNR , $|L_0|$, $d(L_0)$, r_0 , $|L_1|$, $d(L_1)$ and r_1 . Among these parameters, SNR , $|L_0|$, $|L_1|$ are determined by the specific layers L_0 , L_1 and the specific transmit power, while $d(L_0)$ and $d(L_1)$ are gleaned from experiments [56]. Hence the coding rates r_0 , r_1 determine the distortion $E[d(L_0, L_1)]$ expected for the layers L_0 , L_1 . Therefore, the objective of the “Code Rate Optimization” block seen in Figure 3.16 is to find the specific rates r_0 and r_1 capable of minimizing the expected distortion $E[d(L_0, L_1)]$ of Eq. (3.9), which may be expressed as

$$\arg \min_{r_0, r_1} \{E[d(L_0, L_1)]\}, \quad (3.10)$$

subject to the condition of

$$\frac{|L_0|}{r_0} + \frac{|L_1|}{r_1} = \frac{|L_0| + |L_1|}{R}. \quad (3.11)$$

As illustrated in Figure 3.17, Eq. (3.9) relies on the PER definitions of Eqs. (3.7) and (3.8), which cannot be theoretically solved. Below we propose our solution of Eqs. (3.7) and (3.8) with the assistance of LUTs and the MI, where LUTs are employed for characterizing the system components of Figure 3.1 and MI is employed

for numerically quantifying the reliability of the LLRs. Specifically, Section 3.4.1 introduces the preliminaries of solving Eqs. (3.7) and (3.8), including the LUTs, the MI flow of the VND and CND, followed by the PER estimation of the BL and EL in Section 3.4.2. Finally, we exploit both the estimated PERs $p(\cdot)$ and the video distortions $d(\cdot)$ for determining the optimized coding rates in Section 3.4.3.

3.4.1 Preliminaries

The receiver of Figure 3.16 consists of the following components: demodulator, FEC decoder, VND, CND. The characteristics of these components jointly determine the PER $p(L_0)$ and $p(L_1)$, as shown in Figure 3.17. However, neither the demodulator and nor the FEC decoder may be readily characterized theoretically for diverse system configurations, such as different transceivers, FEC generator polynomials, decoding metrics etc. In order to propose a more universal solution, below LUTs are employed for the sake of characterizing both the demodulator and the FEC decoder, while the MI flow of the VND and CND will be detailed afterwards.

3.4.1.1 Lookup Tables

In our analysis we consider a RSC-coded BPSK-modulated system for transmissions over a non-dispersive uncorrelated Rayleigh fading channel. Specifically, we model BPSK based transmissions over a Rayleigh fading channel as a function of the channel SNR and generate the demodulator's output LLRs, where LLRs are quantified by the MI value, as in [133]. Furthermore, since the MI invoked for quantifying the reliability of the soft information can be applied for Gaussian distributed LLRs [133] generated by arbitrary transceivers, our procedure may be deemed generically applicable, provided that the transceivers generate near-Gaussian distributed LLRs⁹. By contrast, the RSC codec is modeled as a function of the SNR, of the LLRs of the systematic bits and of the coding rate, while generating the RSC decoder's output extrinsic LLRs and estimating the PER of the RSC decoded packets, where the LLRs are quantified by the MI value. The following LUTs are created for modeling the transceiver and the RSC codec:

- $T_b(SNR)$: The MI value of the LLRs output by the BPSK demodulator for variable channel SNRs. Since this table relies on the SNR only, it may be stored in a one-dimensional space, where the LUTs' size requirements will be detailed in Section 3.4.4.

⁹Our experiments, not included here for reasons of space-economy, suggest that higher order Quadrature Amplitude Modulation (QAM) can be readily applied for transmission over Rayleigh fading wireless channels, since it generates near-Gaussian distributed LLRs.

- $T_e(SNR, I_s, r)$: The MI value of the extrinsic LLR output of the RSC decoder recorded for variable channel SNRs, where I_s represents the MI value between the *a-priori* LLRs of the systematic information and the corresponding information bits, while r represents the coding rate of the RSC codec. Since this table relies on three parameters, it may be visualized in a three-dimensional space, where the LUTs' size requirements will also be detailed in Section 3.4.4.
- $T_p(SNR, I_s, r)$: The PER value associated with the LLRs output by the RSC decoder at diverse channel SNRs, where I_s represents the MI between the *a-priori* LLRs of the systematic information and the corresponding information bits, while r represents the coding rate of the RSC codec. Note that this LUT may be generated simultaneously with the LUT $T_e(SNR, I_s, r)$, since they have the same input variables.

3.4.1.2 MI Flow of the VND

As seen in Figure 3.3a, the operation of the VND may be characterized as $L_{o_3}(u_1) = L_{i_1}(u_1) + L_{i_2}(u_1)$. Assuming that the inputs $L_{i_1}(u_1)$ and $L_{i_2}(u_1)$ of the VND have the MI values of $I_{i_1}[u_1; L_{i_1}(u_1)]$ and $I_{i_2}[u_1; L_{i_2}(u_1)]$ respectively, the MI value of the output $L_{o_3}(u_1)$ may be expressed as [156]

$$I_{o_3}[u_1; L_{o_3}(u_1)] = J \left(\sqrt{J^{-1}(I_{i_1}[u_1; L_{i_1}(u_1)])^2 + J^{-1}(I_{i_2}[u_1; L_{i_2}(u_1)])^2} \right), \quad (3.12)$$

where $J(\cdot)$ and $J^{-1}(\cdot)$ are defined in [134].

3.4.1.3 MI Flow of the CND

Below, we firstly discuss the MI flow of the CND of Figure 3.3b, which is designed for the layers L_0 and L_1 , both carrying the same number of bits. Then we will discuss the practical CND employed for realistic unequal length layers.

As seen in Figure 3.3b, the action of the CND can be expressed as $L(u_3 = u_1 \oplus u_2) = L(u_1) \boxplus L(u_2)$ [154], where u_1 , u_2 and $u_3 = u_1 \oplus u_2$ are random binary variables. Assuming that the inputs $L_i(u_1)$, $L_i(u_2)$ of the CND have the MI values of $I_i[u_1; L(u_1)]$ and $I_i[u_2; L(u_2)]$ respectively, the MI value $I_o[u_3; L(u_3)]$ of u_3 has not been theoretically derived in the literature and since this is not the focus of our work, we model $I_o[u_3; L(u_3)]$ as

$$I_o[u_3; L(u_3)] = T_I(I_i[u_1; L(u_1)], I_i[u_2; L(u_2)]), \quad (3.13)$$

which is also characterized using LUT.

Given the MI $I_o[u_3; L(u_3)]$ in Eq. (3.13), we now discuss the practical scenario of the CND, where L_0 and L_1 may contain an unequal number of bits. As detailed in [75], the encoding architecture of L_0 and L_1 may be categorized into two scenarios, namely $|L_0| > |L_1|$ and $|L_1| > |L_0|$. Below we consider the scenario of generating $L(x_1^i)$, when $|L_0| > |L_1|$, which can be generalized to other scenarios. As detailed in [75], the bit x_{01}^i may be generated as

$$x_{01}^i = x_1^i \oplus \sum_{j=1}^n \oplus x_0^j, \quad (3.14)$$

where \oplus denotes the binary XOR operation. Additionally, since on average¹⁰ $|L_0|$ bits of L_0 were allocated to the $|L_1|$ bits of L_1 for generating the bits x_{01} , we have $n \in \left\{ \left\lceil \frac{|L_0|}{|L_1|} \right\rceil, \left\lceil \frac{|L_0|}{|L_1|} \right\rceil - 1 \right\}$ [75] for Eq. (3.14). More specifically, $\left(N - \frac{|L_0|}{|L_1|}\right)$ proportion of the bits in the EL L_1 are encoded based on Eq. (3.14) using $n = N - 1$, while the remaining bits of the EL L_1 are encoded based on Eq. (3.14) relying on $n = N$, where we have $N = \left\lceil \frac{|L_0|}{|L_1|} \right\rceil$.

Based on Eq. (3.14), the corresponding LLRs $L(x_1)$ of the EL L_1 may be expressed as

$$L(x_1^i) = L(x_{01}^i) \boxplus \sum_{j=1}^n \boxplus L(x_0^j). \quad (3.15)$$

Then, the LLRs of $L(x_1)$ are constituted by a mixture of LLRs $L^N(x_1)$ and $L^{N-1}(x_1)$, which are calculated based on Eq. (3.15) in conjunction with $n = N$ and $n = N - 1$, respectively. We denote the MI values corresponding to the LLRs $L^N(x_1)$ and $L^{N-1}(x_1)$ as $I^N[x_1; L(x_1)]$ and $I^{N-1}[x_1; L(x_1)]$. Based on Eq. (3.13), the MI values $I^N[x_1; L(x_1)]$ and $I^{N-1}[x_1; L(x_1)]$ may be recursively calculated as

$$I^k[x_1; L(x_1)] = \begin{cases} T_I(I[x_0; L(x_0)], I[x_{01}; L(x_{01})]), & k = 1, \\ T_I(I^{k-1}[x_1; L(x_1)], T_I(I[x_0; L(x_0)], I[x_{01}; L(x_{01})])), & k > 1. \end{cases} \quad (3.16)$$

where $1 \leq k \leq N$. Empirically, we assume that the LLRs $L^N(x_1)$ and $L^{N-1}(x_1)$ are near-Gaussian distributed, while their variance may be denoted as $J^{-1}(I^N[x_1; L(x_1)])$ and $J^{-1}(I^{N-1}[x_1; L(x_1)])$, respectively [134]. Again we assume that $L(x_1)$ exhibits

¹⁰On average, each bit of L_1 is encoded by $\frac{|L_0|}{|L_1|}$ bits of L_0 for generating a single bit of x_{01} , as detailed in [75].

near-Gaussian distribution, hence the variance of $L(x_1)$ may be formulated as

$$\begin{aligned}
 \sigma^2 &= D(L(x_1)) \\
 &= \left(1 - N + \frac{|L_0|}{|L_1|}\right) \cdot D(L^N(x_1)) + \left(N - \frac{|L_0|}{|L_1|}\right) \cdot D(L^{N-1}(x_1)) \\
 &= \left(1 - N + \frac{|L_0|}{|L_1|}\right) \cdot J^{-1}(I^N[x_1; L(x_1)])^2 + \left(N - \frac{|L_0|}{|L_1|}\right) \cdot J^{-1}(I^{N-1}[x_1; L(x_1)])^2.
 \end{aligned} \tag{3.17}$$

Finally, the MI value of $L(x_1)$ may be expressed using the J function as follows:

$$\begin{aligned}
 I[x_1; L(x_1)] &= J(\sigma) \\
 &= J\left(\sqrt{\left(1 - N + \frac{|L_0|}{|L_1|}\right) \cdot J^{-1}(I^N[x_1; L(x_1)])^2 + \left(N - \frac{|L_0|}{|L_1|}\right) \cdot J^{-1}(I^{N-1}[x_1; L(x_1)])^2}\right).
 \end{aligned} \tag{3.18}$$

3.4.2 PER Estimation

The PER of the decoded BL and EL having coding rates of r_0 and r_1 will be estimated in Section 3.4.2.1 and Section 3.4.2.2, respectively.

3.4.2.1 IL-FEC Coded BL

As shown in Figure 3.17, the PER of the BL L_0 depends on the characteristics of the IL-FEC decoder, including the generator polynomials, the trellis decoding techniques employed, as well as the VND and the CND of Figure 3.3. Below we propose our solution conceived for estimating the PER of Eq. (3.8) with the aid of the LUTs defined in Section 3.4.1.1.

The PER estimation flow-chart of the IL-FEC coded BL is illustrated in Figure 3.18, which follows the decoding process of Figure 3.16. Given the SNR , the MI value I_s of y_0 can be generated by the LUT $T_b(SNR)$. Furthermore, the MI value of the extrinsic LLR gleaned from the RSC decoder 0 may be expressed as $T_e[SNR, T_b(SNR), r_0]$. Then the "VND1-VND3-CND-VND2" process of Figure 3.18 is capable of calculating the MI value $I_a(x_1)$ of $L_a(x_1^i)$. Afterwards, based on the LUT $T_e(SNR, I_s, r)$ of the RSC decoder 1, the MI value $I_e(x_1)$ of the extrinsic information may be readily expressed as

$$I_e(x_1) = T_e[SNR, T_b(SNR), r_1].$$

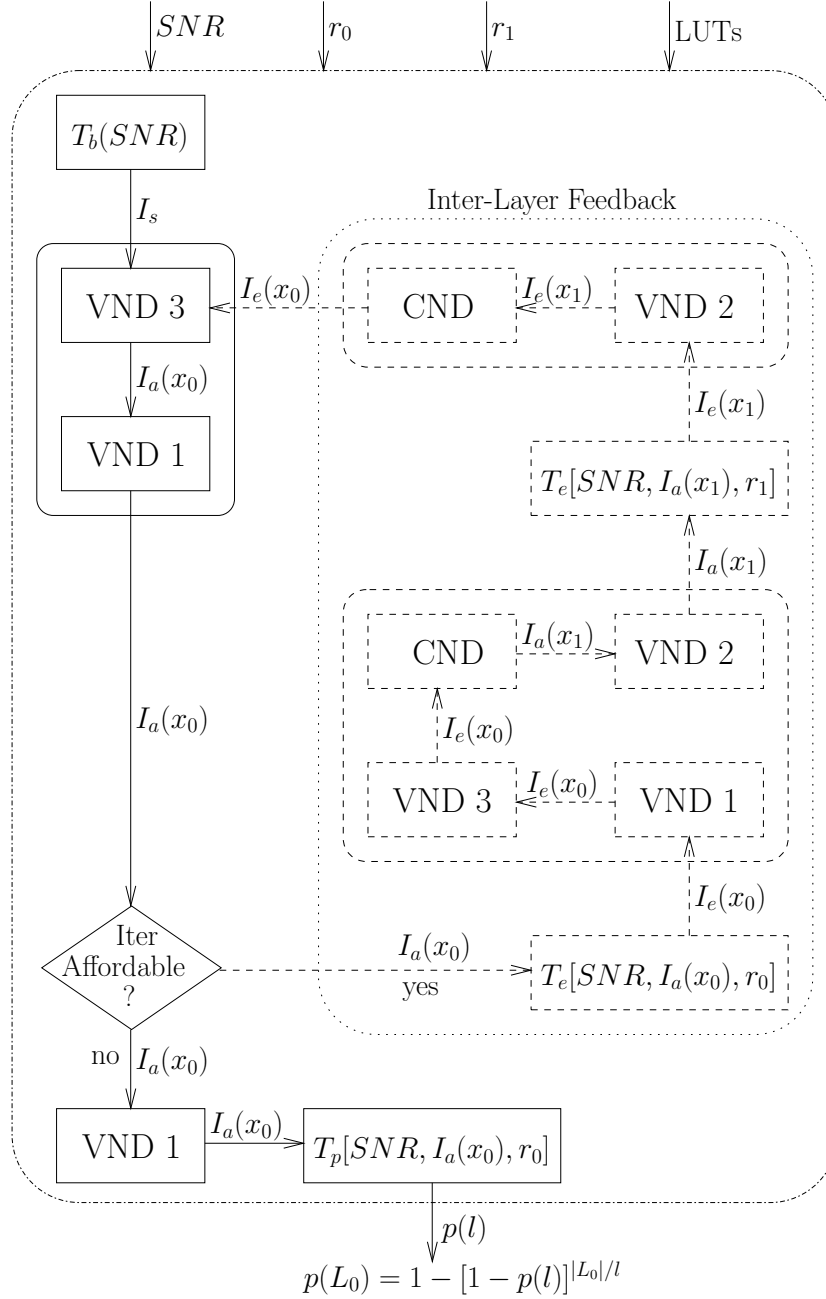


Figure 3.18: Flow-chart of the PER estimation for the IL-FEC coded BL L_0 , where $p(L_0)$ will be employed by the “Code Rate Optimization” block of Figure 3.16 for resolving the objective function of Eq. (3.10).

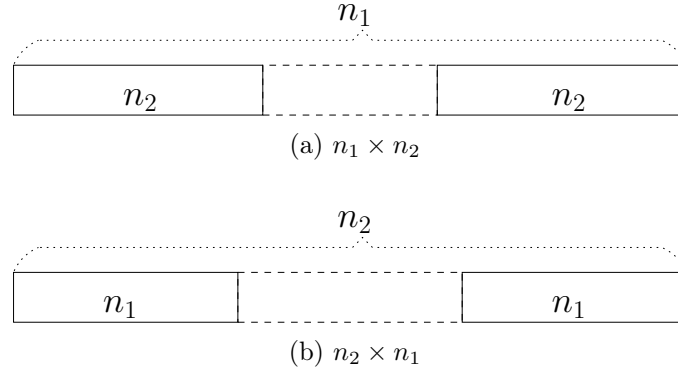


Figure 3.19: A packet having a length of $(n_1 \times n_2)$ bits may be divided into n_1/n_2 shorter packets with a length of n_2/n_1 bits.

Then, following the "VND2-CND-VND3-VND1" process of Figure 3.18, the improved MI value $I_a(x_0)$ of $L_a(x_0^i)$ can be calculated. Finally, the PER associated with the packet length of l^{11} , denoted as $p(l)$, may be estimated as $T_p[SNR, I_a(x_0), r_0]$, which is also shown in Figure 3.18. Below, we now detail the method of deriving the PER $p(L_0)$ of the BL L_0 from $p(l) = T_p[SNR, I_a(x_0), r_0]$.

The burst error distribution of RSC codec has been investigated in [160], which is independent of the packet length. Let us consider a RSC-decoded packet having a length of $(n_1 \times n_2)$ bits. Then this packet may be partitioned in two ways, as displayed in Figure 3.19. Specifically, it may be divided into n_1 packets, each carrying n_2 bits or n_2 packets associated with n_1 bits each. Assuming that $p(n_i)$ indicates the PER of the n_i -bit packet, based on Figure 3.19a the PER $p(n_1 \cdot n_2)$ may be estimated as

$$p(n_1 \cdot n_2) = 1 - [1 - p(n_2)]^{n_1}, \quad (3.19)$$

where $p(n_2)$ is the PER of the n_1 packets of Figure 3.19a. Similarly, we have $p(n_1 \cdot n_2) = 1 - [1 - p(n_1)]^{n_2}$ based on Figure 3.19b. Then, for arbitrary numerical values of n_1, n_2 we have

$$p(n_1) = 1 - [1 - p(n_2)]^{n_1/n_2}. \quad (3.20)$$

Upon assuming that n_1, n_2 of Eq. (3.20) are given by $|L_0|$ and l , respectively, the PER $p(L_0)$ of the BL L_0 may be estimated as

$$\begin{aligned} p(L_0) &= f_0(SNR, |L_0|, |L_1|, r_0, r_1) \\ &= 1 - [1 - T_p[SNR, I_a(x_0), r_0]]^{|L_0|/l}, \end{aligned} \quad (3.21)$$

¹¹The packet containing l bits is employed for generating the LUT $T_p[SNR, I_a(x_0), r_0]$.

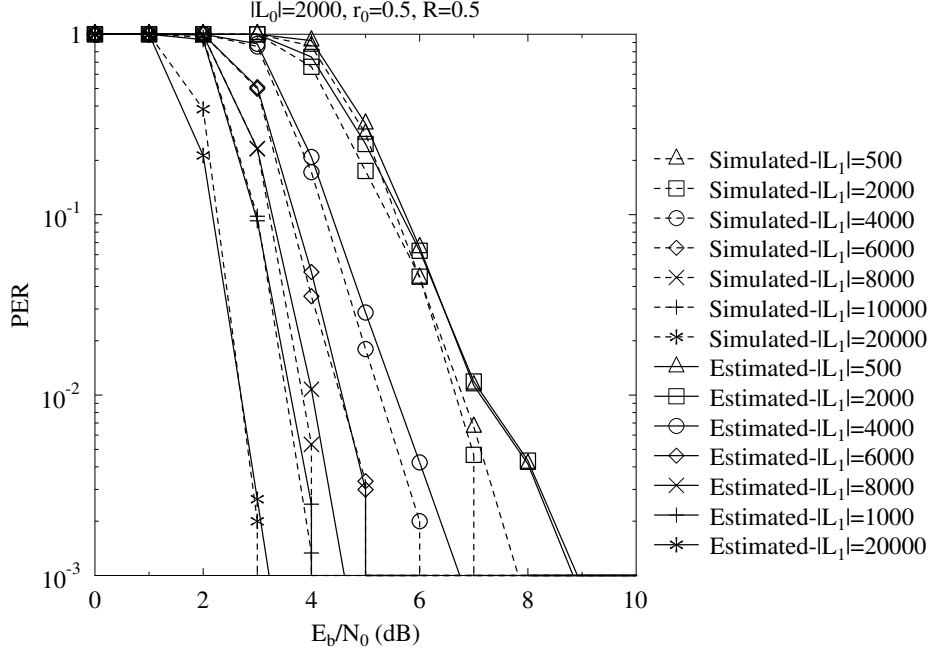


Figure 3.20: Comparison of the simulated and estimated PER of the BL of Figure 3.16, where the parameters $|L_0| = 2000$, $r_0 = 0.5$, $R = 0.5$ are employed.

where l is the packet length employed for generating the LUT T_p .

For visualizing the accuracy of our proposed PER estimation for the BL, we assume $|L_0| = 2000$, $r_0 = 0.5$ and $R = 0.5$, while the RSC codec is configured by the generator polynomials of [1011, 1101, 1111]. Diverse EL packet length of L_1 are tested. The resultant simulated and estimated PERs of the BL L_0 are displayed in Figure 3.20 for different SNR values. As observed in Figure 3.20, the estimated PER does not perfectly match the simulated PER, but it closely follows the same trend, especially in the PER region above 10^{-2} , which is our area of interest. Note that the estimation error may be introduced both by the LUTs and by the near-Gaussian distributed LLRs output by the demodulator.

3.4.2.2 FEC Coded EL

When the BL L_0 is correctly decoded, infinite LLRs will be input to the CND 3 of Figure 3.16 [75], hence the LLRs having the MI value of I_s will be input to the VND 2 by the CND. Therefore the PER of the EL having a length of C bits may be estimated as $T_p(SNR, I_s, r_1)$ with the aid of the LUT T_p . Then, similar to the Eq. (3.21), the PER $p(L_1)$ of the EL L_1 may be finally estimated as

$$\begin{aligned} p(L_1) &= f_1(SNR, |L_1|, r_1) \\ &= 1 - [1 - T_p(SNR, I_s, r_1)]^{|L_1|/l}, \end{aligned} \quad (3.22)$$

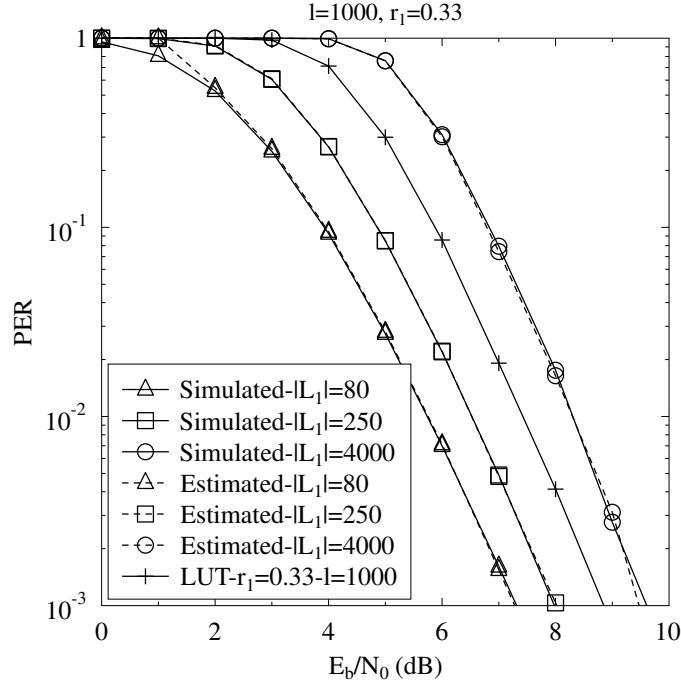


Figure 3.21: Comparison of the simulated and estimated PER for the EL of Figure 3.16, where the parameters $l = 1000$, $r_1 = 0.33$ are employed..

where l is the packet length employed for generating the LUT T_p .

For visualizing the accuracy of our proposed EL PER estimation, we assume $r_1 = 0.33$, while the RSC codec is configured by the generator polynomials of [1011, 1101, 1111]. Moreover, we consider $|L_1|$ values ranging from 80 to 4000, while the LUT $T_p(SNR, I_s, r_1)$ is generated based on $l = 1000$. The resultant simulated and estimated PER of the EL L_1 is displayed in Figure 3.21. Observe in Figure 3.21 that the estimation error is much smaller than that in Figure 3.20 due to the fact that the EL is protected by an RSC codec without using IL techniques.

3.4.3 Optimized Coding Rates

Based on Section 3.4.2, the PERs of L_0 and L_1 are estimated according to the specified coding rates of r_0 and r_1 for the BL and EL, respectively. Hence the expected distortion $E[d(L_0, L_1)]$ imposed by the wireless transmission using rates of r_0 and r_1 may be readily calculated using Eq. (3.9). In order to find the rates r_0, r_1 minimizing the distortion $E[d(L_0, L_1)]$ in Eq. (3.10), a full search of the coding rate space may be performed. Note that r_1 is determined by Eq. (3.11) for a given r_0 , hence the full search is performed in a one-dimensional space to find the optimal r_0 .

To elaborate a little further, we consider the scenario of $SNR = 10dB$, $R = 1/3$, $|L_0| = |L_1| = 100$, $d(L_0) = 20$, $d(L_1) = 10$. Based on the conditions formulated in

Eq. (3.11), the coding rate r_1 of L_1 is determined as follows:

$$r_1 = \frac{|L_1|}{\frac{|L_0|+|L_1|}{R} - \frac{|L_0|}{r_0}} = \frac{r_0}{6 \cdot r_0 - 1} \quad (3.23)$$

$$r_0 \leq 1, r_1 \leq 1.$$

Then, based on Eqs. (3.7), (3.8) and (3.9) the expected value of $E[d(L_0, L_1)]$ in Eq. (3.10) may be formulated as

$$\begin{aligned} E[d(L_0, L_1)] &= p(L_0) \cdot d(L_0) + [1 - p(L_0)] \cdot p(L_1) \cdot d(L_1) \\ &= 20 \cdot f_0 \left(10, 100, 100, r_0, \frac{r_0}{6 \cdot r_0 - 1} \right) \\ &\quad + 10 \cdot \left[1 - f_0 \left(10, 100, 100, r_0, \frac{r_0}{6 \cdot r_0 - 1} \right) \right] \cdot f_1 \left(10, 100, \frac{r_0}{6 \cdot r_0 - 1} \right) \\ &= g(r_0), \end{aligned} \quad (3.24)$$

where the functions f_0 and f_1 are discussed in the 'PER Estimation' Section 3.4.2. Note that we introduced the short-hand of $E[d(L_0, L_1)] = g(r_0)$ in Eq. (3.24) for simplicity. Based on Eq. (3.23), we can readily arrive at $\frac{1}{5} \leq r_0 \leq 1$. Then we artificially limit r_0 to the following discrete practical values $[0.2 : 0.05 : 1]^{12}$. Finally, the solution r_0 of the objective function $\arg \min \{E[d(L_0, L_1)]\}$ in Eq. (3.10) may be readily found to be:

$$\left\{ r_0 | \arg \min_{r_0} g(r_0), r_0 \in \{0.2 : 0.05 : 1\} \right\}, \quad (3.25)$$

which can be solved by performing a one-dimensional full search. Then, given r_0 the coding rate r_1 may be readily expressed from Eq. (3.23).

3.4.4 Overheads

All the optimization operations detailed in Section 3.4 are carried out at the transmitter. Below, we discuss the overheads imposed at the transmitter, while the overheads imposed at the receiver are given in [75]. The overheads imposed at the transmitter include the estimation of $d(\cdot)$, the generation of LUTs, the estimation of $p(\cdot)$, the imposed frame delay and the full search detailed in Section 3.4.3. Among these overheads, the generation of LUTs only imposes extra off-line design-time, while the estimation of $d(\cdot)$, $p(\cdot)$ and the full search impose extra on-line run-time complexity.

¹²The value of r_0 ranges from 0.2 to 1 with a constant step of 0.05.

For notational simplicity, we use n_{snr} , n_I , n_r to denote the number of the parameters SNR , I_s , r , respectively, which are used for generating the LUTs. For example, if the values of I_s range from 0 to 1 with a step size of 0.01, we have $n_I = 101$. In the following, we will analyze these overheads in order to validate our system.

3.4.4.1 Estimation of $d(\cdot)$

$d(L_i)$ is estimated in a similar manner to the procedure of [56], where the distortion $d(L_0)$ may be obtained by decoding the bitstream in the presence of a corrupted BL L_0 . Alternatively, the solutions of [161, 57, 59] may be applied in our system.

3.4.4.2 Generation of LUTs

Three LUTs were generated in our proposed solution. The LUT $T_b(SNR)$ characterizes the channel and transceiver, hence this table has to be regenerated when the channel or the transceiver are changed. However, this LUT only has a single variable, hence it is straight-forward to generate the LUT $T_b(SNR)$. Additionally, the LUTs $T_e(SNR, I_s, r)$ and $T_p(SNR, I_s, r)$ can be simultaneously generated by simulations, since they have the same variables. However, they are independent of the channel and transceiver, but they are dependent on the configuration of the FEC code. Hence the LUTs T_e and T_p have to be regenerated, when the FEC is reconfigured. *Note that all the LUTs are independent of the video sequences and that these LUTs are generated during the design process.* The size of these LUTs depends on the variables, hence the LUTs T_b , T_e , T_p have sizes of n_{snr} , $(n_{snr} \cdot n_I \cdot n_r)$, $(n_{snr} \cdot n_I \cdot n_r)$, respectively.

3.4.4.3 Estimation of $p(\cdot)$

The CND and VND are involved in the PER estimation process, as well as in the search through the LUTs. As detailed in Section 3.4.1, the MI flows of the VND and CND impose a low computational complexity. Furthermore, in our system the values of each variable are chosen using a constant step-size, which guarantees low complexity, while searching through the LUTs. For example, let us assume that the vector $[0 : 0.01 : 1]^{13}$ is employed for representing the values of I_s for generating the LUTs. In this case, the I_s value of 0.5 can be directly located at index 50.

3.4.4.4 Full Search of r_0

n_r coding rates of r_0 may be tentatively tested, while r_1 is determined by Eq. (3.11) for a given r_0 . Hence the complexity of the full search may be expressed as $O(C \times n_r)$, where C is a constant, representing the complexity of estimating $p(L_i)$. However,

¹³The value of I_s ranges from 0 to 1 with a constant step of 0.01.

	Football	Foreman
Representation	YUV 4:2:0	YUV 4:2:0
Format	CIF	CIF
Bits Per Pixel	8	8
FPS	15	30
Number of Frames	30	30
Video Codec	H.264 PM	H.264 PM
Bitrate	1522 kbps	655 kbps
Error-Free PSNR	37.6 dB	38.4 dB
Error Concealment	Motion-Copy	Motion-Copy

Table 3.9: The parameters of the video sequences employed for the characterization of the system in Figure 3.1.

when more layers are encapsulated in the source bitstream, the full-search based complexity increases exponentially, leading to a multi-dimensional optimization problem, which has been widely studied in the literature [162, 163, 164]. Specifically, the adaptive particle swarm optimization (APSO) technique of [164] may be readily employed for finding the global optimum in real-time. Note that in the scenarios where as few as 2-4 layers in the range of 2 to 4 are generated, even elite-force full-search is realistic at a modest complexity.

3.4.4.5 Delay

Our technique is implemented using the SVC of H.264 as detailed in Section 2.5, where each video frame can be encoded into a single BL and multiple ELs. Since the IL encoding and decoding process is performed within each frame, no extra delay is imposed by our proposed technique. Hence, the system can be applied in low-delay applications.

3.5 System Performance for Partitioned H.264

Let us continue by benchmarking our proposed IL-FEC-LSSTC system against the traditional UEP aided FEC-LSSTC system using a RSC and a near-capacity turbo codec [121]. Two 30-frame video sequences, namely the Foreman and Football clips, represented in (352×288) -pixel common intermediate format (CIF) and 4:2:0 YUV format were encoded using the JM/AVC 15.1 H.264 reference video codec operated

in its data partitioning aided mode. The video scanning rates expressed in frame per second (FPS) were 30 and 15 for the Foreman and Football sequences, respectively. The motion-copy¹⁴, based error concealment tool built into H.264 reference codec was employed for the sake of combating the effects of channel impairments. Moreover, the H.264 encoder was configured to generate fixed-byte¹⁵ slices, as defined in [14]. Both of the 30-frame video sequences were encoded into an intra-coded frame (I-frame), followed by 29 predicted frames (P-frames). The bi-directionally predicted frame (B-frame) was disabled due to the fact that it relies on both previous and future frames for decoding, which may introduce more error propagation as well as additional delay. All the above configurations jointly result in a bitrate of 655 kbps and an error-free peak-signal to noise ratio (PSNR) of 38.4 dB for the Foreman sequence. On the other hand, the coded Football bitstream has a bitrate of 1522 kbps and an error-free PSNR of 37.6 dB. We employed the Foreman and Football sequences in order to show the suitability of our scheme for the transmission of both low-motion and high-motion video. The parameters of the employed sequence are shown in Table 3.9, while our system parameters are listed in Table 3.5.

The H.264-compressed bitstream was FEC encoded and transmitted on a network abstract layer unit (NALU) [14] basis, which is the smallest element to be used by the source decoder. At the receiver, each error-infested NALU must be dropped by the video decoder, if errors are detected by the CRC check. All experiments were repeated 100 times for the sake of generating smooth performance curves.

Below, we will firstly describe the error-protection arrangements in Section 3.5.1. Then we will characterize the attainable BER versus channel SNR performance and PSNR versus channel SNR performance employing a lower-complexity RSC codec in Section 3.5.2. We continue by characterizing the attainable BER versus channel SNR performance and PSNR versus channel SNR performance employing a turbo codec in Section 3.5.3. Finally, in Section 3.5.4 we will quantify the system's computational complexity by counting the number of decoding operations executed.

3.5.1 Error Protection Arrangements

In the simulations, we employ the overall coding rate¹⁶ of 1/2 for both EEP and UEP schemes. For each compressed bitstream, all NALUs were scanned to calculate the

¹⁴When the information of a macroblock (MB) is lost, the motion vector of this MB may be copied or estimated from its adjacent MBs or previously decoded reference frames. Then, the MB may be reconstructed using the estimated motion vector.

¹⁵In this mode, the H.264/AVC codec will endeavor to encode a frame into multiple slices, each having a fixed number of bytes.

¹⁶Arbitrary overall coding rates such as 2/3, 1/3, 1/4, etc. can be readily applied by changing the channel codec parameters and the puncturers.

Error Protection Arrangements	Code Rates			
	Type A	Type B	Type C	Average
EEP	0.5/0.5	0.5/0.5	0.5/0.5	0.5/0.5
UEP1	0.35/0.40	0.57/0.65	0.57/0.65	0.5/0.5
UEP2	0.45/0.55	0.52/0.46	0.52/0.46	0.5/0.5
UEP3	0.65/0.60	0.47/0.43	0.47/0.43	0.5/0.5
UEP4	0.75/0.70	0.45/0.39	0.45/0.39	0.5/0.5
UEP5	0.85/0.80	0.44/0.37	0.44/0.37	0.5/0.5
UEP6	0.95/0.90	0.43/0.35	0.43/0.35	0.5/0.5

Table 3.10: Coding rates of different error protection arrangements used in the system of Figure 3.1 for the Football/Foreman sequence. The code-rates were adjusted by variable-rate puncturers.

total number of bits for the A, B, and C partitions. Let us assume that the A, B and C partitions have a total N_a , N_b and N_c bits, respectively. The A, B, C streams have coding rates of r_a , r_b and r_c , respectively. Then the following equation must be satisfied for the sake of guaranteeing that the overall coding rate remains 1/2:

$$2 \times (N_a + N_b + N_c) = \frac{N_a}{r_a} + \frac{N_b}{r_b} + \frac{N_c}{r_c}. \quad (3.26)$$

Again, the A stream is the most important layer, while the B and type C bitstreams are the ELs, where the bitstream B and C are similarly important. Hence in all the error protection arrangements we have $r_b = r_c$. More specifically, we first select a specific value to r_a , then the value of $r_b = r_c$ was calculated as follows:

$$r_b = \frac{N_b + N_c}{2 \times (N_a + N_b + N_c) - \frac{N_a}{r_a}}. \quad (3.27)$$

Note that the total number of bits for each partitions of the different video sequences may be different, which results in different protection arrangements. Based on the above, the five error protection arrangements conceived for the Football and Foreman sequences are shown in Table 3.10, which may be readily combined with arbitrary EEP or UEP schemes, where variable-rate puncturers were designed and employed to achieve a specific coding rate.

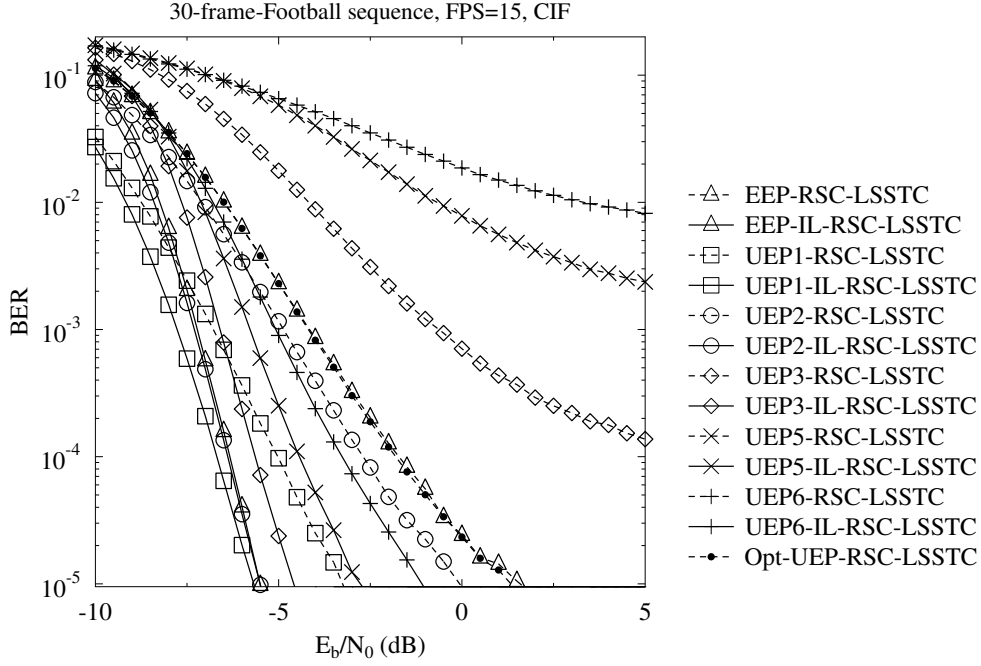


Figure 3.22: BER versus E_b/N_0 performance for the A partition of the *Football* sequence, including the RSC coding schemes of Table 3.10 and the Opt-UEP-RSC-LSSTC [56]. The schematic of Figure 3.1 was used.

3.5.2 System Performance using RSC Codec

In this section, we benchmark our proposed system using the RSC codec of Table 3.5. All the error protection arrangements of Section 3.5.1 will be utilized. Furthermore, in [56] an UEP algorithm was proposed, which the authors of [56] referred to as the optimal UEP. We used this scheme as a benchmark, which we refer to as the Opt-UEP-RSC-LSSTC arrangement.

The BER curves of the A partition in the Football sequence are displayed in Figure 3.22, where the performance of the error protection schemes of Table 3.10 are illustrated. Observe in Figure 3.22 that the schemes using the IL-RSC codec achieve a reduced BER compared to their benchmarks. Specifically, the EEP-IL-RSC-LSSTC scheme outperforms the EEP-RSC-LSSTC benchmark by about 7.2 dB at a BER of 10^{-5} . Furthermore, among all the error protection arrangements, the UEP1-IL-RSC-LSSTC scheme achieves the best BER performance due to the strong error protection assigned for the A partition. Hence, we may conclude that the UEP aided IL-RSC schemes are capable of providing an improved system performance compared to the traditional UEP aided RSC codec. On the other hand, the Opt-UEP-RSC-LSSTC system achieves similar BER performance to that of the EEP-RSC-LSSTC scheme.

The BER versus E_b/N_0 performance of the B partition for the Football sequence is presented in Figure 3.23. Similar trends were observed for the C partition as well,

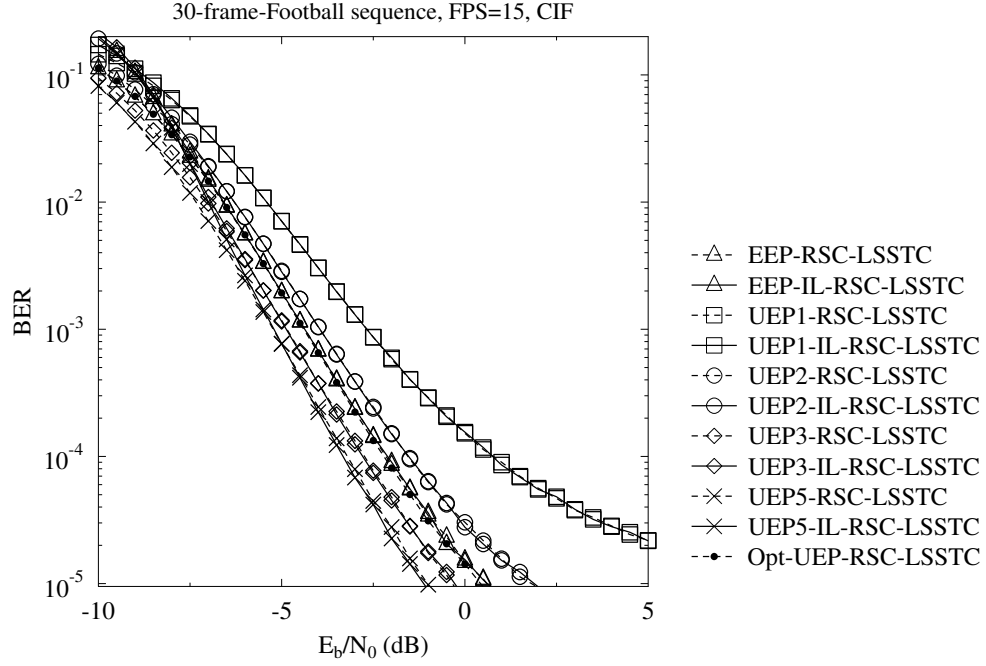


Figure 3.23: BER versus E_b/N_0 performance for the B partition of the *Football* sequence, including the RSC coding schemes of Table 3.10 and the Opt-UEP-RSC-LSSTC [56]. The schematic of Figure 3.1 was used.

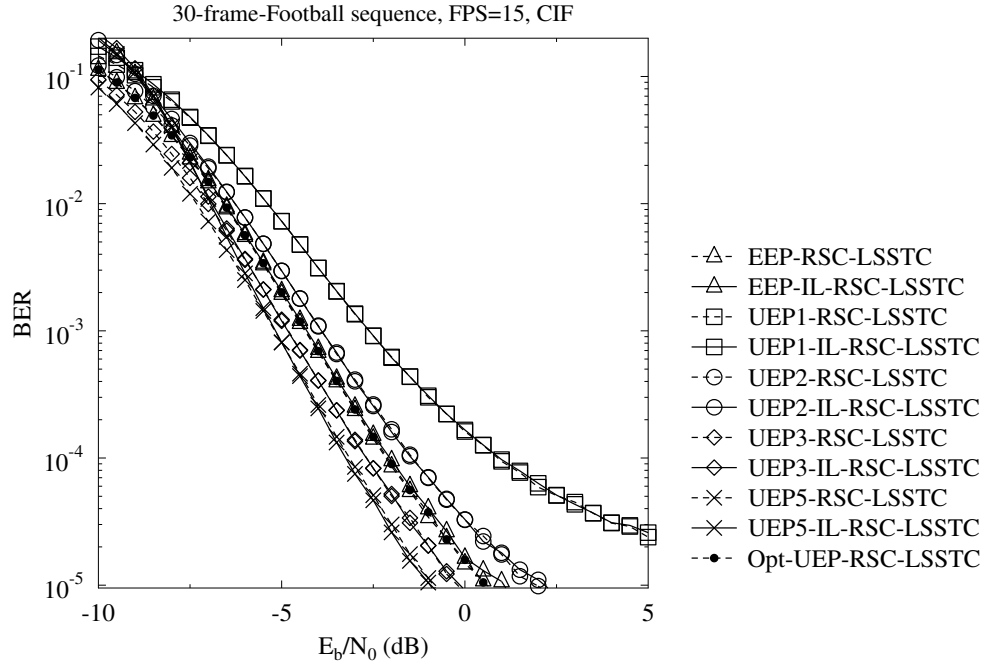


Figure 3.24: BER versus E_b/N_0 performance for the C partition of the *Football* sequence, including the RSC coding schemes of Table 3.10 and the Opt-UEP-RSC-LSSTC [56].

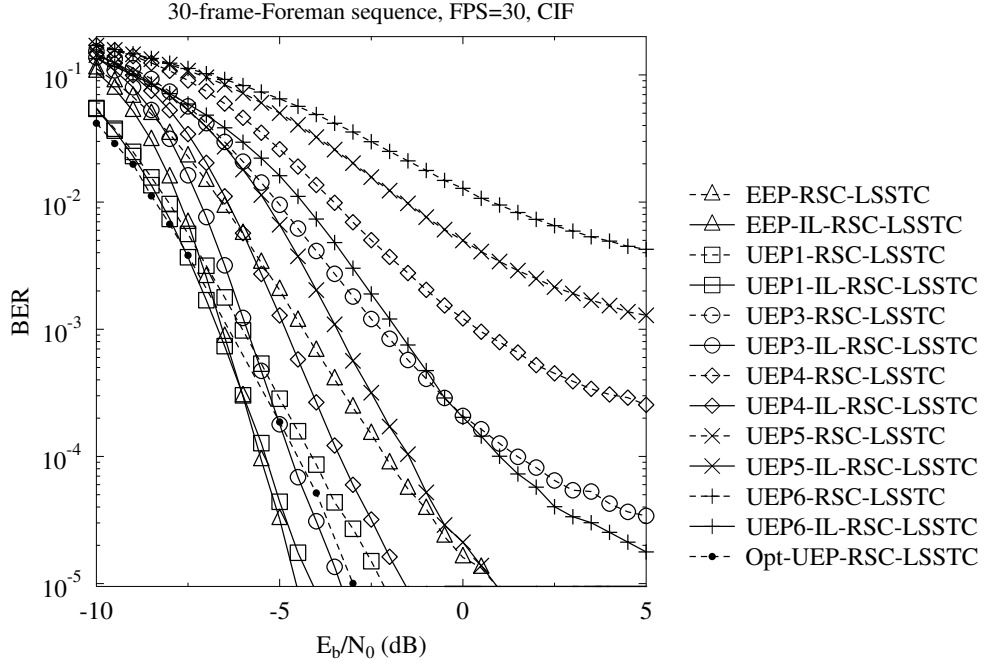


Figure 3.25: BER versus E_b/N_0 performance for the A partition of the *Foreman* sequence, including the RSC coding schemes of Table 3.10 and the Opt-UEP-RSC-LSSTC [56]. The schematic of Figure 3.1 was used.

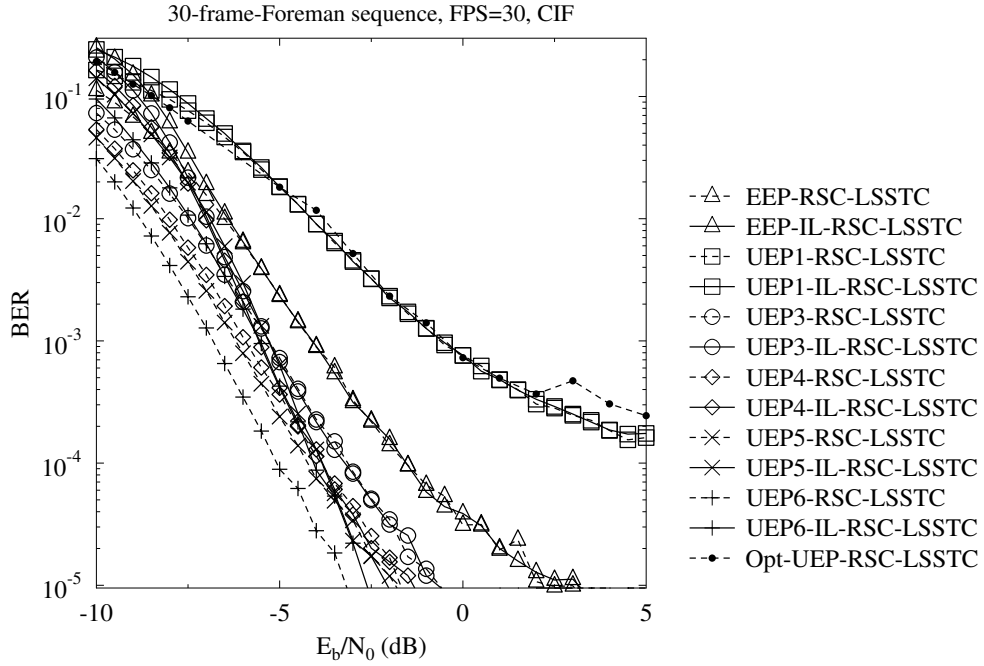


Figure 3.26: BER versus E_b/N_0 performance for the B partition of the *Foreman* sequence, including the RSC coding schemes of Table 3.10 and the Opt-UEP-RSC-LSSTC [56]. The schematic of Figure 3.1 was used.

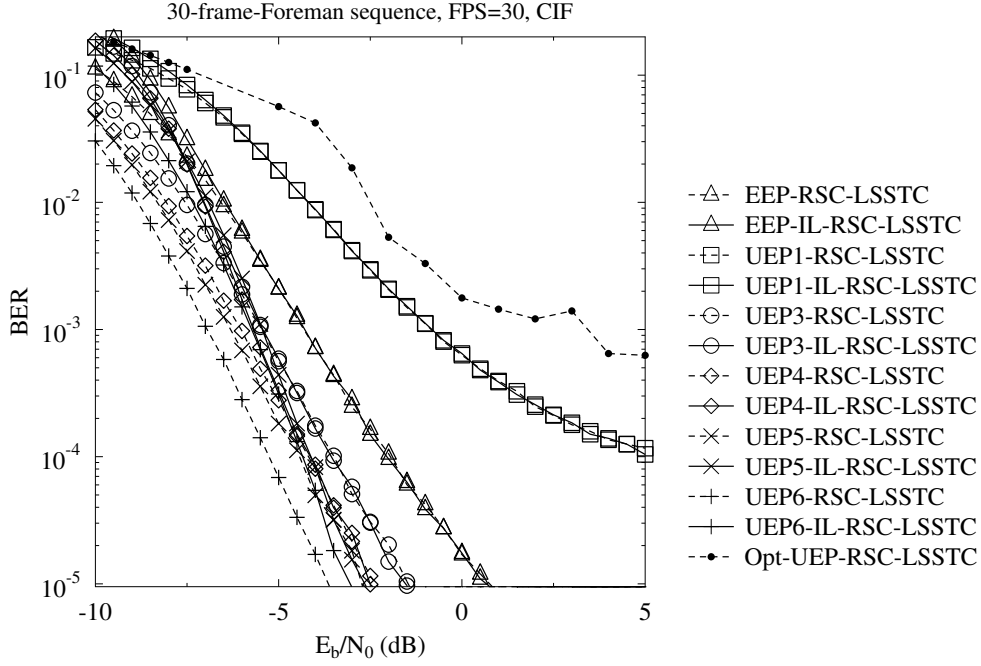


Figure 3.27: BER versus E_b/N_0 performance for the C partition of the *Foreman* sequence, including the RSC coding schemes of Table 3.10 and the Opt-UEP-RSC-LSSTC [56]. The schematic of Figure 3.1 was used.

which are characterized in Figure 3.24. Observe in Figure 3.23 that the performance of the schemes using IL-RSC is slightly worse than that of their benchmarks. This is due to the fact that more errors may be introduced into the B partition, when the A partition cannot be correctly decoded. In this scenario the B partition must be dropped in the traditional UEP aided RSC-LSSTC schemes. Hence the error propagation to the B partition does not further degrade the situation. Moreover, the BER versus E_b/N_0 performance observed for the A, B, C partitions of the *Foreman* sequence are displayed in Figures 3.25, 3.26 and 3.27, respectively, where similar trends prevail to those seen in Figures 3.22, 3.23 and 3.24. Specifically, for the A partition Figure 3.25 suggests that the schemes relying on the IL techniques outperform their benchmarks dispensing with IL techniques. On the other hand, the schemes dispensing with the IL techniques outperform the corresponding IL-coded benchmarks, as suggested by Figures 3.26 and 3.27 for the B and C partitions, respectively. This is due to the fact that the quality of the B and C partitions is sacrificed for the sake of protecting the more important BL.

The PSNR versus E_b/N_0 performance recorded for the Football sequence is shown in Figure 3.28, where we observe that the EEP-RSC-LSSTC scheme achieves the best performance among all the systems without IL techniques, because the A partition carries only the video header information and fails to assist the H.264 decoder in

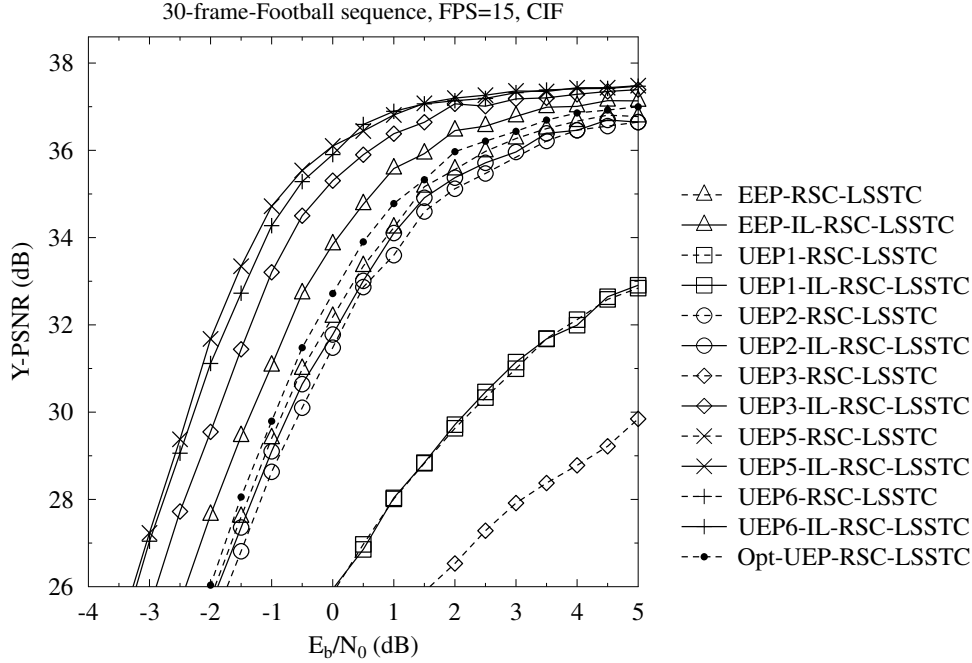


Figure 3.28: PSNR versus E_b/N_0 performance for the *Football* sequence, including the RSC coding schemes of Table 3.10 and the Opt-UEP-RSC-LSSTC [56]. The schematic of Figure 3.1 was used.

concealing the residual errors, when the B and C partitions are corrupted. Furthermore, the systems using our proposed IL-RSC-LSSTC model outperform their corresponding benchmarks. Specifically, the UEP5-IL-RSC-LSSTC constitutes the best protection arrangement among all IL-RSC schemes, which achieves a power reduction of about 3 dB¹⁷ compared to the EEP-RSC-LSSTC scheme at a PSNR of 36 dB. Alternatively, about 3.7 dB of PSNR video quality improvement may be observed at a channel SNR of 0 dB. On the other hand, the Opt-UEP-RSC-LSSTC system dispensing with the IL technique slightly outperforms the EEP-RSC-LSSTC scheme, namely by a power reduction of about 0.5 dB at a PSNR of 36 dB. The UEP5-IL-RSC-LSSTC substantially outperforms the Opt-UEP-RSC-LSSTC arrangement, namely by a power reduction of about 2.5 dB at a PSNR of 36 dB or alternatively, about 3.4 dB of PSNR video quality improvement may be observed at an E_b/N_0 of 0 dB. A subjective comparison of the UEP5-IL-RSC-LSSTC and EEP-RSC-LSSTC arrangements for the Football sequence is presented in Figure 3.30.

For providing further insights for video scenes having different motion-activity, the PSNR versus E_b/N_0 performance of the IL-RSC-LSSTC model is presented in Figure 3.29 using the Foreman sequence, when employing the protection arrangements of Table 3.10. Similar to the Football sequence, the traditional UEP technique can hardly

¹⁷The power reduction is read horizontally. Specifically, the UEP5-IL-RSC-LSSTC achieves the PSNR of 36 dB with 3 dB less power than the EEP-RSC-LSSTC scheme.

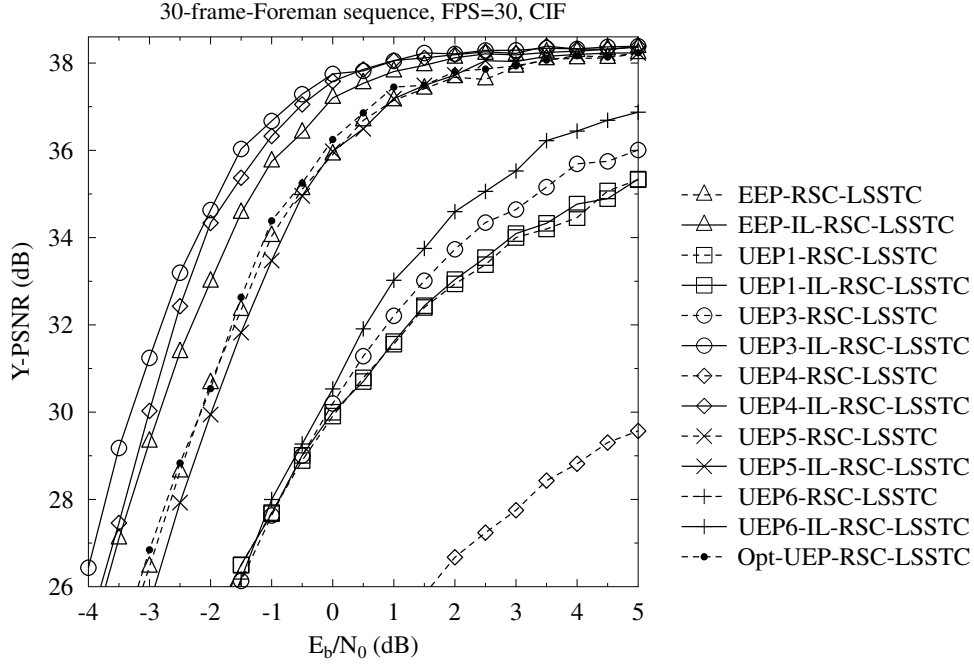


Figure 3.29: PSNR versus E_b/N_0 performance for the *Foreman* sequence, including the RSC coding schemes of Table 3.10 and the Opt-UEP-RSC-LSSTC [56]. The schematic of Figure 3.1 was used.

improve the reconstructed video quality by allocating more FEC redundancy to the more important layers. By contrast, about 2 dB of power reduction is achieved by the UEP3-IL-RSC-LSSTC arrangement compared to the EEP-RSC-LSSTC scheme at a PSNR of 37 dB. Alternatively, about 3.2 dB of PSNR video quality improvement may be observed at a channel SNR of -1 dB. Similar to the Football sequence, a limited gain can be observed for the Opt-UEP-RSC-LSSTC system compared to the EEP-RSC-LSSTC scheme, while the UEP5-IL-RSC-LSSTC substantially outperforms the Opt-UEP-RSC-LSSTC, namely by about 1.8 dB at a PSNR of 37 dB. A subjective comparison of the UEP3-IL-RSC-LSSTC and EEP-RSC-LSSTC arrangements for the Foreman sequence is presented in Figure 3.30.

We may conclude from the above discussion that the A partition should be assigned a code-rate of 0.85 and 0.60 for the Football and Foreman sequence, respectively, for the sake of achieving the best overall system performance, when employing the RSC codec, which contradicts to the traditional UEP strategy. The main reason for this is that the inter-layer aided RSC decoder can still successfully recover the weaker protected A partition relying on the extrinsic information fed back from the B and C partitions with the aid of inter-layer decoding, because B and C are more strongly protected than the A partition.



Figure 3.30: Video comparison at $E_b/N_0 = -2.5$ dB for the Football and Foreman sequences. The first column indicates the original frames. The second column indicates the EEP-IL-RSC-LSSTC decoded frames. The third column indicates the Opt-UEP-RSC-LSSTC [56] decoded frames. The fourth column represents the UEP5-IL-RSC-LSSTC and UEP3-IL-RSC-LSSTC decoded frames for the Football and Foreman sequences, respectively.

3.5.3 System Performance using Turbo Coding

In this section, we benchmark our proposed system using a turbo codec, which consists of two identical RSC codec relying on the generator polynomials of $G = [111, 101]$. Furthermore, all the FEC arrangements of Section 3.5.1 will be utilized.

Firstly, we present the BER versus E_b/N_0 performance employing the turbo codec for the Football sequence, noting that similar trends were observed for the Foreman sequence. The BER versus E_b/N_0 curves for the A partition are displayed in Figure 3.31, comparing the performance of the different schemes of Table 3.10. Observe in Figure 3.31 that the IL-turbo scheme achieved a reduced BER compared to the benchmarks. Specifically, the EEP-IL-turbo-LSSTC scheme outperforms the EEP-turbo-LSSTC benchmark by about 2 dB at a BER of 10^{-5} . Furthermore, among all the error protection arrangements, the UEP1-turbo-LSSTC scheme achieves the best BER performance due to the high error protection assigned to the A partition. Hence, we may conclude that the UEP aided IL-turbo scheme is capable of providing an improved system performance compared to the traditional UEP aided turbo codec.

The BER versus E_b/N_0 performance of the type B partition for the Football sequence is presented in Figure 3.32, noting that similar trends were observed also for the C partition as displayed in Figure 3.33. Observe in Figure 3.32 that the performance of the schemes using the IL-turbo arrangements is worse than that of their benchmarks. This is due to the fact that more errors may be introduced

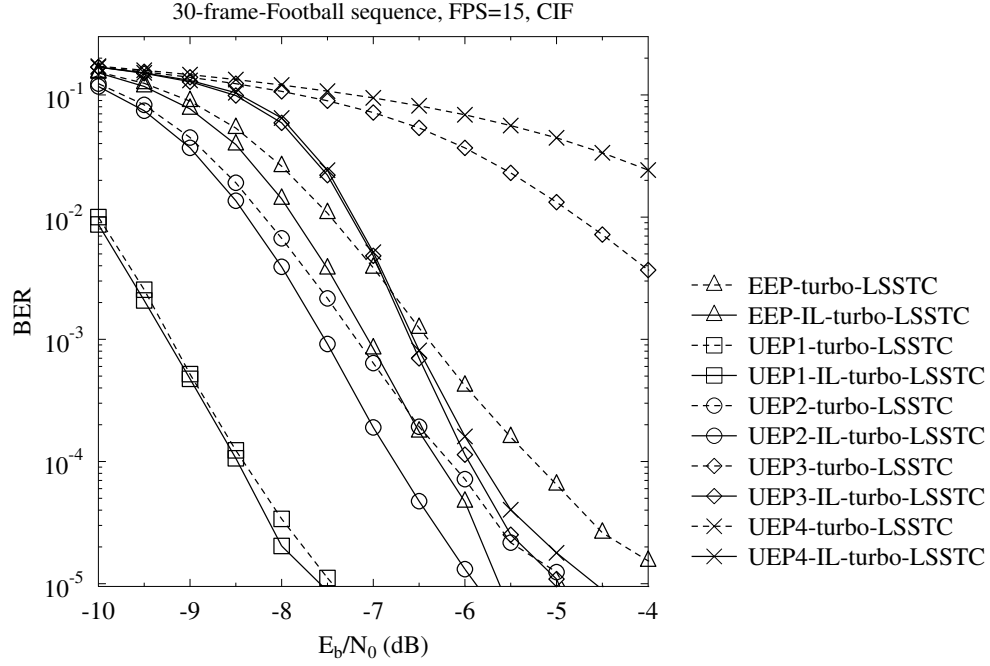


Figure 3.31: BER versus E_b/N_0 performance for the type A partition of the *Football* sequence using the turbo coding schemes of Table 3.10. The schematic of Figure 3.1 was used.

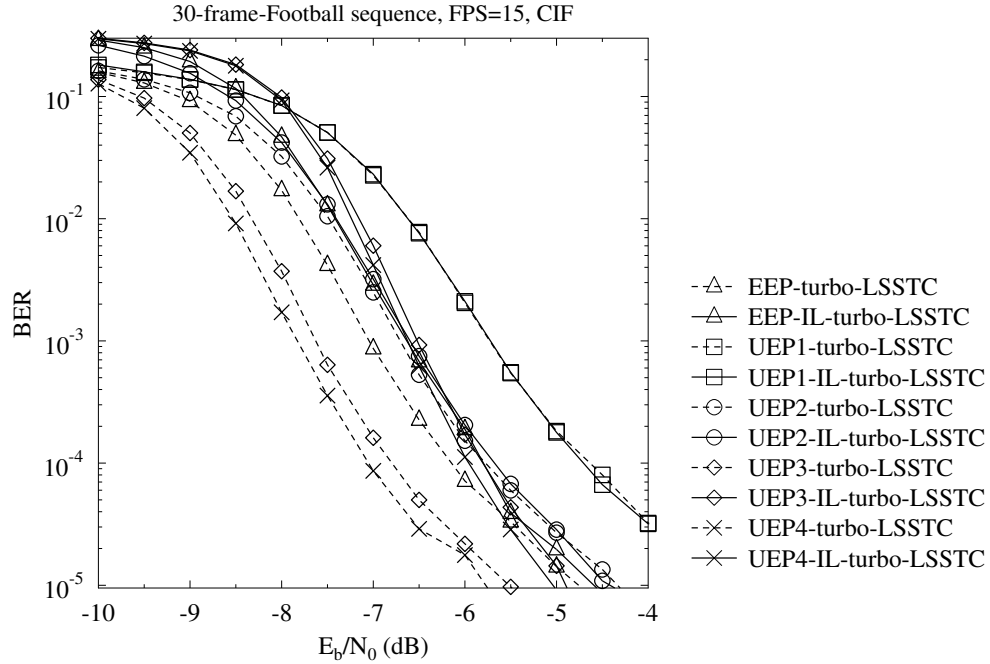


Figure 3.32: BER versus E_b/N_0 performance for the type B partition of the *Football* sequence using the turbo related coding schemes of Table 3.10. The schematic of Figure 3.1 was used.

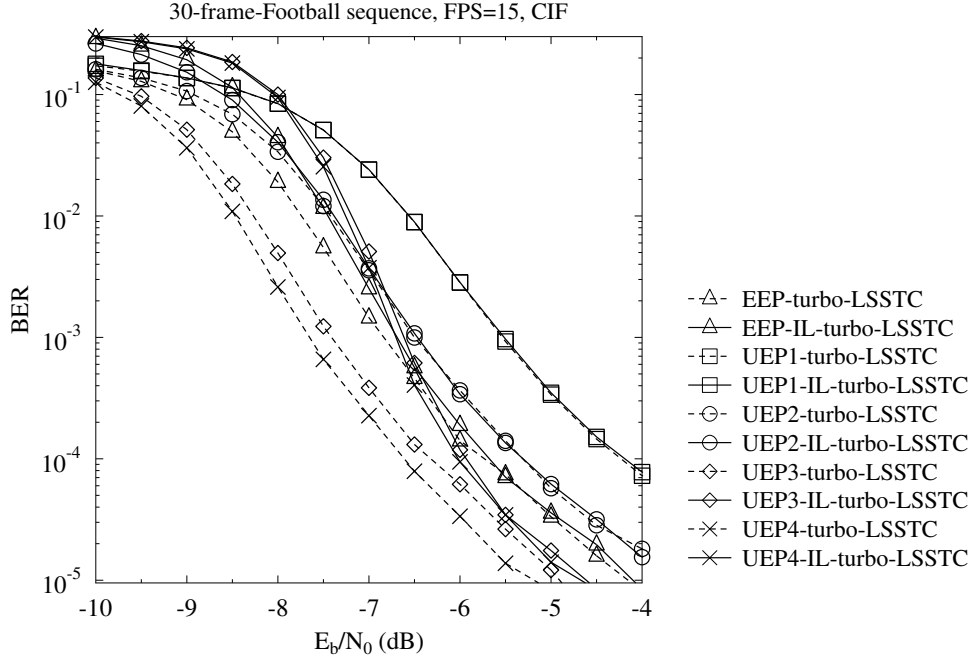


Figure 3.33: BER versus E_b/N_0 performance for the type C partition of the *Football* sequence using the turbo related coding schemes of Table 3.10. The schematic of Figure 3.1 was used.

into the type B partition, when the A partition cannot be correctly decoded. Note however that extra bit errors are introduced only if the A partition is corrupted, and in this scenario the B partition must be dropped in the traditional UEP aided turbo-LSSTC schemes. Hence the error propagation to the B partition does not degrade the situation further. Moreover, the BER versus E_b/N_0 performance for the A, B, C partitions of Foreman sequence is displayed in Figures 3.34, 3.35, 3.36, respectively, where similar trends to the Figures 3.31, 3.32, 3.33 are observed.

The PSNR versus E_b/N_0 performance recorded for the Football sequence is shown in Figure 3.37, where the UEP2-turbo-LSSTC scheme is seen to achieve the best performance, albeit it has a limited gain compared to the EEP-turbo-LSSTC system. The reason for this trend is that the A partition predominantly carries the video header information and fails to assist the H.264 decoder to conceal the residual errors, when the B and C partitions are corrupted. Furthermore, the systems using our proposed IL-turbo-LSSTC model outperform their corresponding benchmarks. Specifically, the UEP3-IL-turbo-LSSTC, is the best protection arrangement among all IL-turbo schemes, which achieves a power reduction of 1.1 dB compared to the UEP2-turbo-LSSTC scheme at a PSNR of 36 dB. Alternatively, about 3 dB of PSNR video quality improvement may be observed in Figure 3.37 at a channel SNR of -5.5 dB.

For providing further insights for video scenes exhibiting different motion-activity,

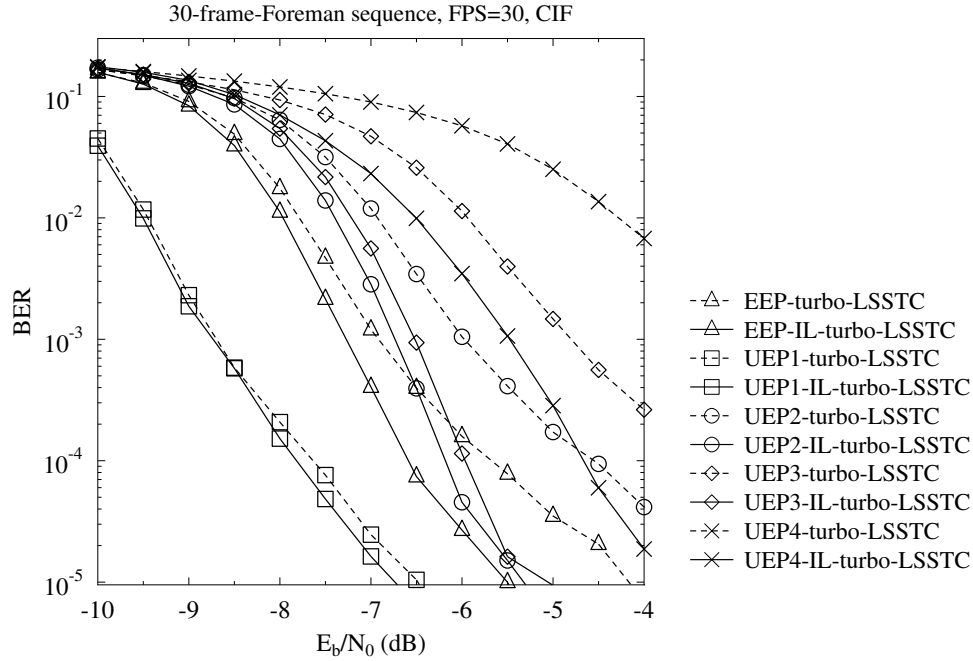


Figure 3.34: BER versus E_b/N_0 performance for the type A partition of the *Foreman* sequence using the turbo coding schemes of Table 3.10. The schematic of Figure 3.1 was used.

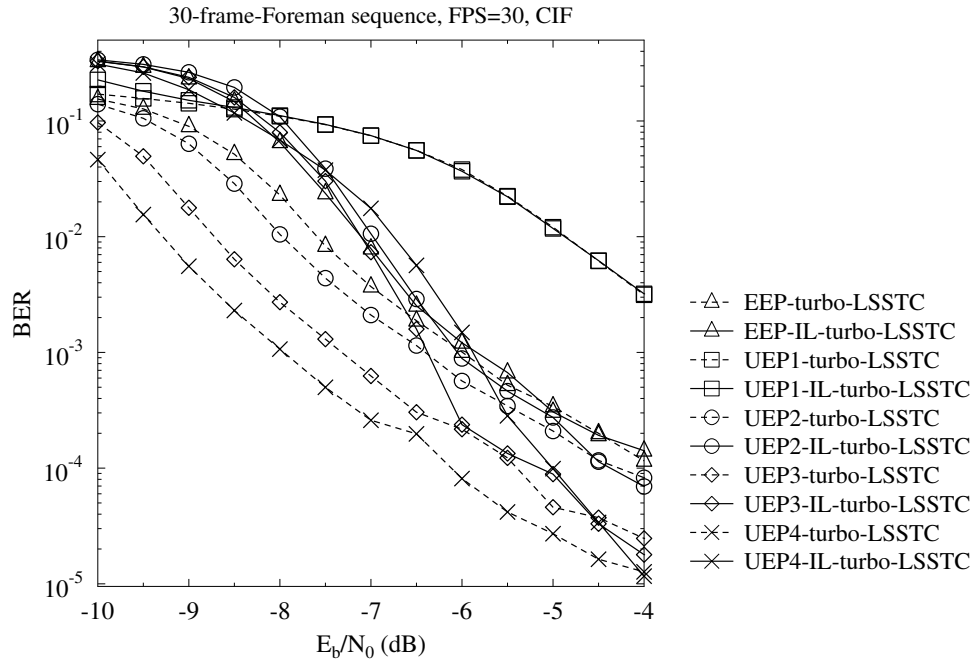


Figure 3.35: BER versus E_b/N_0 performance for the type B partition of the *Foreman* sequence using the turbo coding schemes of Table 3.10. The schematic of Figure 3.1 was used.

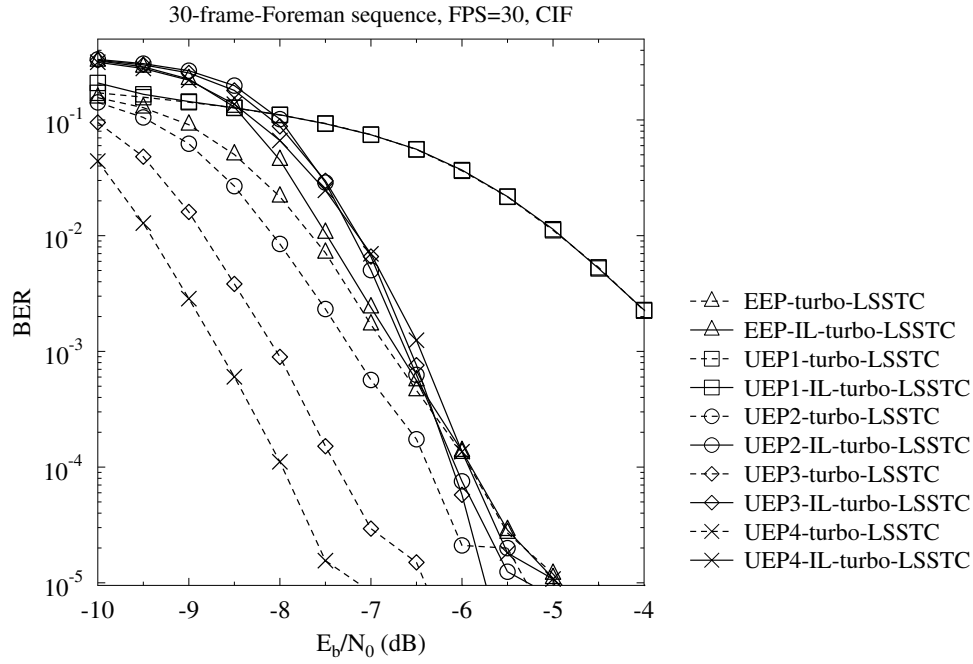


Figure 3.36: BER versus E_b/N_0 performance for the type C partition of the *Foreman* sequence using the turbo coding schemes of Table 3.10. The schematic of Figure 3.1 was used.

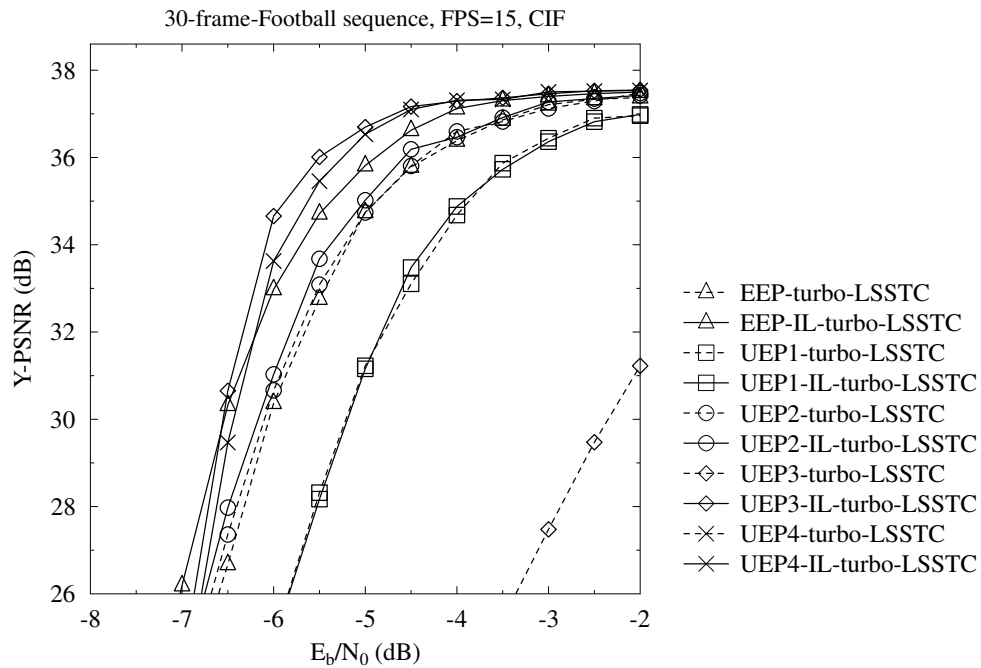


Figure 3.37: PSNR versus E_b/N_0 performance recorded for the *Football* sequence using the turbo coding schemes of Table 3.10. The schematic of Figure 3.1 was used.

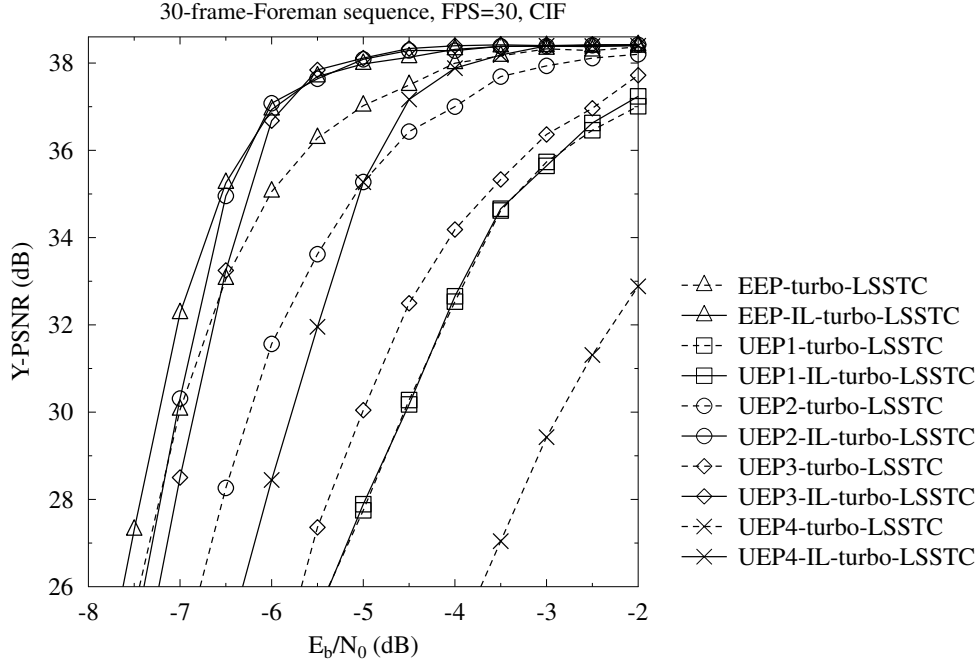


Figure 3.38: PSNR versus E_b/N_0 performance for the *Foreman* sequence using the turbo coding schemes of Table 3.10. The schematic of Figure 3.1 was used.

the PSNR versus E_b/N_0 performance of the IL-turbo-LSSTC model is presented in Figure 3.38 using the *Foreman* sequence, when using the protection arrangements of Table 3.10. Similar to the *Football* sequence, the traditional UEP technique can hardly benefit from assigning different protections to the different layers. By contrast, about 1.1 dB of power reduction is achieved by the UEP2-IL-turbo-LSSTC arrangement over the EEP-turbo-LSSTC scheme at a PSNR of 37 dB. Alternatively, about 2 dB of PSNR video quality improvement may be observed at a channel SNR of -6 dB. A subjective comparison of the EEP-IL-turbo-LSSTC and EEP-turbo-LSSTC arrangements is presented in Figure 3.39.

Based on the above discussions, we may conclude that the A partition should be allocated less FEC-redundancy for the sake of achieving the best overall system performance, when employing the turbo codec, which contradicts the traditional UEP strategy.

3.5.4 Complexity Analysis

In order to provide insights into the complexity of our scheme, we benchmark the complexity of our IL-FEC-LSSTC scheme using both the RSC codec and the turbo codec in Figure 3.40 and Figure 3.41, respectively. We emphasize that if the A partition was corrupted, the corresponding complexity imposed by the B and C partitions was not taken into account, since they cannot be utilized by the video decoder in this case. Therefore, the complexity of both the IL-FEC-LSSTC system



Figure 3.39: Video comparison at $E_b/N_0 = -6$ dB for the Football sequence and Foreman sequence. The first column indicates the original frames. The second column represents the EEP-turbo-LSSTC decoded frames. The third column indicates the EEP-IL-turbo-LSSTC decoded frames. The schematic of Figure 3.1 was used.

and of the benchmarkers is directly proportional to the E_b/N_0 value. Furthermore, in the simulations each NALU was encoded by the FEC as a single packet. The total computational complexity is dominated by that of FEC decoding. Hence, the total number of FEC decoding operations substantially affects the system's complexity, which was hence used for comparing the system's complexity. The y-axis of Figure 3.40 and Figure 3.41 represents the average number of FEC decoding operations per NALU, which was averaged over 2221 NALUs in the H.264 encoded Football bitstream for the sake of statistical relevance, where again each NALU was encoded as a single packet in the experiments.

Observe from Figure 3.40 that each curve of the IL-RSC-LSSTC schemes may be divided into two regions, where the complexity of the systems increases and decreases upon the increasing E_b/N_0 . For example, the curve of the UEP3-IL-RSC-LSSTC scheme can be split at E_b/N_0 of about -6.5 dB. Specifically, in the E_b/N_0 region of $[-10, -6.5]$ dB, the complexity of the UEP3-IL-RSC-LSSTC scheme increases upon increasing the E_b/N_0 value. This is due to the fact that the IL decoding technique was activated frequently for assisting the decoding of A partition. By contrast, for higher E_b/N_0 values the A partition is more likely to be recovered with the aid of the IL technique, which in turn results in decoding the B and C partitions more than once. In the E_b/N_0 region of $[-6.5, 5]$ dB, the complexity of the UEP3-IL-RSC-LSSTC scheme decreases upon increasing E_b/N_0 value. The reason for this

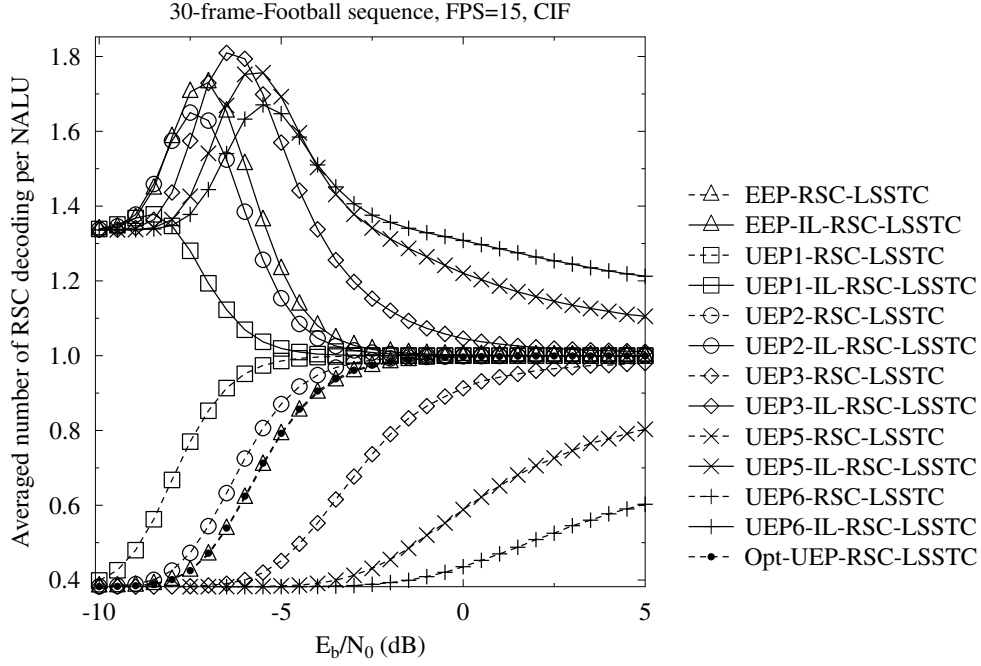


Figure 3.40: Complexity comparison of the Opt-UEP-RSC-LSSTC system, the IL-RSC-LSSTC schemes and the classic RSC-LSSTC schemes for the error protection arrangements of Table 3.10 for the *Football* sequence. The schematic of Figure 3.1 was used.

phenomenon is that the IL decoding technique is less frequently activated, when the A partition is more likely to be perfectly decoded in its own right at higher E_b/N_0 values. Moreover, the complexity of all the RSC-LSSTC schemes increases upon increasing E_b/N_0 . This may be attributed to the fact that at lower E_b/N_0 the B and C partition were more likely to be dropped by the decoder due to the corruption of the A partition. Since low E_b/N_0 results in unacceptable video quality, here we only focus on higher E_b/N_0 region. More specifically, the UEP5-IL-RSC-LSSTC scheme achieves E_b/N_0 gains of 3 dB and 2.5 dB by imposing about 21% higher complexity than the EEP-RSC-LSSTC and Opt-UEP-RSC-LSSTC schemes at a video quality of 36 dB, respectively. Alternatively, the UEP5-IL-RSC-LSSTC has PSNR gains of 3.7 dB and 3.4 dB at the cost of a 21% complexity increase compared to the EEP-RSC-LSSTC and Opt-UEP-RSC-LSSTC schemes at an E_b/N_0 of 0 dB, respectively.

Additionally, observe from Figure 3.41 that the UEP3-IL-turbo-LSSTC scheme achieves an E_b/N_0 gain of 1.1 dB by imposing about 13% higher complexity compared to the UEP2-turbo-LSSTC scheme at a video quality of 36 dB. View from a different perspective, the UEP3-IL-turbo-LSSTC scheme has a PSNR gain of 3 dB at the cost of a 13% complexity increase compared to the UEP2-turbo-LSSTC scheme at an E_b/N_0 value of -5.5 dB. Moreover, our complexity comparison of the IL-turbo-LSSTC schemes and of the classic turbo-LSSTC schemes recorded for the error protection

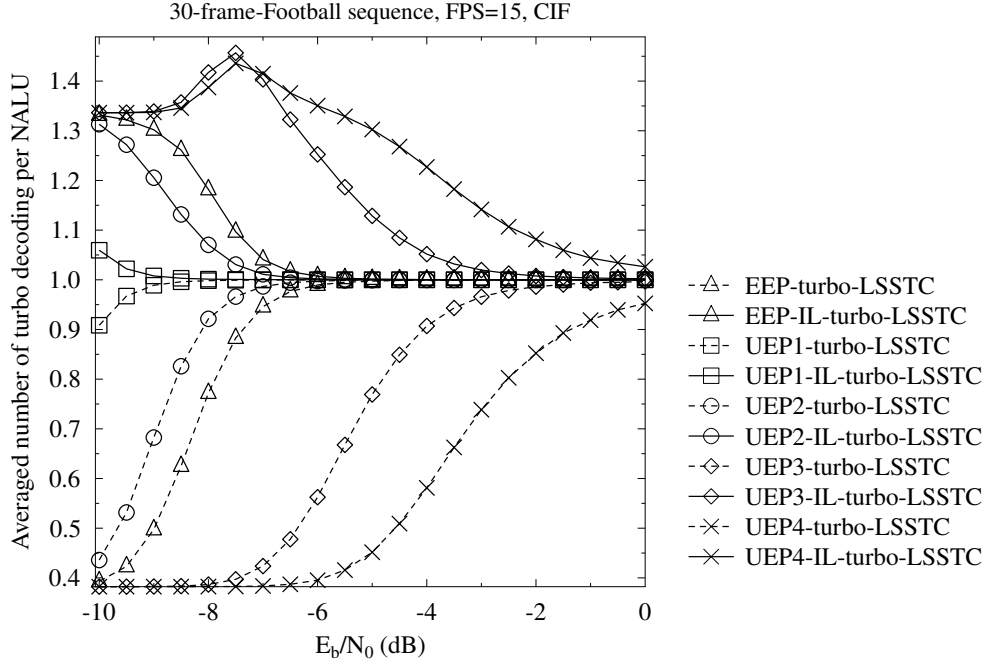


Figure 3.41: Complexity comparison of the IL-turbo-LSSTC schemes and the classic turbo-LSSTC schemes for the error protection arrangements of Table 3.10 for the *Football* sequence. The schematic of Figure 3.1 was used.

arrangements of Table 3.10 for the *Foreman* sequence is displayed in Figure 3.42, where similar trends are observed to those seen in Figure 3.41.

In conclusion of the Section 2.5:

1. In both the RSC and turbo based systems, the most significant layer should be assigned less redundancy than partitions B and C for the sake of achieving the best overall system performance, which is in contrast to the traditional UEP strategy. For example, the system arrangement having channel coding rates of 0.85, 0.44 and 0.44 for the A, B and C partitions, respectively, achieves the best system performance when employing the RSC code for the transmission of the *Football* sequence.
2. As observed in Figures 3.28, 3.29, 3.37 and 3.38 of Section 3.5.2 and Section 3.5.3, higher E_b/N_0 gains were achieved by employing a RSC codec than that of the turbo codec. Furthermore, Section 3.5.4 demonstrated that the IL-RSC-LSSTC schemes imposes only a modest complexity increase over the IL-turbo-LSSTC schemes. We may conclude that our proposed IL coding technique is capable of achieving more substantial power reductions in the lower-complexity channel coding scenarios at the cost of a 21% complexity increase.

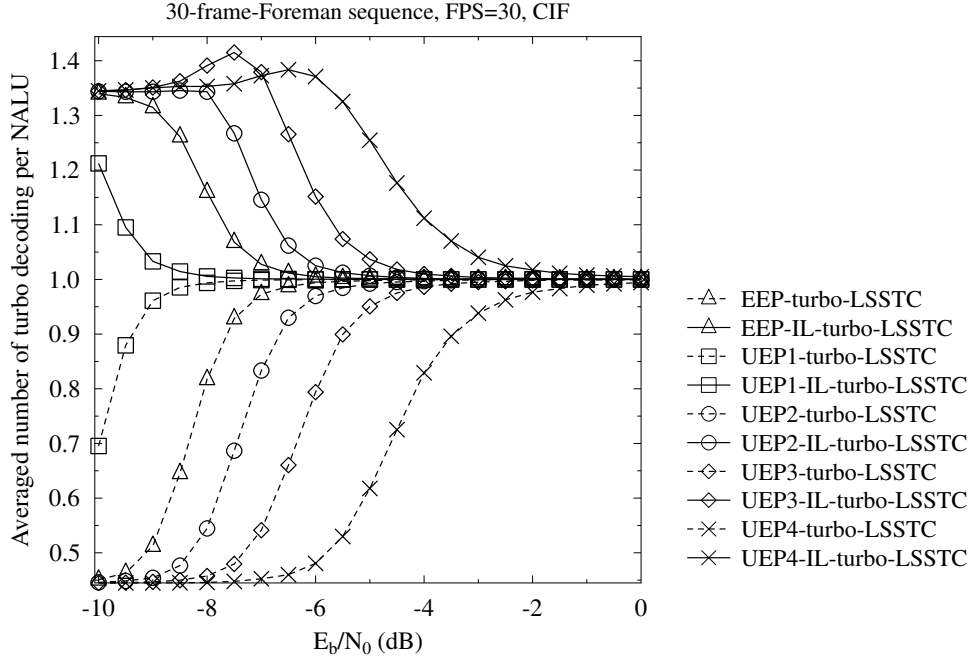


Figure 3.42: Complexity comparison of the IL-turbo-LSSTC schemes and the classic turbo-LSSTC schemes for the error protection arrangements of Table 3.10 for the *Foreman* sequence. The schematic of Figure 3.1 was used.

3.6 System Performance for Scalable Video Coding

In this section, the SECCC scheme of Figure 3.43 is incorporated into the system of Figure 3.1 in the scenarios relying on the H.264 SVC codec. We benchmark our proposed IL-SECCC-LSSTC system against the SECCC-LSSTC system employing the traditional UEP technique. The 30-frame Foreman and Football video clips represented in (176×144) -pixel QCIF and 4:2:0 YUV format were encoded by the JSVM H.264/AVC reference video codec. The Football and Foreman sequences exhibit different motion-activity, hence they allow us to demonstrate the universal nature of our proposed system. These sequences were scanned at 15 and 30 FPS, respectively. The "frame-copy" based error concealment tool built into the JSVM H.264/AVC reference codec was activated for combating the effects of channel impairments. The group of pictures (GOP) duration was set to 15, hence an Intra-coded frame (I-frame) was inserted every 15 frames. Correspondingly, both of the two video sequences were encoded into GOPs, consisting of an I-frame, followed by 14 predicted frames (P-frame). Since the bi-directionally predicted frames (B-frames) may impose error propagation on their forward- and backward-predicted dependent-frames, the B-frames are disabled in our JSVM configuration. Additionally, only the MGS feature is enabled, when encoding the video sequences into three different quality-ELs, namely layers L_0 , L_1 and L_2 using the standardized quantization parameters (QP) of 40, 32 and 24, respectively.

These configurations jointly result in a bitrate of 2297 kbps and a peak-signal to noise ratio (PSNR) of 38.8 dB for the Football sequence in the absence of transmission errors, while achieving 37.3 dB PSNR at 218 kbps for the Foreman sequence. Moreover, each SVC H.264/AVC-compressed bitstream was channel encoded and transmitted on a NALU by NALU [14] basis, which is the smallest unit to be decoded by the SVC decoder. Each NALU was protected by CRC codes. At the receiver, each decoded NALU failing to pass the CRC check process was removed before the SVC video decoding process. In all of our experiments, the compressed bitstreams were transmitted 300 times in order to generate statistically sound performance curves.

3.6.1 Self-Concatenated Covolutional Code

3.6.1.1 Encoder Structure

The encoder architecture of the SECCC codec [165] is displayed in Figure 3.43, where a RSC codec is employed. The original bits u are firstly interleaved by the interleaver π of Figure 3.43 for generating the bit-sequence u' . Then, the original bits u and the interleaved bits u' are converted to the bit sequence v by the "P/S" converter, which implies that the resultant bit sequence v contains twice the bits contained in the bit sequence u . Afterwards, the bit sequence v is encoded by a classic RSC codec for generating the parity bits v_p , as seen in Figure 3.43. The parity bits v_p and the systematic bits v are the final output of the SECCC encoder of Figure 3.43.

3.6.1.2 Decoder Structure

The decoder architecture of the SECCC codec [165] is also shown in Figure 3.43, where the LLR information $L_{ch}(v)$ and $L_{ch}(v_p)$ related to the systematic bits v and the parity bits v_p , respectively, are input to a classic RSC decoder employed as the core of the SECCC codec [165]. After each round of RSC decoding, the extrinsic information $L_e(v)$ is generated, which will be split into two parts by the "S/P" converter of Figure 3.43, namely the extrinsic information $L_e(u)$ related to the original bits u and the extrinsic information $L_e(u')$ of the interleaved bits u' . Then these two types of information will be appropriately interleaved/deinterleaved in order to generate the *a-priori* information $L_a(u')$ for the interleaved bit sequence u' and the *a-priori* information $L_a(u)$ for the original bit sequence u , which is converted by the "P/S" converter into the *a-priori* information $L_a(v)$ for the bit sequence v . Afterwards, the next decoding iteration will be performed based on the new *a-priori* information. We emphasize that the extrinsic information of the original bits u and the interleaved bits u' is exchanged in each decoding iteration, which enables the SECCC to carry out a turbo-like decoding process and hence to attain a turbo-like

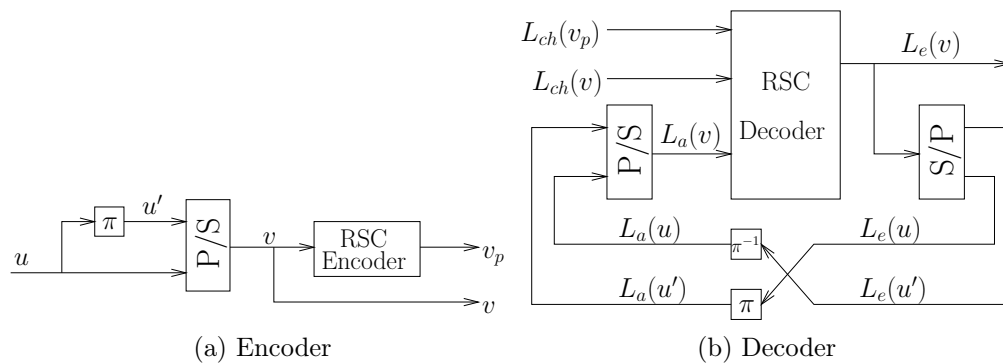


Figure 3.43: The encoding (left) and decoding (right) structure of the SECCC codec, where "P/S" and "S/P" represent a parallel to serial and serial to parallel converter, respectively.

System Parameters	Value
FEC for SECCC	RSC
Coding rate	$1/2$
Interleaver	bit-level random
Modulation	BPSK
Channel	Narrowband Rayleigh Fading Channel

Table 3.11: Parameters employed for the performance evaluation of the SECCC scheme of Figure 3.43.

error correction capability, while using the single RSC decoder shown in Figure 3.43.

3.6.1.3 Performance Study

Below we characterize the performance of a half-rate SECCC for transmission over non-dispersive uncorrelated Rayleigh fading channels using BPSK modulation. Firstly, the SECCC employing a RSC generator of [111, 101] is characterized in Figure 3.44. Observe from Figure 3.44 that the BER of the SECCC generally decreases upon increasing the number of iterations. Specifically, the SECCC using 8-iterations achieves a BER of 10^{-5} around an E_b/N_0 value of 6.5 dB. Note that the SECCC using $I = 8$ iterations outperforms the SECCCs using $I = 16$ and $I = 32$ iterations for SNR in excess of 5.5 dB, which may be attributed to the fact that the relatively short interleaver used may result in error propagation during the iterative decoding process.

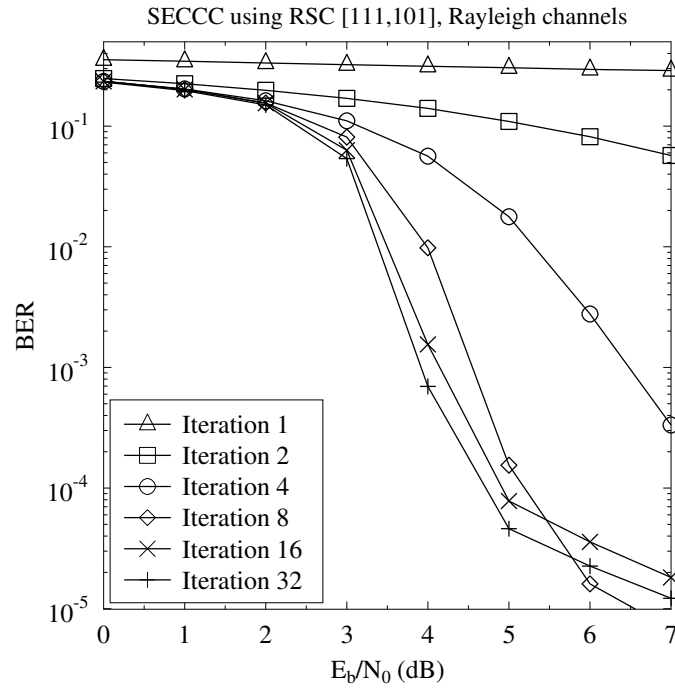


Figure 3.44: BER vs E_b/N_0 performance SECCC with diverse decoding iterations, where a RSC generator of [111, 101] and an 1000 bit interleaver are employed for BPSK modulated transmission over Rayleigh fading channels. More parameters are listed in Table 3.11. The SECCC of Figure 3.43 was used.

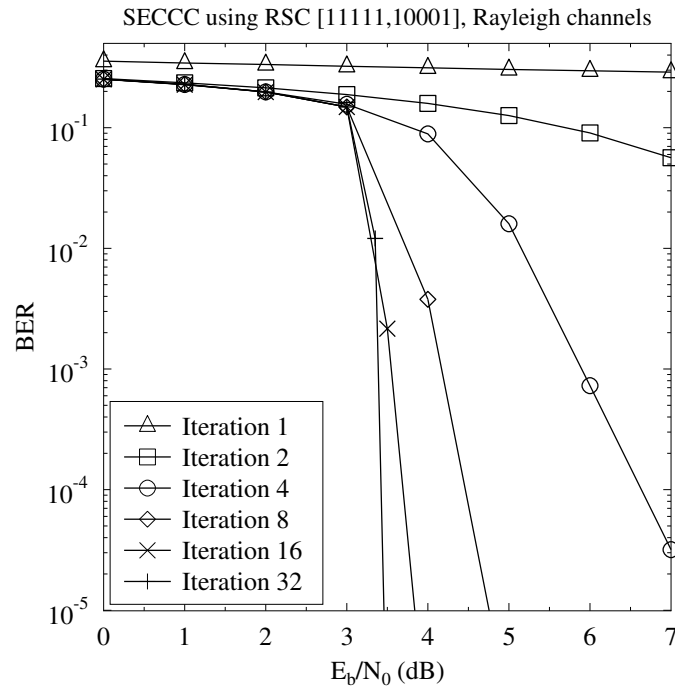


Figure 3.45: BER vs E_b/N_0 performance of SECCC with diverse decoding iterations, where an RSC generators of [11111, 10001] and 60 000 bit interleaver are employed for BPSK modulated transmission over Rayleigh fading channels. More parameters are listed in Table 3.11. The SECCC of Figure 3.43 was used.

The SECCC employing a RSC generator of $[11111, 10001]$ is characterized in Figure 3.45. Moreover, a 60 000-bit interleaver is employed for the sake of providing further insights into the attainable limits of the SECCC codec. Observe from Figure 3.45 that the BERs of the SECCC generally decrease upon increasing the number of iterations. Specifically, the SECCC using $I = 32$ iterations achieves a BER of 10^{-5} around an E_b/N_0 of 3.5 dB, where a BER cliff is observed.

3.6.2 Error Protection Arrangements

Error Protection Arrangements	Code Rates			
	Layer 0	Layer 1	Layer 2	Average
EEP	0.5/0.5	0.5/0.5	0.5/0.5	0.5/0.5
UEP1	0.64/0.64	0.54/0.55	0.46/0.47	0.5/0.5
UEP2	0.41/0.47	0.47/0.49	0.54/0.51	0.5/0.5
UEP3	0.37/0.41	0.46/0.47	0.57/0.54	0.5/0.5
UEP4	0.33/0.36	0.45/0.45	0.60/0.56	0.5/0.5

Table 3.12: Coding rates of different error protection arrangements for the *Football/Foreman* sequence used in the system of Figure 3.1.

In our simulations, an overall coding rate of 0.5 was employed for both the equal error protection (EEP) and the UEP schemes for the sake of having a fair comparison. All NALUs within each encoded bitstream were taken into account, when calculating the total number of bits of each layer.

We consider the error protection arrangements shown in Table 3.12 for the Football and Foreman sequences. All the arrangements can be applied in both the EEP or UEP schemes with the aid of variable-rate puncturers.

3.6.3 System Performance Using SECCC

In this section, we characterize the performance of our proposed IL-SECCC-LSSTC system, where a RSC codec using the generator polynomials $G = [111, 101]$ was employed for the SECCC. Moreover, the system was investigated for all the arrangements presented in Section 3.6.2.

In Figures 3.46 and 3.48, we present the BER versus E_b/N_0 curves for different layers of the encoded Football bitstream, noting that as expected for random bits, similar trends were observed for the Foreman sequence. Specifically, the BER versus

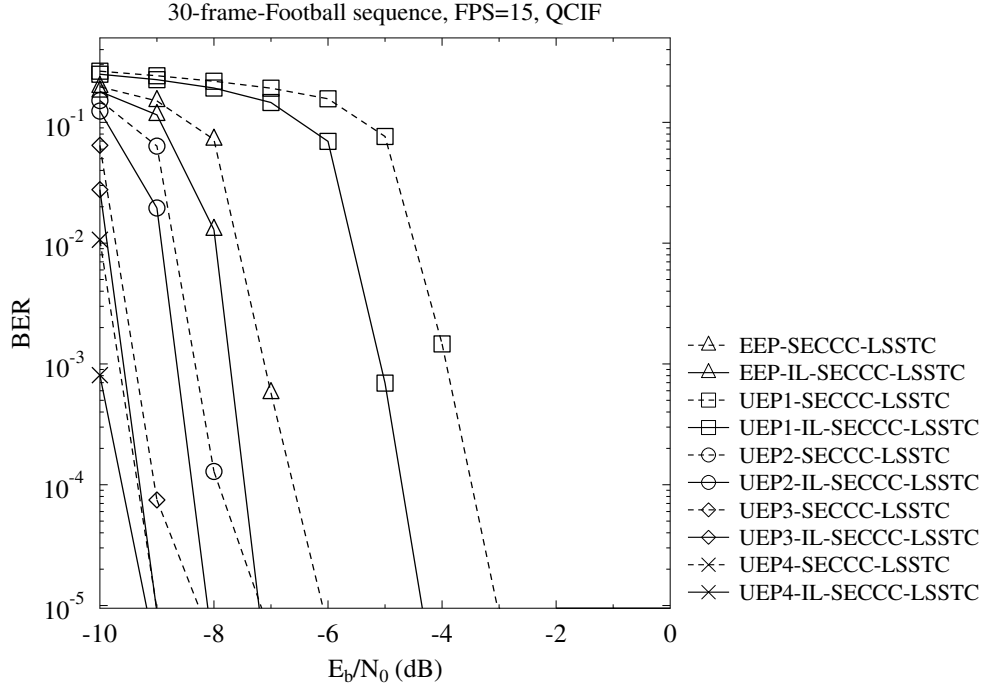


Figure 3.46: BER vs E_b/N_0 performance for L_0 -Football, where the systems used all the parameters of Table 3.12 for transmission over uncorrelated Rayleigh channels. The schematic of Figure 3.1 was used.

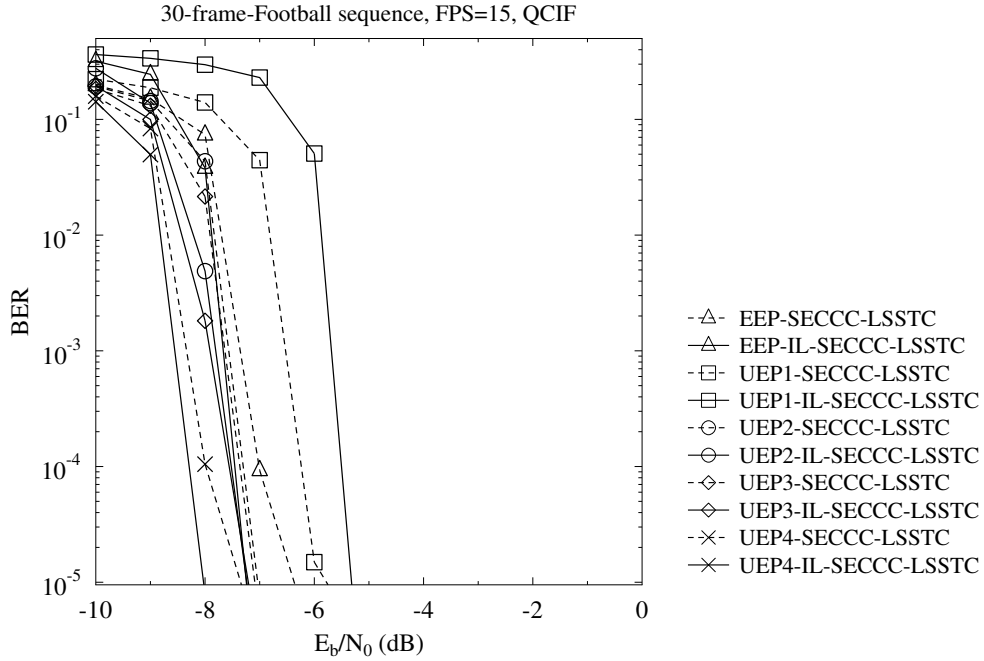


Figure 3.47: BER vs E_b/N_0 performance for L_1 -Football, where the systems used all the parameters of Table 3.12 for transmission over uncorrelated Rayleigh channels. The schematic of Figure 3.1 was used.

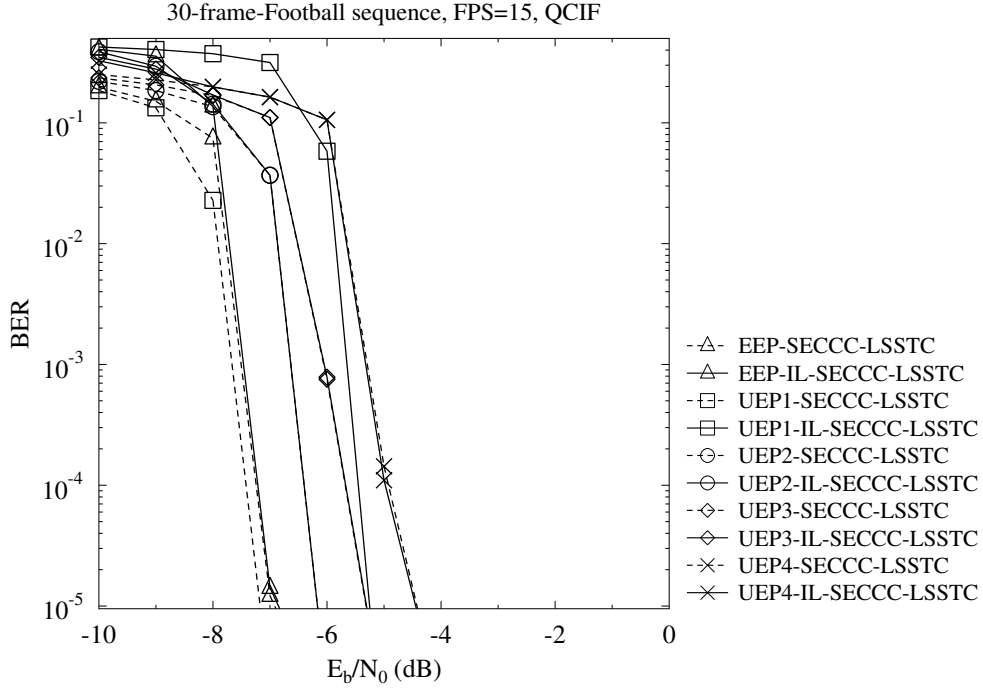


Figure 3.48: BER vs E_b/N_0 performance for L_2 -Football, where the systems used all the parameters of Table 3.12 for transmission over uncorrelated Rayleigh channels. The schematic of Figure 3.1 was used.

E_b/N_0 curves recorded for L_0 are displayed in Figure 3.46, which allow us to compare the attainable performance for all the different error protection arrangements of Table 3.12. Observe in Figure 3.46 that the IL-SECCC-LSSTC system achieved a reduced BER compared to its SECCC-LSSTC benchmarker. Specifically, the EEP-IL-SECCC-LSSTC scheme outperforms the EEP-SECCC-LSSTC benchmarker by about 1.1 dB at a BER of 10^{-5} . Moreover, among all the error protection arrangements of Table 3.12, the UEP4-IL-SECCC-LSSTC system achieves the best BER performance due to the powerful error protection assigned for layer L_0 . Therefore, we may conclude that the UEP aided IL-SECCC schemes are capable of achieving a better performance than that of the traditional UEP aided SECCC-LSSTC systems.

The BER versus E_b/N_0 curves of L_2 of the Football sequence are displayed in Figure 3.48, where the performance of the schemes using the IL technique is seen to be worse than that of the SECCC-LSSTC systems. This is due to the fact that errors may propagate to L_2 from the error-infested layer L_1 . Note that this cross-layer error propagation only occurs, when the layer L_1 is corrupted, while L_2 must be discarded in the traditional UEP aided SECCC-LSSTC schemes. Similar trends were also observed for the EL L_1 , as displayed in Figure 3.47. Moreover, the BER versus E_b/N_0 performance of the SECCC scheme is portrayed in Figures 3.49, 3.50, 3.51 for the L_0 , L_1 , L_2 layers of the Foreman sequence, respectively.

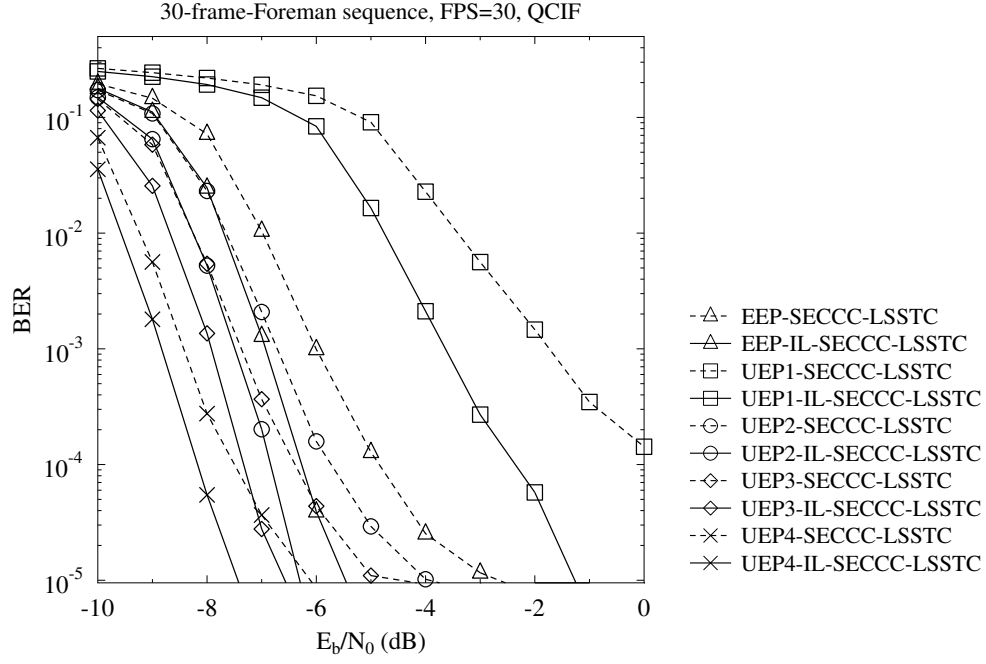


Figure 3.49: BER vs E_b/N_0 performance for L_0 -Foreman, where the systems used all the parameters of Table 3.12 for transmission over uncorrelated Rayleigh channels. The schematic of Figure 3.1 was used.

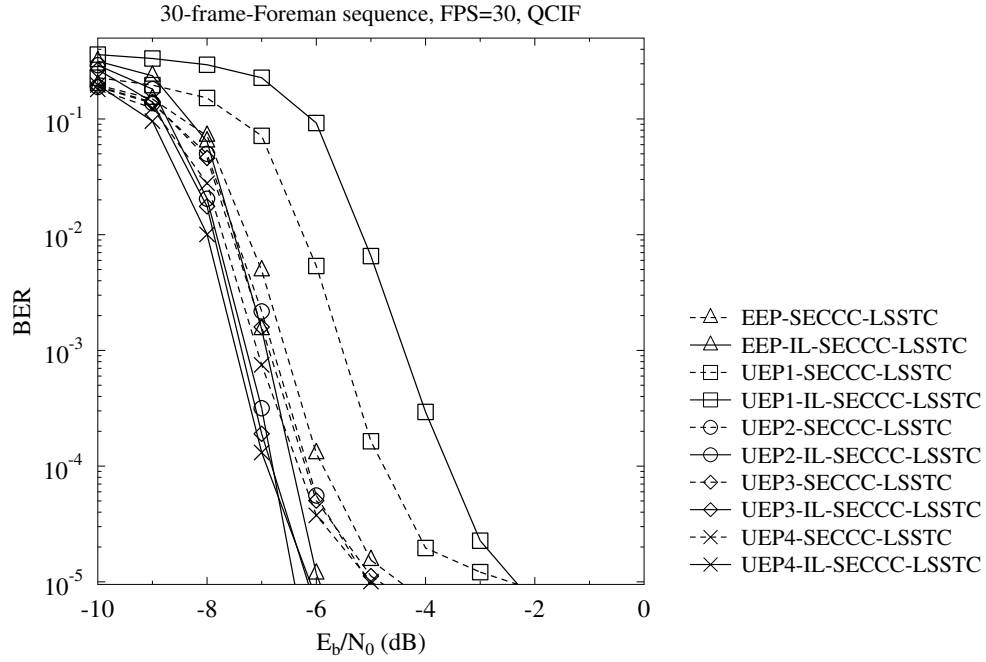


Figure 3.50: BER vs E_b/N_0 performance for L_1 -Foreman, where the systems used all the parameters of Table 3.12 for transmission over uncorrelated Rayleigh channels. The schematic of Figure 3.1 was used.

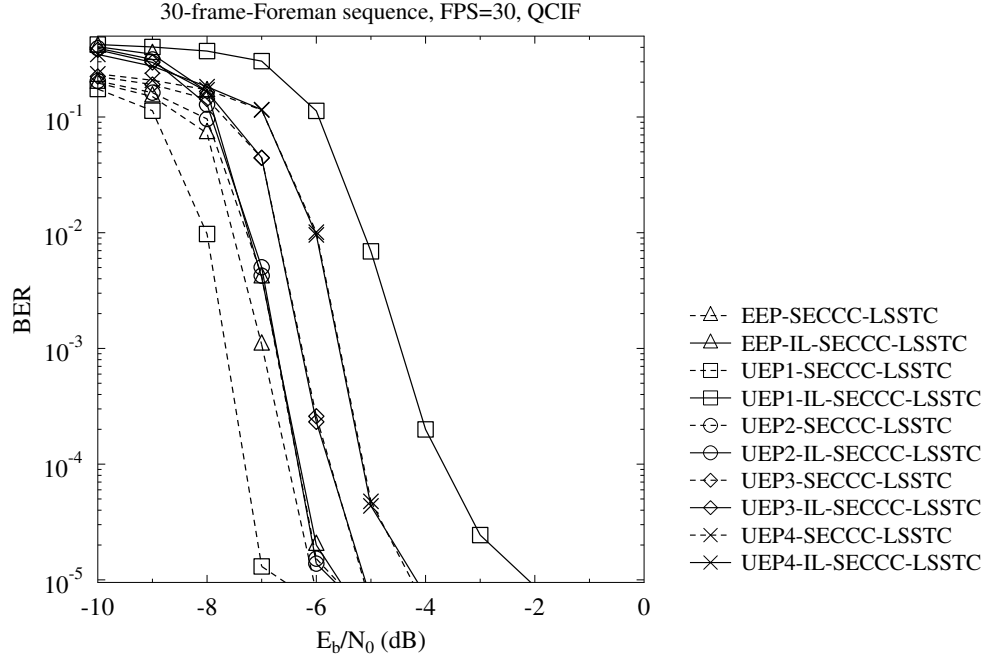


Figure 3.51: BER vs E_b/N_0 performance for L_2 -Foreman, where the systems used all the parameters of Table 3.12 for transmission over uncorrelated Rayleigh channels. The schematic of Figure 3.1 was used.

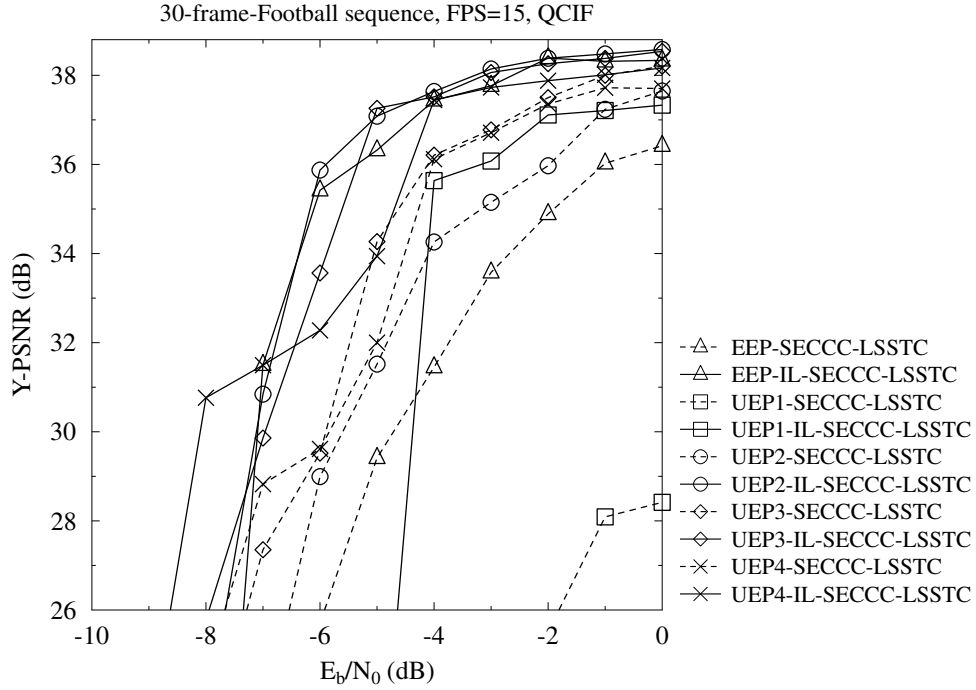


Figure 3.52: PSNR vs E_b/N_0 performance for *Football*, where the systems used all the parameters of Table 3.12 for transmission over uncorrelated Rayleigh channels. The schematic of Figure 3.1 was used.

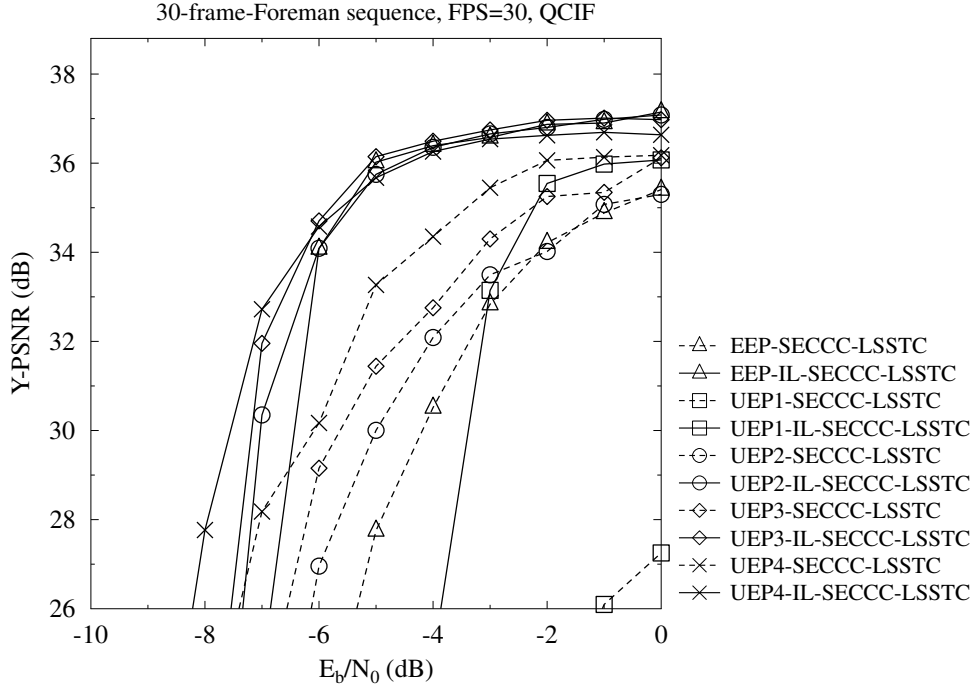


Figure 3.53: PSNR vs E_b/N_0 performance for *Foreman*, where the systems used all the parameters of Table 3.12 for transmission over uncorrelated Rayleigh channels. The schematic of Figure 3.1 was used.

The PSNR versus E_b/N_0 performance recorded for the Football sequence is displayed in Figure 3.52. Observe in Figure 3.52 that among all the traditional UEP error arrangements, the UEP3-SECCC-LSSTC scheme of Table 3.12 achieves the best performance, attaining an E_b/N_0 gain of 3.1 dB compared to the EEP-SECCC-LSSTC system of Table 3.12 at a PSNR of 36 dB. Moreover, the IL-SECCC-LSSTC systems of Table 3.12 outperform their corresponding SECCC-LSSTC benchmarks. More explicitly, the UEP2-IL-SECCC-LSSTC scheme achieves an E_b/N_0 reduction of 2.4 dB compared to the UEP3-SECCC-LSSTC scheme at a PSNR of 37 dB. Alternatively, about 2.8 dB of PSNR improvement may be achieved at an E_b/N_0 of -5 dB. Note that in the E_b/N_0 region of $[-8, -6]$ dB only a modest PSNR improvement can be observed with the increase of the E_b/N_0 value for the UEP4-IL-SECCC-LSSTC scheme in Figure 3.52. This may be attributed to the fact that a weaker error protection was assigned to the ELs, hence this may result in corrupting the ELs at these low E_b/N_0 values. A subjective comparison of the EEP-SECCC-LSSTC, UEP3-SECCC-LSSTC and UEP2-IL-SECCC-LSSTC schemes of Table 3.12 employing the Football sequence is illustrated in Figure 3.54.

For providing further insights into the system's behavior for video sequences



Figure 3.54: Comparison of frames at E_b/N_0 of -6 dB for the *Football* and *Foreman* sequences. The three columns (from left to right) indicate frames of the original video, of the traditional UEP scheme, and of the IL-UEP scheme, respectively. The schematic of Figure 3.1 was used.

exhibiting lower motion-activity, the PSNR versus E_b/N_0 performance of the IL-SECCC-LSSTC model employing the Foreman sequence is illustrated in Figure 3.53, where the error protection arrangements of Table 3.12 were tested. Similar to the Football sequence, about 3 dB of power reduction is achieved by the UEP3-IL-SECCC-LSSTC compared to the best traditional UEP4-SECCC-LSSTC scheme at a PSNR of 36 dB. Alternatively, about 2.9 dB of PSNR improvement may be observed at an E_b/N_0 of -5 dB. A subjective comparison of the EEP-SECCC-LSSTC, UEP4-SECCC-LSSTC and UEP3-IL-SECCC-LSSTC schemes employing the Foreman sequence is illustrated in Figure 3.54.

3.6.4 Complexity Analysis

Below we benchmark the complexity of our proposed IL-SECCC-LSSTC system against that of the traditional SECCC-LSSTC systems for the sake of providing insights into the complexity of our scheme. The related performance comparisons are plotted in Figure 3.55. Note that when layer i is corrupted, the complexity imposed by the decoding of the higher-order layers is not taken into account, since they will be discarded by the SVC decoder. In the simulations, each NALU was encoded by the SECCC as a stand-alone packet. Therefore, the number of SECCC

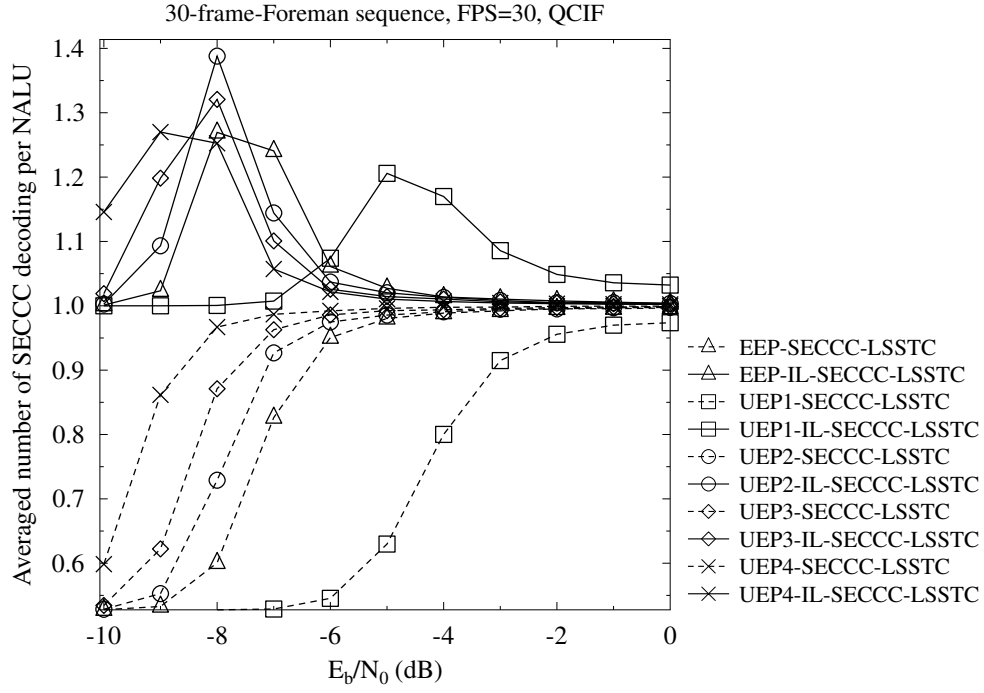


Figure 3.55: Complexity vs E_b/N_0 for *Foreman*, where the systems used all the parameters of Table 3.12 for transmission over uncorrelated Rayleigh channels. The schematic of Figure 3.1 was used.

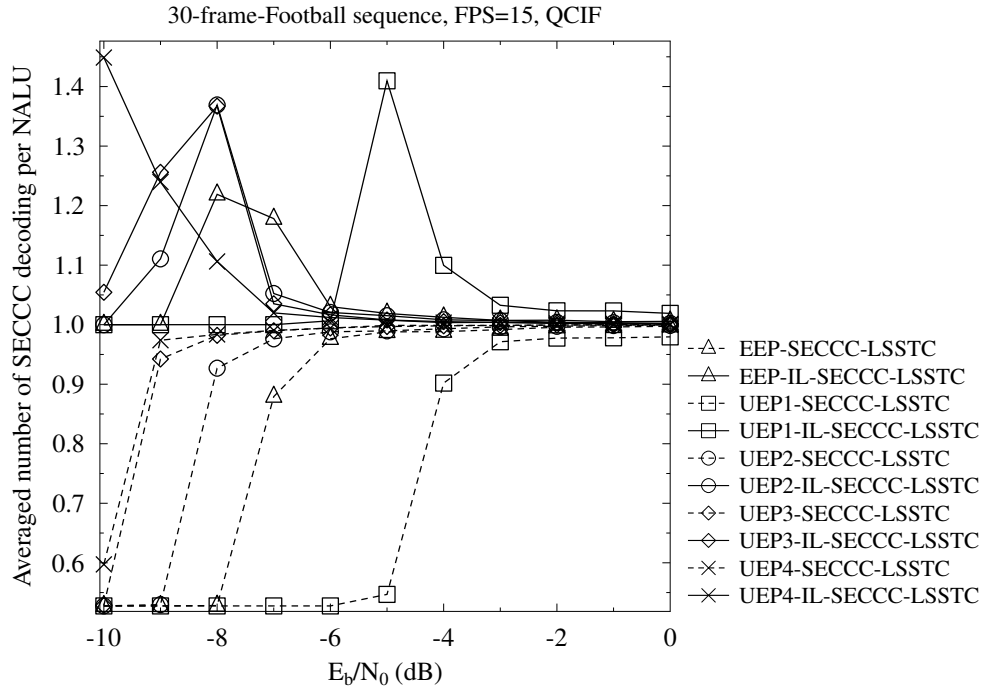


Figure 3.56: Complexity vs E_b/N_0 for the *Football* sequence, where the systems used all the parameters of Table 3.12 for transmission over uncorrelated Rayleigh channels. The schematic of Figure 3.1 was used.

decoding operations is representative of the system's complexity. Specifically, the y-axis of Figure 3.55 indicates the average number of SECCC decoder activations per NALU. There are a total of 127 NALUs within the SVC H.264/AVC encoded Foreman bitstream, where each NALU was streamed as a stand-alone packet in the simulations. Note that similar trends were observed for the Football sequence.

The complexity of all the traditional SECCC-LSSTC schemes increases upon increasing the E_b/N_0 value due to the fact that at lower E_b/N_0 values layer L_1 and layer L_2 were more likely to be discarded by the decoder, upon receiving a corrupted layer L_0 . Moreover, at E_b/N_0 values around -8 dB complexity peaks appear in Figure 3.55 for the IL-SECCC-LSSTC schemes due to the fact that the BL cannot be successfully decoded in its own right, which in turn activates the IL feedback more frequently. Here we are only interested in the relatively higher E_b/N_0 values, where achieving a sufficiently high video quality is possible. Specifically, the UEP3-IL-SECCC-LSSTC scheme achieves an E_b/N_0 gain of 3 dB by imposing about 1.5%¹⁸ higher complexity than the UEP4-SECCC-LSSTC scheme at a PSNR of 36 dB. Alternatively, the UEP3-IL-SECCC-LSSTC achieves a PSNR gain of 2.9 dB at the cost of a 1.5% complexity increase compared to the UEP4-SECCC-LSSTC of Table 3.12 at a E_b/N_0 of -5 dB. Moreover, the complexity vs E_b/N_0 curves recorded for the *Football* sequence are displayed in Figure 3.56, where similar trends are observed to those of Figure 3.55.

3.7 System Performance for Optimized Scalable Video Coding

In this section, we benchmark our proposed Optimized-IL-RSC system against an identical RSC-aided system employing the traditional UEP technique. The generator polynomials of $[1011, 1101, 1111]$ are employed for configuring the RSC codec. Moreover, BPSK modulated signals were transmitted through non-dispersive uncorrelated Rayleigh fading channels. These parameters are listed in Table 3.13. The 30-frame Foreman and Football video clips represented in (176×144) -pixel quarter common intermediate format (QCIF) and 4:2:0 YUV format were encoded by the JSVM H.264/AVC reference video codec. The Football and Foreman sequences exhibit different motion-activity, hence they allow us to demonstrate the universal nature of our proposed system. These sequences were scanned at 15 and 30 FPS, respectively. The "frame-copy" based error concealment tool built into the JSVM

¹⁸The complexity imposed is read from $E_b/N_0 = -5$ dB and -2 dB for the UEP3-IL-SECCC-LSSTC and UEP4-SECCC-LSSTC, respectively.

System Parameters	Value
FEC	RSC[1011, 1101, 1111]
Overall Coding rate	1/2
Modulation	BPSK
Channel	Uncorrelated Rayleigh Fading Channel

Table 3.13: The parameters of the system of Figure 3.16.

H.264/AVC reference codec was activated for combating the effects of channel impairments. The GOP duration was set to 15, hence an Intra-coded frame (I-frame) was inserted every 15 frames. Correspondingly, both of the two video sequences were encoded into GOPs, consisting of an I-frame, followed by 14 predicted frames (P-frame). Since the bi-directionally predicted B-frames may impose error propagation on their forward- and backward-predicted dependent-frames, the B-frames are disabled in our JSVM configuration. Additionally, only the Medium Grain Scalability (MGS) [15, 43] feature is enabled, when encoding the video sequences into three different-quality ELs, namely into the layers L_0 , L_1 and L_2 using the standardized quantization parameters (QP) of 40, 32 and 24, respectively. These configurations jointly result in a bitrate of 2297 kbps and a PSNR of 38.8 dB for the Football sequence in the absence of transmission errors, while achieving 37.3 dB PSNR at 218 kbps for the Foreman sequence. These parameters of the video sequences are listed in Table 3.14.

Moreover, each SVC H.264/AVC-compressed bitstream was channel encoded and transmitted on a network abstract layer unit (NALU) by NALU [14] basis, which is the smallest unit to be decoded by the SVC decoder. Each NALU was protected by CRC codes. At the receiver, each decoded NALU failing to pass the CRC check process was removed before the SVC video decoding process. In all of our experiments, the compressed bitstreams were transmitted 300 times in order to generate statistically sound performance curves.

3.7.1 Off-line LUTs Generation

In our experiments, the vectors of $[0 : 0.5 : 15]$, $[0 : 0.01 : 1]$, $[0.33 : 0.02 : 1]$ ¹⁹ are utilized for the variables SNR , I_s , r , respectively, for generating the LUTs, which result

¹⁹These values can be stored as floats in 8 bytes each.

	Football	Foreman
Representation	YUV 4:2:0	YUV 4:2:0
Format	QCIF	QCIF
Bits Per Pixel	8	8
FPS	15	30
Number of Frames	30	30
Video Codec	SVC-H.264	SVC-H.264
Bitrate	2297 kbps	218 kbps
Error-Free PSNR	38.8 dB	37.3 dB
Error Concealment	Frame-Copy	Frame-Copy

Table 3.14: The parameters of the video sequences employed in the evaluation of the system of Figure 3.16.

in $n_{snr} = 31$, $n_I = 101$, $n_r = 33$. Firstly, we recorded the MI values of the LLRs output by the BPSK demodulator for the SNR values of $[0 : 0.5 : 15]$ for the sake of generating the LUT $T_b(SNR)$. Then for each SNR , the MI values $[0 : 0.01 : 1]$ for I_s and the coding rates $[0.33 : 0.02 : 1]$ for r were utilized for generating the LUTs $T_e(SNR, I_s, r)$ and $T_p(SNR, I_s, r)$, simultaneously. Furthermore, 8-byte floating values were utilized for storing the LUTs in memory. Correspondingly, the LUTs T_b , T_e and T_p require memory sizes of about 248 bytes, 800 KB and 800 KB, respectively. Finally, $n_r = 33$ results in a full search complexity of $(C \times n_r^2 = C \cdot 1089)$, thereby finding the optimized coding rates of r_0 , r_1 and r_2 . A number of entries of the LUTs generated for our system are displayed in Table 3.15.

3.7.2 System Performance

In this section, we benchmark our system against the traditional equal error protection (EEP) system, which is referred to as EEP-RSC. Furthermore, the traditional optimal UEP-RSC (Opt-UEP-RSC) system is also presented, which is the system of [56] applied in the scenario of an RSC coded system, where all the layers are unequally protected by an RSC code dispensing with the IL technique. Moreover, the IL aided EEP system (EEP-IL-RSC) is also considered.

SNR	I_s	SNR	I_a	r	I_e	$p(l)$
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
1	0.45	3	0.67	0.87	0.81	0.93
2	0.50	3	0.67	0.88	0.83	0.89
3	0.56	3	0.67	0.89	0.85	0.84
4	0.62	3	0.67	0.90	0.87	0.77
5	0.67	3	0.67	0.91	0.89	0.69
6	0.72	3	0.67	0.92	0.91	0.57
7	0.76	3	0.67	0.93	0.93	0.46
8	0.80	3	0.67	0.94	0.94	0.33
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots

(a) $T_b(SNR)$ (b) $T_e(SNR, I_s, r)$ and $T_p(SNR, I_s, r)$

Table 3.15: Example of the LUTs used by the system of Figure 3.16.

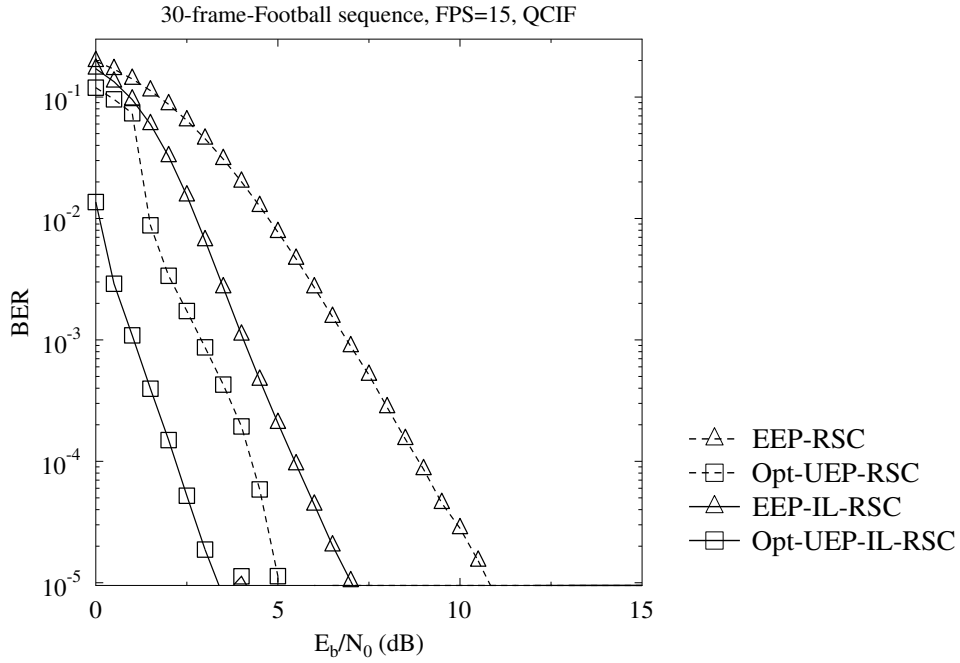


Figure 3.57: BER vs E_b/N_0 for L_0 -Football, comparison of the proposed system and the benchmarks, namely the EEP-RSC scheme, the Opt-UEP-RSC scheme [56], the EEP-IL-RSC scheme and the Opt-UEP-IL-RSC scheme. The schematic of Figure 3.16 configured with the parameters in Tables 3.13 and 3.14 was used.

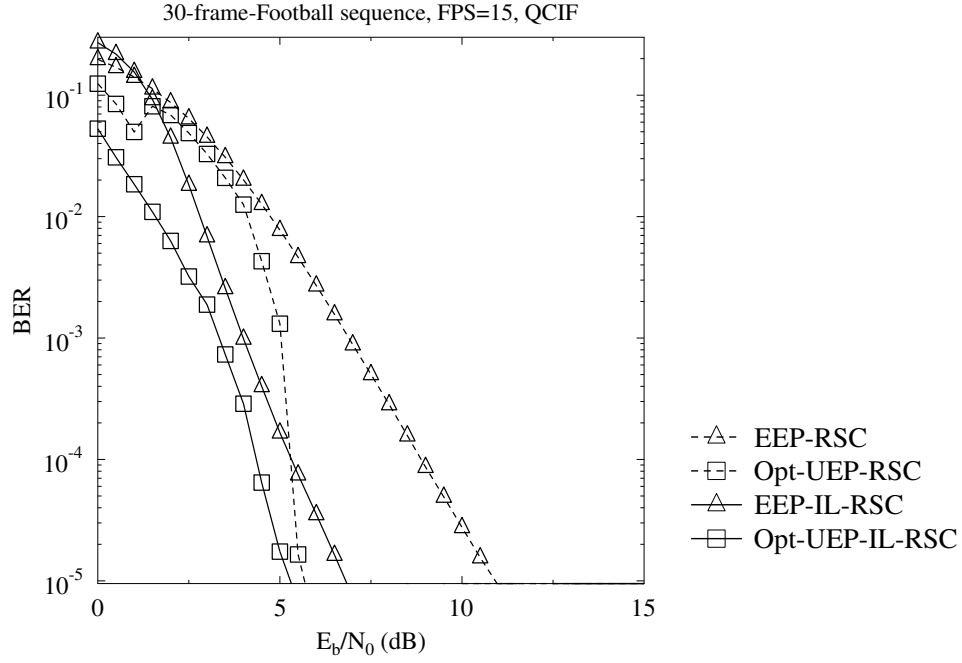


Figure 3.58: BER vs E_b/N_0 for L_1 -Football, comparison of the proposed system and the benchmarks, namely the EEP-RSC scheme, the Opt-UEP-RSC scheme [56], the EEP-IL-RSC scheme and the Opt-UEP-IL-RSC scheme. The schematic of Figure 3.16 configured with the parameters in Tables 3.13 and 3.14 was used.

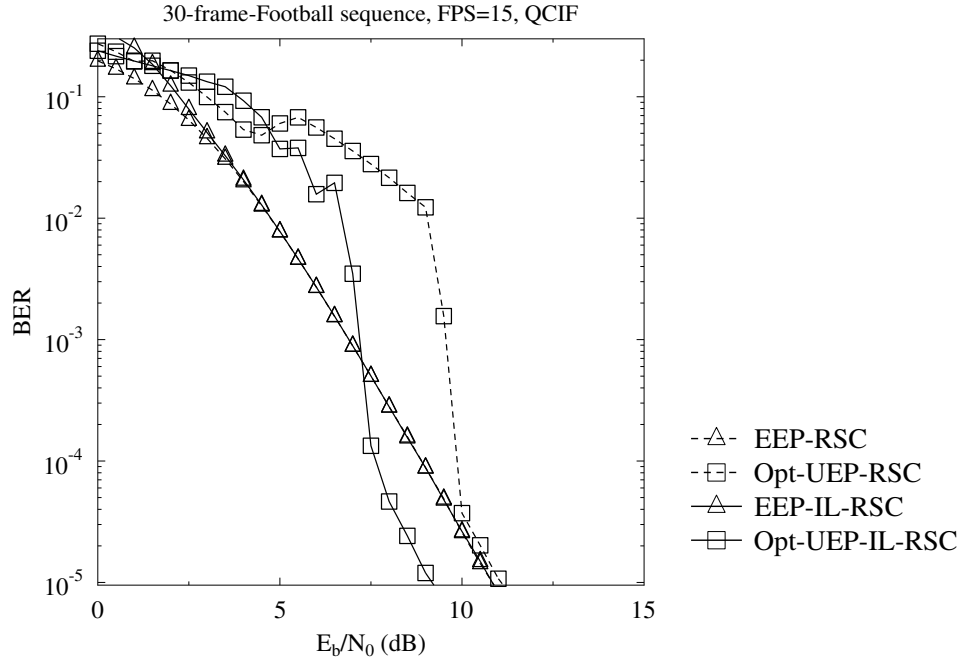


Figure 3.59: BER vs E_b/N_0 for L_2 -Football, comparison of the proposed system and the benchmarks, namely the EEP-RSC scheme, the Opt-UEP-RSC scheme [56], the EEP-IL-RSC scheme and the Opt-UEP-IL-RSC scheme. The schematic of Figure 3.16 configured with the parameters in Tables 3.13 and 3.14 was used.

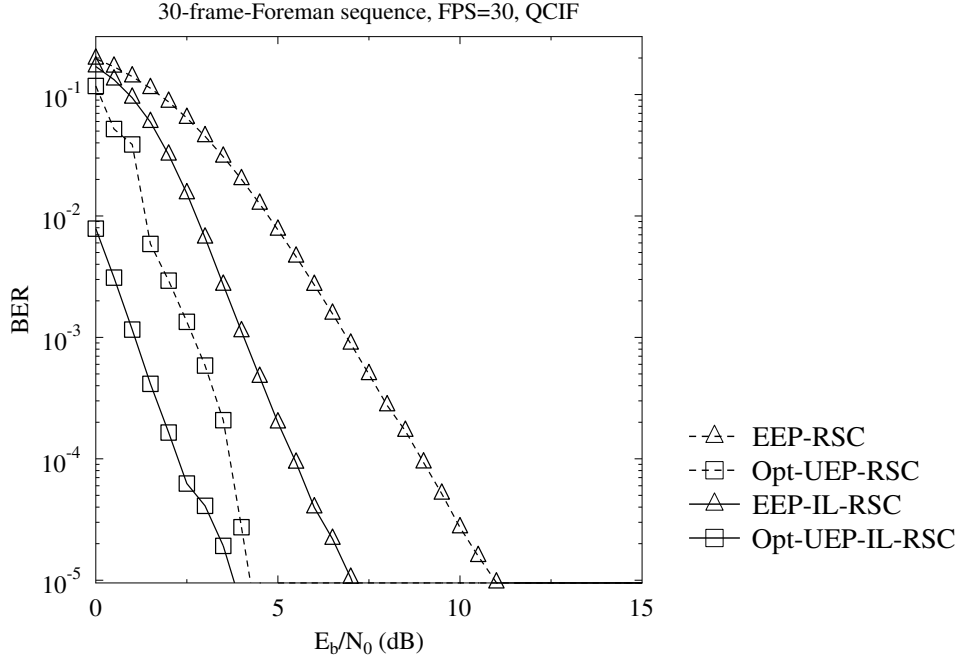


Figure 3.60: BER vs E_b/N_0 for L_0 -Foreman, comparison of the proposed system and the benchmarks, namely the EEP-RSC scheme, the Opt-UEP-RSC scheme [56], the EEP-IL-RSC scheme and the Opt-UEP-IL-RSC scheme. The schematic of Figure 3.16 configured with the parameters in Tables 3.13 and 3.14 was used.

The BER curves of the L_0 , L_1 and L_2 layers of the Football sequence are displayed in Figures 3.57, 3.58 and 3.59, respectively, where the Opt-UEP-IL-RSC system always outperforms the Opt-UEP-RSC system. Moreover, as seen in Figure 3.59, the EEP schemes outperform the Opt-UEP-IL-RSC and Opt-UEP-RSC schemes in the lower E_b/N_0 range. This is due to the fact that the coding-rate of layer L_2 is sacrificed for the sake of protecting the L_0 and L_1 layers. Similar trends were observed for the Foreman sequence, which are displayed in Figures 3.60, 3.61, 3.62 for layers L_0 , L_1 , L_2 , respectively.

The PSNR versus E_b/N_0 performance recorded for the Football sequence is displayed in Figure 3.63, where we observe that the EEP-IL-RSC scheme substantially outperforms the EEP-RSC system, while it only slightly outperforms the Opt-UEP-RSC scheme. However, by optimizing the coding rates, the Opt-UEP-IL-RSC scheme becomes capable of substantially outperforming the Opt-UEP-RSC system. In the low E_b/N_0 range, the Opt-UEP-IL-RSC and Opt-UEP-RSC schemes have a similar PSNR performance and tend to exhibit a residual error floor. This is due to the fact that these two systems endeavor to protect the BL L_0 by sacrificing the protection of

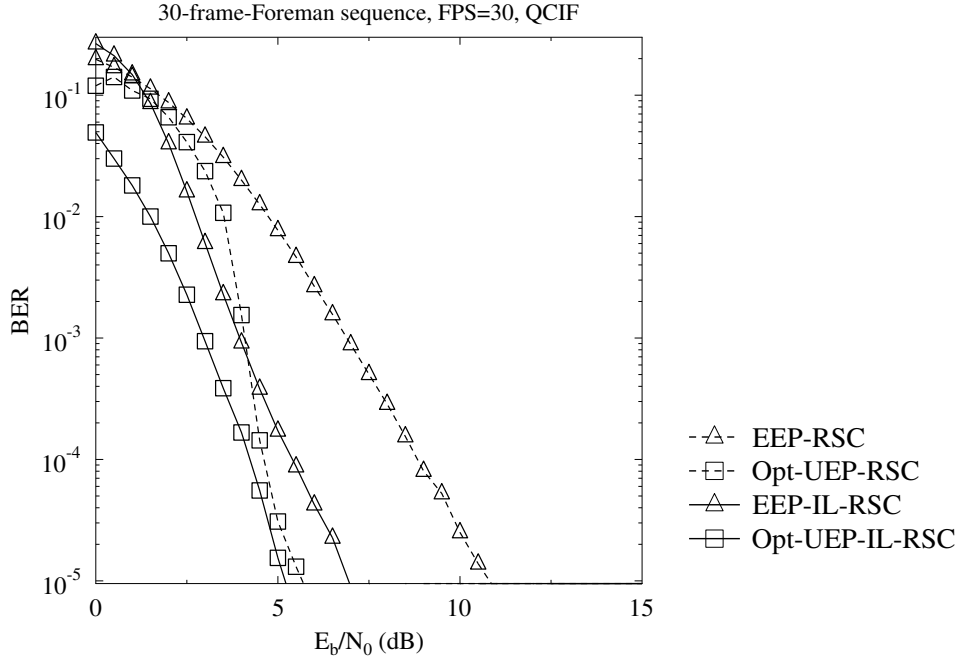


Figure 3.61: BER vs E_b/N_0 for L_1 -Foreman, comparison of the proposed system and the benchmarks, namely the EEP-RSC scheme, the Opt-UEP-RSC scheme [56], the EEP-IL-RSC scheme and the Opt-UEP-IL-RSC scheme. The schematic of Figure 3.16 configured with the parameters in Tables 3.13 and 3.14 was used.

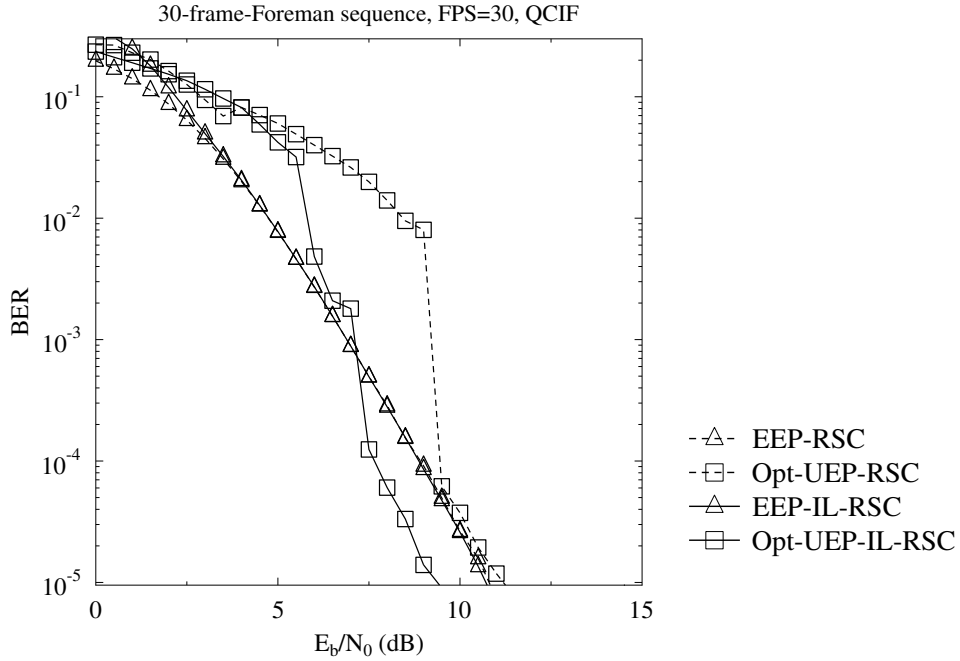


Figure 3.62: BER vs E_b/N_0 for L_2 -Foreman, comparison of the proposed system and the benchmarks, namely the EEP-RSC scheme, the Opt-UEP-RSC scheme [56], the EEP-IL-RSC scheme and the Opt-UEP-IL-RSC scheme. The schematic of Figure 3.16 configured with the parameters in Tables 3.13 and 3.14 was used.

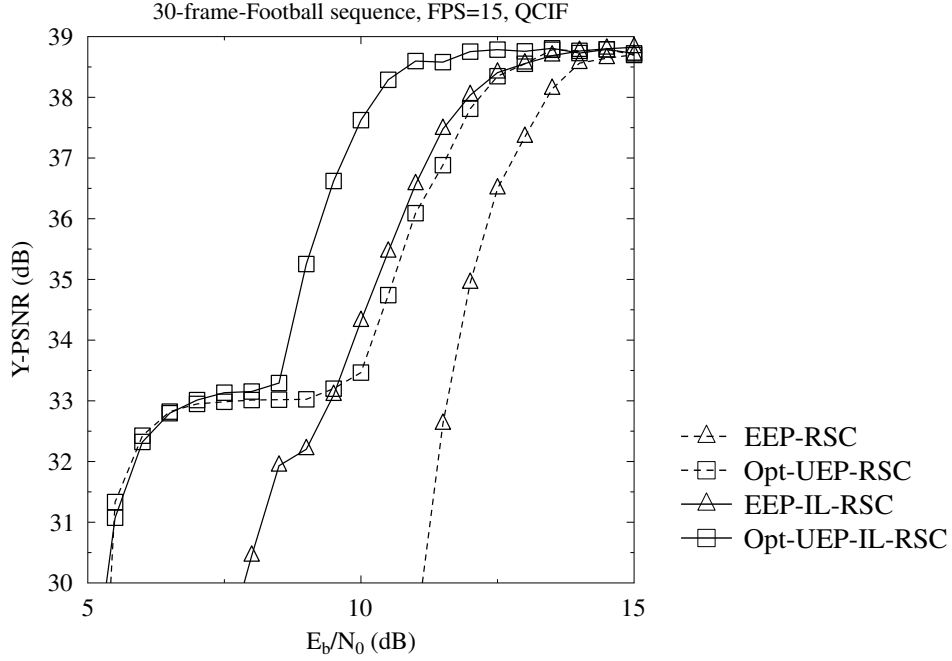


Figure 3.63: PSNR vs E_b/N_0 for *Football*, comparison of the proposed system and the benchmarks, namely the EEP-RSC scheme, the Opt-UEP-RSC scheme [56], the EEP-IL-RSC scheme and the Opt-UEP-IL-RSC scheme for the *Football* and *Foreman* sequences. The schematic of Figure 3.16 configured with the parameters in Tables 3.13 and 3.14 was used.

the ELs, where the ELs are unlikely to be recovered before reaching $E_b/N_0 = 10$ dB. This error floor becomes explicit in the BER curves of L_2 displayed in Figure 3.59. On the other hand, the Opt-UEP-RSC scheme outperforms the EEP-RSC scheme by about $E_b/N_0 = 1.3$ dB at a PSNR of 38 dB. The Opt-UEP-IL-RSC scheme achieves an E_b/N_0 reduction of about 1.9 dB compared to the Opt-UEP-RSC scheme at a PSNR of 38 dB. Alternatively, about 3.3 dB of PSNR video quality improvement is observed at an E_b/N_0 of 10 dB. A subjective comparison of the benchmarks recorded for the Football sequence is presented in Figure 3.65.

For providing further insights for video scenes having different motion-activity, the PSNR versus E_b/N_0 performance of the Opt-UEP-IL-RSC is portrayed in Figure 3.64 using the Foreman sequence. The Opt-UEP-RSC scheme outperforms the EEP-RSC system by about $E_b/N_0 = 0.8$ dB at a PSNR of 37 dB. Moreover, about 1.7 dB of power reduction is achieved by the Opt-UEP-IL-RSC scheme compared to the Opt-UEP-RSC scheme at a PSNR of 37 dB. Alternatively, about 1.6 dB of PSNR video quality improvement may be observed at an E_b/N_0 of 10 dB.

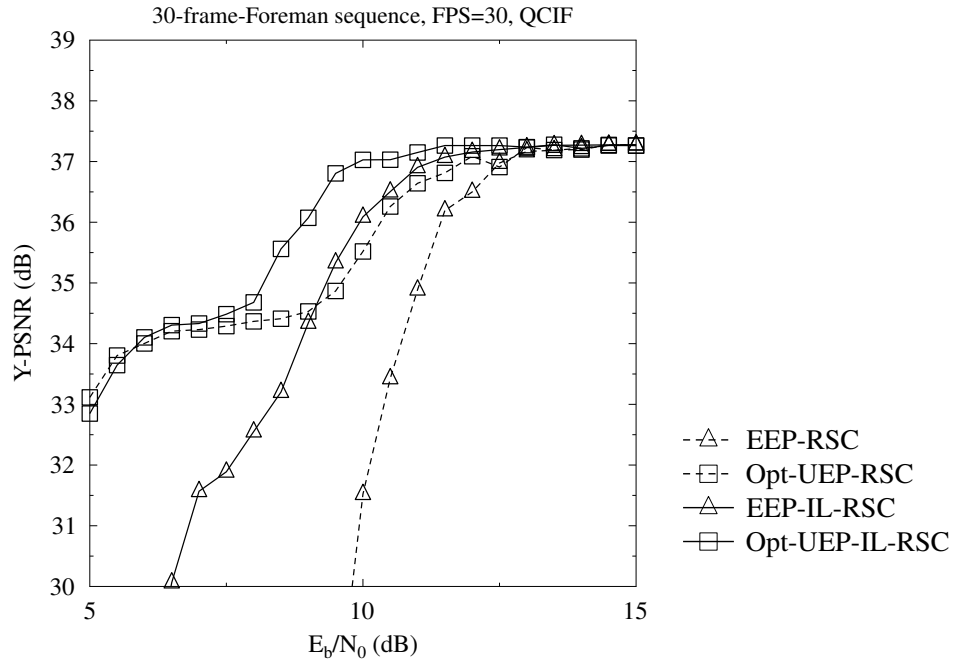


Figure 3.64: PSNR vs E_b/N_0 for *Foreman*, comparison of the proposed system and the benchmarkers, namely the EEP-RSC scheme, the Opt-UEP-RSC scheme [56], the EEP-IL-RSC scheme and the Opt-UEP-IL-RSC scheme for the *Football* and *Foreman* sequences. The schematic of Figure 3.16 configured with the parameters in Tables 3.13 and 3.14 was used.

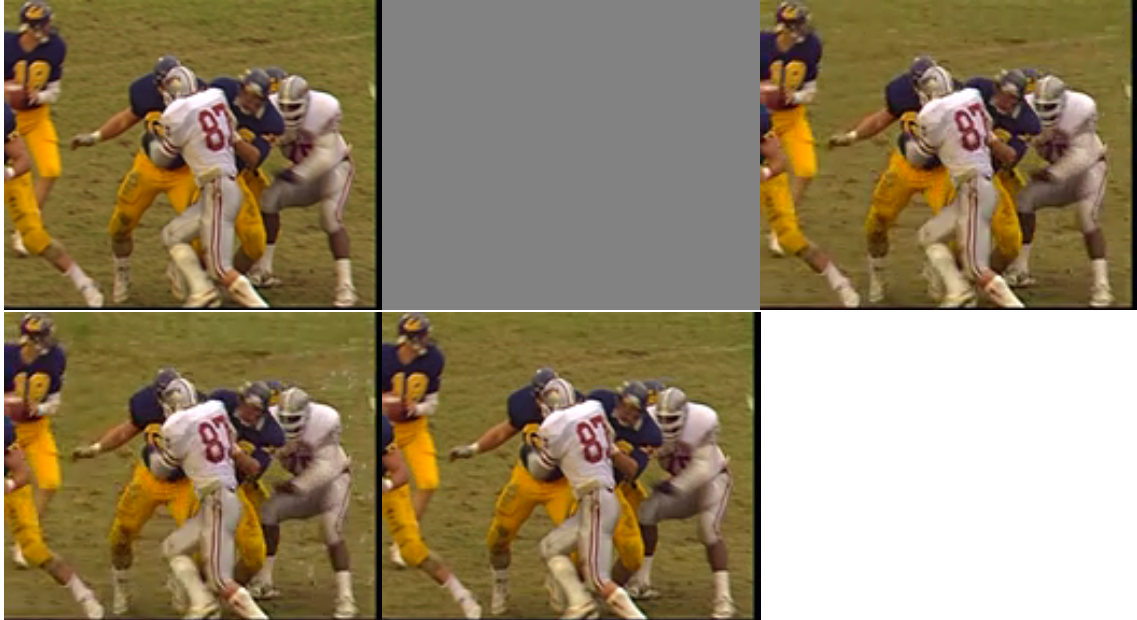


Figure 3.65: Comparison of frames at E_b/N_0 of 10 dB for the *Football* sequences. The five columns (from left to right, from top to bottom) indicate frames of the original video, the EEP-RSC scheme, the Opt-UEP-RSC [56] scheme, the EEP-IL-RSC scheme and the Opt-UEP-IL-RSC scheme, respectively. The schematic of Figure 3.16 configured with the parameters in Tables 3.13 and 3.14 was used.

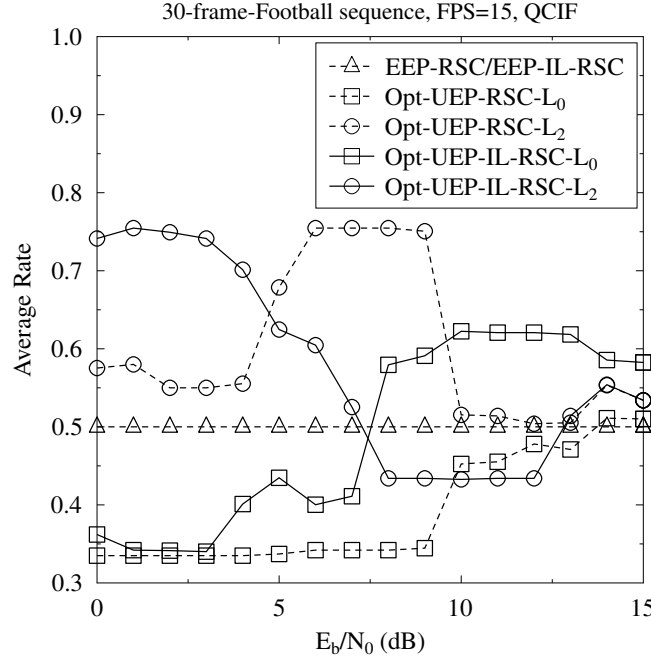


Figure 3.66: Coding rates comparison of the benchmarks, namely the EEP-RSC scheme, the Opt-UEP-RSC [56] scheme, the EEP-IL-RSC scheme and the Opt-UEP-IL-RSC scheme, for the *Football* sequence. The schematic of Figure 3.16 configured with the parameters in Tables 3.13 and 3.14 was used.

3.7.3 Optimized Coding Rates

The optimized coding rates found by our proposed regime for the *Football* sequence are shown in Figure 3.66 in comparison to the benchmarks. Specifically, the y axis of Figure 3.66 indicates the averaged coding rates of all frames. Furthermore, the coding rates of L_0 and L_2 are presented, while the curves of L_1 are similar to those of the BL L_0 . Observe in Figure 3.66 that in the low E_b/N_0 range the coding rates of the BL L_0 of the Opt-UEP-RSC and Opt-UEP-IL-RSC schemes are lower than the overall coding rate of 0.5. The reason of this observation is that the protection of the layers L_1 and L_2 is sacrificed for the sake of protecting the more important BL L_0 . At high E_b/N_0 values, the coding rates of the BL L_0 are increased due to the fact that L_0 is more likely to be corrected at high coding rates, when favorable channel conditions prevail. In comparison to the BL L_0 , the coding rates of L_2 are reduced upon increasing E_b/N_0 . Moreover, similar trends are observed for the *Foreman* sequence, which are displayed in Figure 3.67.

3.8 Summary and Conclusions

In this chapter, an IL-FEC coded video scheme relying on multi-functional MIMOs was proposed for mobile TV broadcasting, where the data partitioning mode and SVC of H.264 video coding was utilized and the systematic bits of the A partition

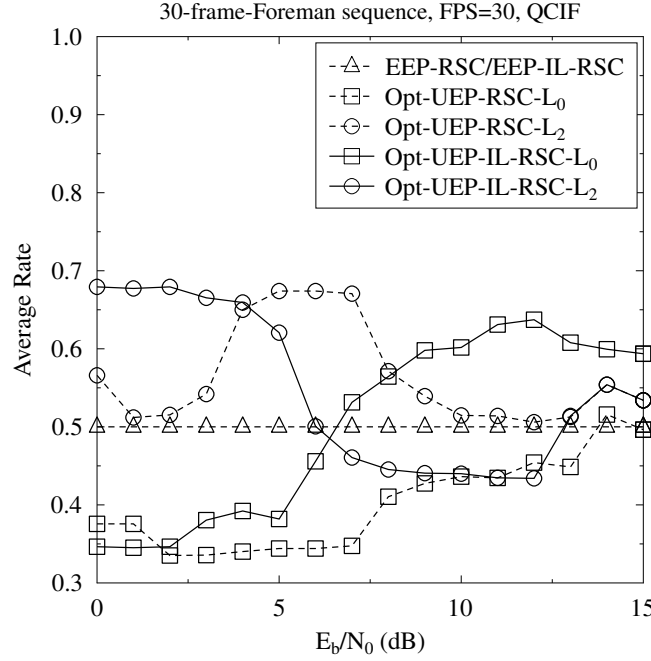


Figure 3.67: Coding rates comparison of the benchmarks, namely the EEP-RSC scheme, the Opt-UEP-RSC [56] scheme, the EEP-IL-RSC scheme and the Opt-UEP-IL-RSC scheme, for the *Foreman* sequence. The schematic of Figure 3.16 configured with the parameters in Tables 3.13 and 3.14 was used.

were incorporated into the systematic bits of the B and C partitions using an XOR operation. At the receiver, our IL-FEC decoding technique of Figure 3.1 was activated for the sake of attaining an improved system performance. The RSC codec, the turbo codec and the SECCC codec of Figure 3.43 were invoked for demonstrating that the proposed scheme is capable of substantially outperforming the traditional UEP FEC codecs. The system advocated was analyzed using mutual information for providing insights into the gain attained using our IL-FEC coding scheme.

The PSNR performance of the IL-RSC-LSSTC system was portrayed benchmarked in Figure 3.28 using the Football sequence, where the FEC arrangements of Table 3.10 are employed. The IL-RSC-LSSTC system of Figure 3.1 outperforms the traditional UEP aided system by about 2.5 dB of E_b/N_0 or 3.4 dB of PSNR at the cost of 21% complexity increase. More specifically, the E_b/N_0 gains are listed in Table 3.16 for the Football and Foreman sequences with respect to the EEP-RSC-LSSTC at a PSNR of 36 dB.

The PSNR performance of the IL-Turbo-LSSTC system is characterized in Figure 3.37 using the Football sequence, where the FEC arrangements of Table 3.10 are employed. The IL-Turbo-LSSTC system of Figure 3.1 outperforms the traditional UEP aided system by about 1.1 dB of E_b/N_0 or 3 dB of PSNR at the cost of 13% complexity increase. More specifically, the E_b/N_0 gains are listed in Table 3.17 for

(dB)	Football	Foreman	(dB)	Football	Foreman
EEP-IL	1	0.79	UEP3-IL	2	1.5
Opt-UEP	0.53	0.13	UEP4	na	<-5
UEP1	<-2.5	<-5	UEP4-IL	na	1.18
UEP1-IL	<-2.5	<-5	UEP5	<-2.5	<-5
UEP2	-0.66	na	UEP5-IL	2.63	0
UEP2-IL	-0.53	na	UEP6	<-2.5	<-5
UEP3	<-2.5	-5	UEP6-IL	2.43	3.28

Table 3.16: E_b/N_0 gains obtained by different RSC-LSSTC benchmarks compared to the EEP-RSC-LSSTC at a PSNR of 36 dB. The schematic of Figure 3.1 configured with the parameters of Table 3.10 was used for partitioned-H.264 coded Football and Foreman sequences.

(dB)	Football	Foreman	(dB)	Football	Foreman
EEP-IL	0.57	0.66	UEP3	<-2.4	-2.4
UEP1	0.92	-2.76	UEP3-IL	1.18	0.48
UEP1-IL	1	-2.76	UEP4	<-2.4	<-4
UEP2	0.04	-1	UEP4-IL	0.92	-0.79
UEP2-IL	0.26	0.66			

Table 3.17: E_b/N_0 gains obtained by different Turbo-LSSTC benchmarks compared to the EEP-Turbo-LSSTC at a PSNR of 36 dB. The schematic of Figure 3.1 configured with the parameters of Table 3.10 was used for partitioned-H.264 coded Football and Foreman sequences.

(dB)	Football	Foreman	(dB)	Football	Foreman
EEP-IL	4.3	4.61	UEP3	3.05	1.33
UEP1	<-1	<-1	UEP3-IL	4.22	4.84
UEP1-IL	1.33	1.33	UEP4	2.89	2.5
UEP2	0.94	0.23	UEP4-IL	3.28	4.69
UEP2-IL	4.84	4.61			

Table 3.18: E_b/N_0 gains obtained by different SECCC-LSSTC benchmarkers compared to the EEP-SECCC-LSSTC at a PSNR of 36/35 dB. The schematic of Figure 3.1 configured with the parameters of Table 3.11 was used for SVC-H.264 coded Football and Foreman sequences.

the Football and Foreman sequences relative to the EEP-Turbo-LSSTC at a PSNR of 36 dB.

The PSNR performance of the IL-SECCC-LSSTC system is portrayed in Figure 3.28 using the Football sequence, where the FEC arrangements of Table 3.12 are employed. The IL-SECCC-LSSTC system of Figure 3.1 outperforms the conventional UEP aided system by about 3 dB of E_b/N_0 or 2.9 dB of PSNR at the cost of 1.5% complexity increase. The E_b/N_0 gains attained are listed in Table 3.18 relative to the EEP-SECCC-LSSTC at PSNR of 36/35 dB for the Football/Foreman sequence, respectively.

Furthermore, we proposed an optimization technique for IL-FEC coded scalable video transmission over wireless channels, where soft-decoding aided FEC is utilized. Firstly, LUTs were generated for the sake of optimizing the wireless transceivers and the FEC codec, which relied on exploiting the off-line training based MI. Moreover, the MI exchange of the VND and CND of Figure 3.3 was exploited for realistic video systems, where both the PERs of the IL-FEC coded BL and of the FEC coded ELs were estimated, based on the knowledgeable specific coding rates of all the scalable video layers. Then the expected PSNR reduction was minimized by performing a full search. BPSK modulation, an RSC FEC code and the SVC-H.264 video codec were employed in our design example, but our solution may be readily applied for arbitrary transceivers generating near-Gaussian distributed LLRs, for arbitrary non-iteratively decoded channel codes and for arbitrary layered video transmissions.

Finally, the PSNR performance of the Opt-UEP-IL system of Figure 3.16 is characterized in Figure 3.63 and Figure 3.64 using the Football and Foreman sequences,

(dB)	Football	Foreman
Opt-UEP	1.33	1.11
EEP-IL	1.56	1.44
Opt-UEP-IL	3.11	2.56

Table 3.19: E_b/N_0 gains obtained by different benchmarkers compared to the EEP-RSC at a PSNR of 36 dB. The schematic of Figure 3.16 configured with the parameters of Tables 3.13 and 3.14 was used for SVC-H.264 coded Football and Foreman sequences.

respectively. The proposed Opt-UEP-IL system of Figure 3.16 outperforms the traditional UEP aided system by about 1.9 dB of E_b/N_0 or 3.3 dB of PSNR. More specifically, the E_b/N_0 gains are listed in Table 3.19 for the Football and Foreman sequences compared to the EEP scheme at a PSNR of 36 dB.

For optimizing the IL-FEC system, it is an open challenge to theoretically solve the objective function of Eqs. (3.9) and (3.10), since all the characteristics of the fading wireless channel, of the modulator, of the soft-decoded FEC, of the interleaver etc. are involved. Estimating the PER $p(L_0)$ and $p(L_1)$ is one of the key issues for resolving the Eqs. (3.9) and (3.10). As shown in Eq. (3.8), the packet error ratio (PER) $p(L_1)$ of the enhancement layer (EL) L_1 is defined as

$$p(L_1) = f_1(SNR, |L_1|, r_1), \quad (3.28)$$

where the parameters SNR , $|L_1|$, r_1 jointly determine the PER $p(L_1)$. Hence we provide a three-dimensional graph in Figure 3.68, showing the trends of $p(L_1)$, when fitting r_1 as a constant of 0.33. Upon varying the values of r_1 , the 3D surface will extend to a 4D space.

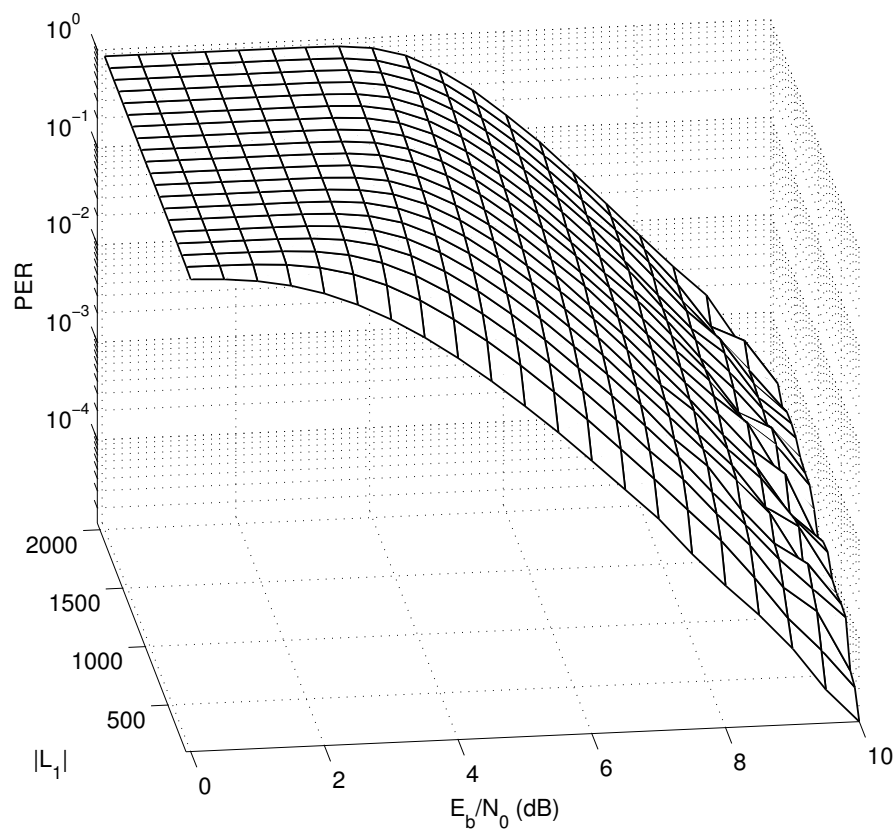


Figure 3.68: $p(L_1)$ with diverse SNR and $|L_1|$, when fitting r_1 as a constant of 0.33.

Iterative Source-Channel Decoding Aided Video Transmission

4.1 Introduction

Shannon's source-and channel-coding separation theorem [82] states that reliable transmission can be accomplished by separate source coding using lossless entropy codes and channel coding under the idealized assumption of Gaussian channels and potentially infinite encoding/decoding delay and complexity. However, it is impossible to remove all source redundancy with the aid of practical finite-delay and finite-complexity source encoders [166]. As a remedy, joint source-channel coding (JSCC) [95] has been used for achieving an improved system performance by exploiting the residual source correlation. A number of contributions [45, 83, 84, 98, 87] have been dedicated to audio transmission, which relied on exploiting the audio signal's residual correlation. Similarly, numerous techniques [86, 41] have been investigated in the context of video transmission scenarios.

The family of lossy video coding methods [167], such as the MPEG family and the ITU-T H.26x codecs have been researched for decades [1]. However, they impose a high encoder complexity, which may become excessive for deployment in wireless personal area networks (WPAN) [168, 169], for example. Secondly, the intrinsic latency may violate the delay budget of delay-sensitive applications, such as interactive gaming [170, 171]. Thirdly, some video quality degradation is inevitable and remains unrecoverable at the receiver, which may be unacceptable in high-quality medical applications [172, 173]. Last but not the least, compressed video streaming is limited to certain devices, categories where the matching encoding/decoding techniques are available. A transcodec converting between compressed video formats

is required, when a device has to relay the received video stream to another device employing a different compression technique, which may increase both the cost and complexity. Uncompressed video transmission [168,171] meets the requirements of high-quality home networking and quality/delay-sensitive medical applications, where low-complexity transmitters deliver unimpaired video signals, which may be processed by the receivers without any video decoder. The state-of-the-art techniques are reviewed below.

The emerging 60 GHz wireless personal area networks (WPAN) within the IEEE 802.15.3c standard family [174,175] is designed for short-range (<10 m) transmission of very-high-speed (>2 Gb/s) multimedia data to computer terminals and consumer appliances centered around an individual person's workspace, such as in residential rooms, offices, etc. The WirelessHD specification [176,177], as another WPAN standard, increases the maximum data rate to 28 Gb/s, which supports the transmission of either compressed or uncompressed digital high definition (HD) multimedia signals. In recent years, numerous investigations have been conducted in uncompressed video transmissions. Singh *et al.* [178,168,179] developed a system, where both unequal error protection (UEP) and automatic repeat request (ARQ) protocols were conceived for achieving an improved video quality. The authors of [180] investigated the specific technical challenges imposed by mm-wave systems supporting reliable video streaming using multi-beam transmissions. A flexible UEP method was proposed for the uncompressed video context in [181], which offers an improved visual quality and resource efficiency over both conventional UEP and equal error protection (EEP).

Since substantial residual source redundancy exists in many practical video applications, especially in uncompressed video transmission [168], it is beneficial to exploit the residual video redundancy for the sake of achieving an improved system performance. In this chapter, we consider a range of high-quality video transmission [171,168] scenarios. Specifically, we study the application of ISCD aided video communications, where the video signal is modelled by a first-order Markov process. Firstly, we derive reduced-complexity formulas for first-order Markov modelling (FOMM) [87] aided source decoding in Section 4.2.3. Then we propose a bit-based iterative horizontal-vertical scanline model (IHVSM) aided source decoding algorithm in Section 4.2.1, where a horizontal and a vertical source decoder exchange their extrinsic information using the iterative decoding philosophy. The iterative IHVSM aided decoder is then employed in a FEC encoded uncompressed video transmission scenario, where the IHVSM and the FEC decoder exchange softbit-information for performing turbo-like [182] ISCD for the sake of improving the reconstructed video quality. Section 4.2.4 quantifies the attainable system performance of the intra-frame

ISCD scheme against a near-lossless H.264/AVC [14] video communication system and the existing FOMM based softbit source decoding scheme, where the softbit decoding is performed by a one-dimensional Markov model aided decoder.

Furthermore, we study the application of three-dimensional spatio-temporal ISCD aided uncompressed video transmission in Section 4.3, where the correlation inherent in the video signals is modelled by a first-order Markov process. Firstly, we propose first-order Markov process aided Three-Dimensional Iterative Source-Channel decoding using a Recursive Systematic Convolutional (M3DISC-RSC) codec in Section 4.3.1. More explicitly, both the horizontal and the vertical intra-frame correlations as well as the inter-frame correlations are exploited by the receiver, hence we rely on three-dimensional (3D) information exchange. This scheme may be combined with arbitrary channel codecs. Then in Section 4.3.2.2 we will analyze the three-stage decoder's convergence behavior using 3D EXIT charts [132, 133]. Finally, we benchmark the attainable system performance against a couple of video communication systems in Section 4.3.2.4, including the 2D ISCD scheme of Section 4.2, where only the intra-frame correlations were exploited without invoking a channel codec.

4.2 Intra-Frame Iterative Source-Channel Decoding

4.2.1 Iterative Source Decoding Model

A one-dimensional iterative system model has been proposed and analyzed by Vary and his team [84, 83, 98, 87] in the context of audio signals. Since their model was designed for one-dimensional signals, it cannot directly exploit the two-dimensional (2D) correlation of practical video signals. In this section, the 2D video correlation is modeled and exploited by our proposed IHVSM, which consists of multiple horizontal and vertical scanline based Markov processes. Moreover, the IHVSM exchanges extrinsic information between the horizontal and vertical scanlines by performing iterative source decoding, thereby correcting the errors within the 2D video frames. We initially focus our attention on the receiver side. The details of applying the 2D IHVSM decoder in FEC encoded uncompressed video transmission will then be illustrated in Section 4.2.4. Let us commence by stipulating the following assumptions:

- x_i : the m -bit pattern of pixel scanned from the original video pixels at time instant i , which is expressed as $\{x_i(0), \dots, x_i(m-1)\} = x_i \binom{m-1}{0}$;
- m : the number of bits in each m -bit pattern x_i of pixels;
- $X_m = \{0, 1, \dots, 2^m - 1\}$: the set of all possible values in an m -bit pattern x_i ;
- $x_0^t = x_0, \dots, x_t$: the bit patterns of the 1^{st} frame of the original video consisting of $(t+1)$ m -bit patterns during the time interval spanning from 0 to t ;

- $y_0^t = y_0, \dots, y_t$: potentially error-infested bit pattern of the 1st frame;

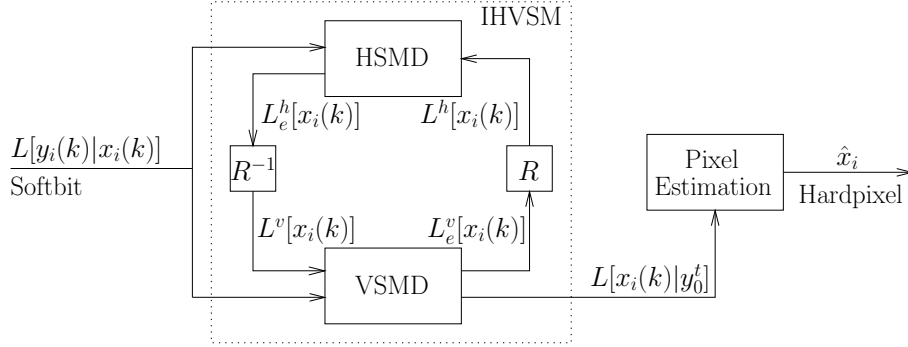


Figure 4.1: Iterative error concealment architecture using IHVSM, where R represents reordering of the video pixels.

The 2D iterative source decoding model is displayed in Figure 4.1, which only accepts the soft bit input and generates the corresponding hard-decision-based pixel as output. We will discuss the details of 2D iterative source decoding techniques below.

4.2.1.1 Softbit¹ Input

Let us now focus our attention on the 1st original video frame carrying $(t + 1)$ consecutive and hence correlated bit patterns, eg. x_0, \dots, x_t . Due to the channel-impairments, the receiver can only reconstruct the error-infested version of x_0, \dots, x_t . Again, the 2D iterative source decoder shown in Figure 4.1 only accepts softbit information as its input. Then, the receiver has to estimate the softbit information of each bit representing the original pixels x_0, \dots, x_t , namely y_0, \dots, y_t , which carry the error-infested bit sequence $y_0 \binom{m-1}{0}, \dots, y_t \binom{m-1}{0}$. The softbit information represented by the log-likelihood ratios (LLR) may be readily derived from the softbit patterns y_0, \dots, y_t , yielding $L[y_i(k)|x_i(k)]$ in Figure 4.1, which indicates the receiver's confidence in the original m -bit pixel $x_i(k)$.

For example, let us assume that we have an original pixel value of $x_i = 101$ ($0 \leq i \leq t$), which can be represented using $m = 8$ bits, namely as $x_i \binom{7}{0} = (01100101)$. Then the related soft pixel value may be expressed as

$$y_i = (0.3, -0.2, -0.4, 0.1, 0.2, -0.6, 0.4, -0.4) \quad (4.1)$$

representing the confidence of the bits (01100101), respectively.

¹Softbit source decoding was proposed in [84]. Instead of expressing a bit as 0 or 1, a softbit represents one bit of information using a floating-point number, indicating our confidence in that bit.

4.2.1.2 Iterative Source Decoding

Given the softbit LLR input $L[y_i(k) | x_i(k)]$ of Figure 4.1, our 2D iterative source decoder may be invoked for correcting the effects of channel-errors on the error-infested m -bit sequence of $(t + 1)$ pixels $y_0 \binom{m-1}{0}, \dots, y_t \binom{m-1}{0}$. Two stages are involved in the iterative source decoding process, namely the IHVSM based decoding and the related pixel estimation, as seen in Figure 4.1. In the IHVSM based decoding stage, the reliability of the soft information is improved by exchanging extrinsic information between the horizontal and vertical decoders. Then the final LLRs are generated, which will be used for estimating the pixels during the pixel estimation stage.

4.2.1.2.1 IHVSM based decoding At the first stage, two similar source decoders are employed, namely the Horizontal Scanline Model Decoder (HSMD) modeling the video correlation in the horizontal direction and the Vertical Scanline Model Decoder (VSMD) modeling the video correlation in the vertical direction. Let us now consider H horizontal scanlines and V vertical scanlines, which are modeled by H HSMDs and V VSMDs, respectively. Then the IHVSM decoding may be performed based on the $(H \times V)$ -pixel block, which is represented by the $(H \cdot V)$ - m -softbit patterns y_i . Here we consider the HSMD as an example for highlighting the decoding process. The m -softbit-based pixel value y_i is used for generating the systematic LLR information $L[y_i(k) | x_i(k)]$, which is then input to the HSMD without the assistance of any channel decoding. However, our technique may be conceived with arbitrary channel codecs. The HSMD also exploits the *a-priori* LLR information $L^h[x_i(k)]$ generated from the extrinsic information $L_e^v[x_i(k)]^2$ provided by the vertical decoder. The horizontal decoder independently performs source-modelling aided soft decoding of each of the H horizontal scanlines. Hence, following horizontal decoding, the extrinsic LLR information $L_e^h[x_i(k)]$ may be generated for all H scanlines. Then the relevant $(H \times V)$ -line extrinsic information block will be appropriately reordered in the block R^{-1} of Figure 4.1 for generating the *a-priori* information $L^v[x_i(k)]$ for the vertical decoder.

For further illustration of the IHVSM decoding process, consider the example of the horizontal and vertical scanlines displayed in Figure 4.2, where the pixels connected by the solid/dashed arrows belong to a horizontal/vertical Markov process. Specifically, the 5 pixels of the first row in Figure 4.2 are modelled by a 5-pixel Markov process h_1 . Let us assume that the 5 pixel values available at the receiver may be expressed as $y_{1,1}, \dots, y_{1,5}$ of Figure 4.2, each of which consists of m soft

²The first-order Markov process aided decoding algorithm will be detailed in Section 4.2.3, along with the associated iterative decoding principle.

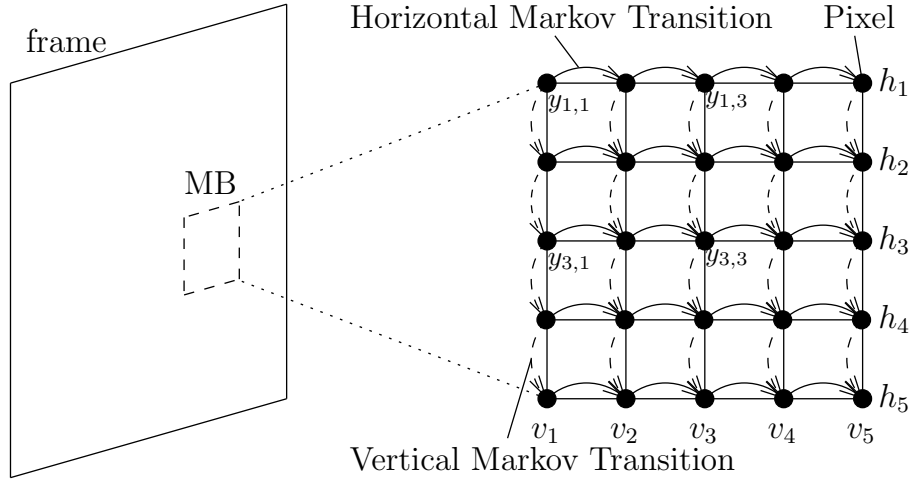


Figure 4.2: The structure of horizontal scanlines and vertical scanlines and their corresponding Markov processes. $y_{i,j}$ represents the received pixel at the position (i, j) in the MB, which corresponds to the error-infested version of original pixel $x_{i,j}$.

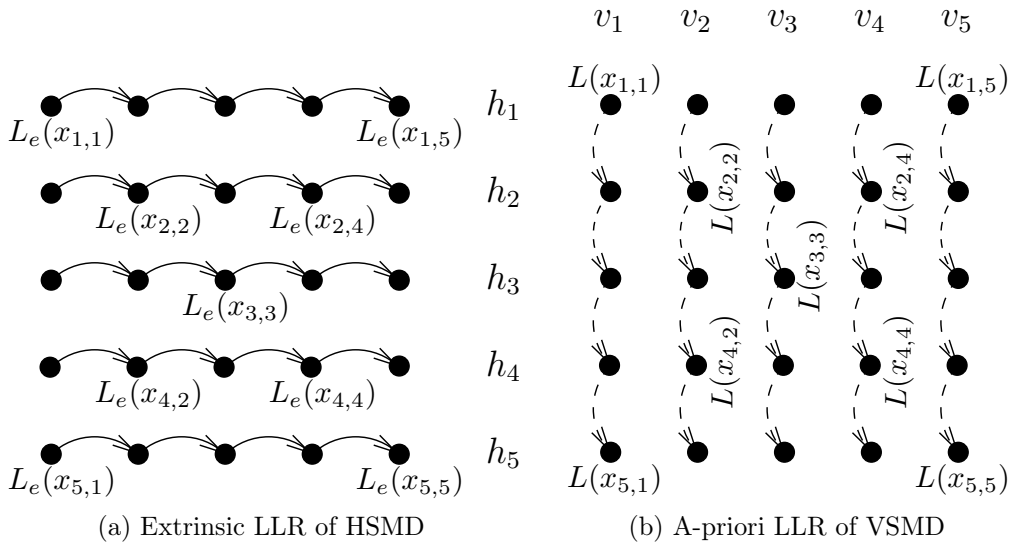


Figure 4.3: The extrinsic information generated by the horizontal decoders is reordered for the sake of generating the a-priori information of the vertical decoders by the R^{-1} block of Figure 4.2.

values indicating the decoder's confidence in each of the m systematic bits³. The corresponding *a-priori* LLR information $L^h [x_{1,1} \binom{m-1}{0}], \dots, L^h [x_{1,5} \binom{m-1}{0}]$ for the 5 pixels is provided by the VSMD decoder. The $(5 \times m)$ soft values representing the 5 pixels can exchange extrinsic information with each other, when the HSMD decoder is processing scanline h_1 . Similarly, the pixel confidences of all 5 horizontal scanlines may be improved by performing HSMD on h_1, \dots, h_5 , respectively. Note that the pixels only shared their information within the horizontal scanlines in the HSMD process. After HSMD decoding, the extrinsic information provided by the (5×5) pixels, namely the $(5 \times 5 \times m)$ soft values of $L_e^h [x_{1,1} \binom{m-1}{0}], \dots, L_e^h [x_{5,5} \binom{m-1}{0}]$ are generated, which are organized as seen in Figure 4.3a. Then, these $(5 \times 5 \times m)$ extrinsic information values will be reordered into vertical scanlines by the block R^{-1} of Figure 4.1, which will be utilized as *a-priori* information by the VSMD decoder, as organized in Figure 4.3b. Then the VSMD decoder will decode the vertical Markov process v_1 based on the systematic information $y_{1,1}, \dots, y_{5,1}$, whilst also exploiting the *a-priori* LLR information $L^v [x_{1,1} \binom{m-1}{0}], \dots, L^v [x_{5,1} \binom{m-1}{0}]$ reordered by R^{-1} . Similar to the HSMD, the VSMD will improve the decoder's confidence related to the pixels by exchanging extrinsic information in the vertical direction. After VSMD processing of the vertical scanlines v_1, \dots, v_5 , respectively, a full iteration of the IHVSM scheme is completed and the HSMD will exploit the extrinsic information fed back by the VSMD decoder in order to start the next iteration. This iterative process will terminate after the affordable number of iterations. Observe that any two pixels of an MB are related to each other according to our Markov model and can exchange information with each other with the aid of the IHVSM decoder. Moreover, the IHVSM decoder performs decoding on a block by block basis, as defined in the system.

4.2.1.2.2 Pixel Estimation Following the IHVSM decoding process, the relevant *a-posteriori* information is generated at the parameter estimation stage by summing the systematic LLR information $L[y_i(k)|x_i(k)]$, the *a-priori* LLR information $L^v[x_i(k)]$ and the extrinsic LLR information $L_e^h[x_i(k)]$, where the *a-posteriori* information $L[x_i|y_0^t]$ may be exploited either by the minimum mean square error (MMSE) decoder or the maximum *a posteriori* probability (MAP) decoder for estimating the m -bit pattern x_i and ultimately to output the estimated original pixel \hat{x}_i [84], which may be formulated as

³The m soft values may be acquired in different ways in different applications. In the application of DVC, the m soft values of the pixels in the WZ frames are estimated from the key frames, which are intra-coded and transmitted to the receiver [103].

- MAP estimator

$$\hat{x}_i = \arg \max_{\forall x_i \in X_m} p(x_i | y_0^t); \quad (4.2)$$

- MMSE estimator

$$\hat{x}_i = \sum_{x_i \in X_m} x_i \cdot p(x_i | y_0^t). \quad (4.3)$$

Both the MMSE estimator of Eq. (4.3) and the MAP estimator of Eq. (4.2) may be deemed to be symbol-based estimators, since they are based on the *a-posteriori* probability $p(x_i | y_0^t)$ of the m -bit pattern x_i , which is conditioned on the received frame of bit patterns y_0^t . However, since the specific bits $x_i(0), \dots, x_i(m-1)$ of the pixels are independent of each other, if the bit-based *a-posteriori* probability $p[x_i(k) | y_0^t]$ is provided by the iterative decoding process, the symbol-based *a-posteriori* probability $p(x_i | y_0^t)$ in Eq. (4.2) and Eq. (4.3) may be derived from the bit-based information as their product:

$$p(x_i | y_0^t) = \prod_{k=0}^{m-1} p[x_i(k) | y_0^t]. \quad (4.4)$$

Furthermore, the symbol-based MMSE estimation rule may also be derived from the bit-based probability $p[x_i(k) | y_0^t]$ as follows

$$\begin{aligned} \hat{x}_i &= \sum_{x_i \in X_m} x_i \cdot p(x_i | y_0^t) \\ &= \sum_{x_i \in X_m} x_i \cdot \prod_{k=0}^{m-1} p[x_i(k) | y_0^t] \\ &= \sum_{x_i \in X_m} \sum_{k=0}^{m-1} x_i(k) \cdot 2^k \cdot \prod_{l=0}^{m-1} p[x_i(l) | y_0^t] \\ &= \sum_{k=0}^{m-1} 2^k \cdot p[x_i(k) = 1 | y_0^t] \sum_{\substack{x_i \in X_m \\ x_i(k)=1}} \prod_{\substack{l=0 \\ l \neq k}}^{m-1} p[x_i(l) | y_0^t], \end{aligned} \quad (4.5)$$

It may be readily shown that $\sum_{\substack{x_i \in X_m \\ x_i(k)=1}} \prod_{\substack{l=0 \\ l \neq k}}^{m-1} p[x_i(l)|y_0^t] = 1$, hence the symbol-based MMSE estimator \hat{x}_i can be derived from the bit-based probabilities as

$$\hat{x}_i = \sum_{k=0}^{m-1} 2^k \cdot p[x_i(k) = 1|y_0^t]. \quad (4.6)$$

Finally, the hard pixel \hat{x}_i of Figure 4.1 may be returned as the final estimate of the original pixel x_i .

4.2.2 Existing First-Order Markov Modeled Softbit Source Decoding

The employment of first-order Markov modelling aided softbit source decoding was discussed in [84, 98, 87], which we refer to as FOMM-SBSD for simplicity. The *a-posteriori* log-likelihood ratio of bit $x_i(k)$ conditioned on y_0^t was derived in [87, 98] based on either a first-order or a zero-order Markov model, together with the rules of determining the extrinsic information $L_e[x_i(k)]$. Below we will briefly review the formulation of the first-order Markov case, while the zero-order Markov model may be readily deduced.

The extrinsic information gleaned from another decoder, for example from a channel decoder or a short block codec (SBC) [40], will be used as the *a-priori* information $L[x_i(l)]$ in the source decoder. Provided that a systematic encoder, such as for example SBC code is employed, the decoder's output information $L[y_i(l)|x_i(l)]$ becomes readily available for the source decoder. The bit-wise log-likelihood information has to be combined in order to generate its corresponding symbol-based version for the source decoder. In the case of a systematic encoder, the symbol-based information may be directly derived from the superposition of the decoder's output information and of the *a-priori* information, which is formulated as [87]

$$\gamma_i(x_i) = \exp \left\{ \sum_{l=0}^{m-1} \frac{\bar{x}_i(l)}{2} \cdot \{L[x_i(l)] + L[y_i(l)|x_i(l)]\} \right\}. \quad (4.7)$$

By contrast, when a non-systematic encoder is employed, we have

$$\gamma_i(x_i) = \exp \left\{ \sum_{l=0}^{m-1} \frac{\bar{x}_i(l)}{2} \cdot L[x_i(l)] \right\}, \quad (4.8)$$

where $\gamma_i(x_i)$ is the occurrence probability of the m -bit pattern x_i .

Below, we will consider the systematic encoding case, while the non-systematic case may be derived similarly. In the systematic case, the *a-posteriori* log-likelihood ratio of bit $x_i(k)$ conditioned on y_0^t may be expressed as

$$L[x_i(k)|y_0^t] = \log \frac{\sum_{\substack{x_i \in X_m \\ x_i(k)=0}} \beta_i(x_i) \cdot \gamma_i(x_i) \cdot \sum_{x_{i-1} \in X_m} p(x_i|x_{i-1}) \cdot \alpha_{i-1}(x_{i-1})}{\sum_{\substack{x_i \in X_m \\ x_i(k)=1}} \beta_i(x_i) \cdot \gamma_i(x_i) \cdot \sum_{x_{i-1} \in X_m} p(x_i|x_{i-1}) \cdot \alpha_{i-1}(x_{i-1})} \quad (4.9)$$

$$= L[x_i(k)] + L[y_i(k)|x_i(k)] + L_e[x_i(k)],$$

where $L_e[x_i(k)]$ is the extrinsic log-likelihood ratio (LLR) of the bit $x_i(k)$ gleaned from the SBS, which will be used as *a-priori* information by the other decoder. The extrinsic LLR component in Eq. (4.9) is expressed in [87]

$$L_e[x_i(k)] = \log \frac{\sum_{\substack{x_i \in X_m \\ x_i(k)=0}} \beta_i(x_i) \cdot \gamma_i^{[ext]}[x_i(k)] \cdot \sum_{x_{i-1} \in X_m} p(x_i|x_{i-1}) \cdot \alpha_{i-1}(x_{i-1})}{\sum_{\substack{x_i \in X_m \\ x_i(k)=1}} \beta_i(x_i) \cdot \gamma_i^{[ext]}[x_i(k)] \cdot \sum_{x_{i-1} \in X_m} p(x_i|x_{i-1}) \cdot \alpha_{i-1}(x_{i-1})} \quad (4.10)$$

where $\gamma_i^{[ext]}[x_i(k)]$ is the extrinsic information of bit $x_i(k)$, which is derived from the other $(m-1)$ bits of the m -bit pattern x_i describing a pixel. Moreover, $\gamma_i^{[ext]}[x_i(k)]$ may be formulated as

$$\gamma_i^{[ext]}[x_i(k)] = \exp \left\{ \sum_{l=0, l \neq k}^{m-1} \frac{\bar{x}_i(l)}{2} \cdot \{L[x_i(l)] + L[y_i(l)|x_i(l)]\} \right\} \quad (4.11)$$

In Eq. (4.10), the components $\alpha_{i-1}(x_{i-1})$ and $\beta_i(x_i)$ may be computed using the classic forward and backward recursion calculation respectively [183] as

$$\alpha_{i-1}(x_{i-1}) = \gamma_{i-1}(x_{i-1}) \sum_{x_{i-2} \in X_m} p(x_{i-1}|x_{i-2}) \cdot \alpha_{i-2}(x_{i-2})$$

$$\beta_i(x_i) = \sum_{x_{i+1} \in X_m} p(x_{i+1}|x_i) \cdot \gamma_{i+1}(x_{i+1}) \cdot \beta_{i+1}(x_{i+1}). \quad (4.12)$$

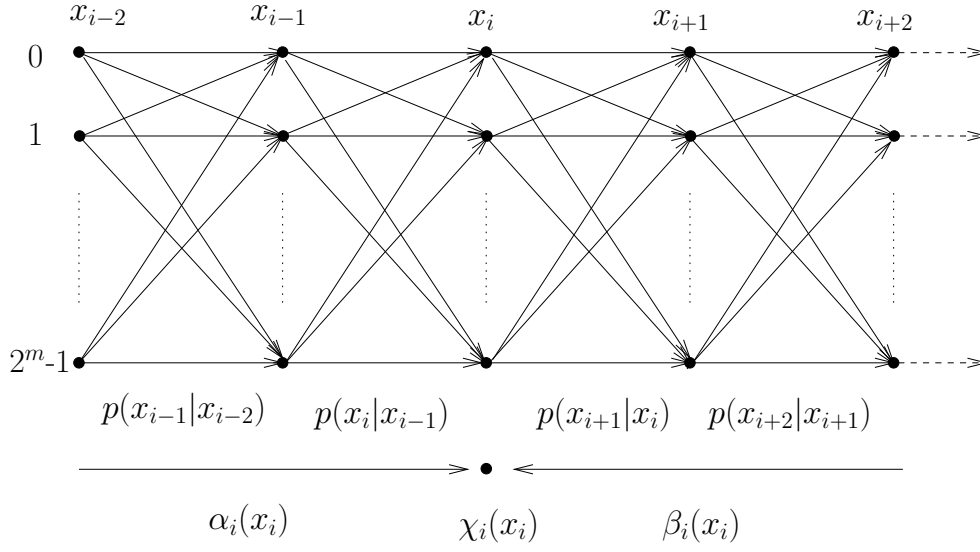


Figure 4.4: Trellis of first-order Markov process for BCJR decoding, where $p(x_{i+1}|x_i)$ is the Markov transition probability.

4.2.3 Reduced Complexity Markov-Modelled Softbit Source Decoding

The existing first-order Markov modelling of the source decoder was briefly reviewed in Section 4.2.2. The determination of the bit-based extrinsic information was detailed in Eq. (4.10). However, this formulation of the bit-based extrinsic information $L_e[x_i(k)]$ may be further simplified and correspondingly the complexity imposed may be reduced. Below we will first present our novel trellis representation of the first-order Markov process in Section 4.2.3.1. Then we will detail the decoding rules of our trellis, which is a reduced complexity version of the existing technique. Finally we will present our extrinsic information formula invoked for iterative decoding.

4.2.3.1 Trellis Representation of First-Order Markov Chain

The traditional MAP decoder trellis for an RSC code is detailed in Figure 4.6 of [78], where the nodes indicate the memory states of the RSC encoder, while the lines indicate the state transitions. Specifically, there are four states, each of which has two possible transitions indicating the input bits of 0 and 1. All the transitions are determined by the generator polynomials of the RSC encoder.

In contrast to the traditional one, the trellis of the first-order Markov process is shown in Figure 4.4, where the nodes are the trellis states, namely the m -bit pixels x_{i-2}, \dots, x_{i+2} , while the lines indicate the state transitions. Moreover, the probability of a transition from state x_i to state x_{i+1} is represented by $p(x_{i+1}|x_i)$, which is the state transition probability of the related Markov process. There are 2^m states for the m -bit pixels x_{i-2}, \dots, x_{i+2} , each of which has 2^m possible transitions.

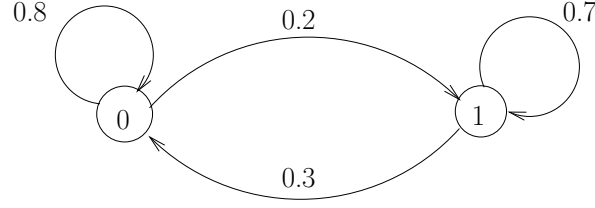


Figure 4.5: An instance of a 2-state ($m = 1$) first-order Markov process transition probability graph.

States	0	1
0	0.8	0.2
1	0.3	0.7

Table 4.1: Markov transition probability table for Figure 4.5.

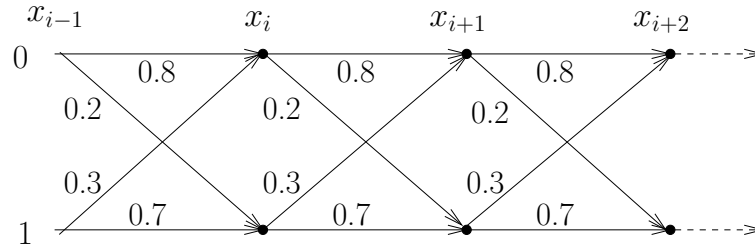


Figure 4.6: Trellis-representation of the first-order Markov process of Figure 4.5.

Our proposed trellis is different from the traditional trellis in the following aspects. Firstly, the states of the traditional trellis indicate the memory states of the RSC encoder, while the states of our trellis indicate the legitimate m -bit pixel values. Secondly, the transitions of the traditional trellis indicate the binary inputs 0, 1, while the transitions of our trellis indicate the Markov state transitions. Finally, there are two possible transitions for each state of the traditional trellis, while there are 2^m possible transitions for each state of our trellis. A specific instance of the trellis representation and its Markov state transition graph is displayed in Figure 4.5 and Figure 4.6. The state transition graph of a 2-state first-order Markov process is displayed in Figure 4.5, while the related probability transition table is shown in Table 4.1.

4.2.3.2 BCJR Decoding of First-Order Markov Chain

The existing APP generation rule conceived for first-order Markov modeled SBSD is shown in Eq. (4.9). However, we will demonstrate that the complexity imposed may be reduced by invoking our novel trellis introduced in Section 4.2.3.1.

As illustrated in Section 4.2.1, the soft Markov processes y_0, \dots, y_t ⁴ are known at the Markov decoder, which represented the error-infested soft information of the original Markov process x_0, \dots, x_t . The objective of the Markov decoder is to estimate the quantized pixels x_0, \dots, x_t based on the soft pixels. Given the soft information y_0, \dots, y_t of the Markov process, the quantized pixel x_i may be determined by the APP

$$p(x_i | y_0, \dots, y_t) = p(x_i | y_0^t), \quad (4.13)$$

where we have $x_i \in X_m$, $0 \leq i \leq t$. In the Appendix A, we show that the bit-based LLR version of the APP $p(x_i | y_0^t)$ may be expressed as

$$L[x_i(k) | y_0^t] = \ln \frac{\sum_{\substack{x_i(k)=0 \\ x_i \in X_m}} \beta_i(x_i) \cdot \chi_i(x_i) \cdot \alpha_i(x_i)}{\sum_{\substack{x_i(k)=1 \\ x_i \in X_m}} \beta_i(x_i) \cdot \chi_i(x_i) \cdot \alpha_i(x_i)}, \quad (4.14)$$

and based on this, the decoder may determine the bit $x_i(k)$.

We have hence derived the final rule of determining the bit-based *a-posteriori* probability LLR, which is represented in Eq. (A.8). For the conventional determination rule formulated in Eq. (4.9), the components β , χ and α are defined in [87], which impose a similar complexity to our definitions in Eqs. (A.6), (A.4) and (A.5). However, the inner summation $\sum p(x_i | x_{i-1}) \cdot \alpha_{i-1}(x_{i-1})$ of Eq. (4.9) is avoided in our decoding rule of Eq. (A.8), which reduces the computational complexity imposed.

4.2.3.3 Extrinsic Information Exchange for Iterative Decoding

A limitation of the formulas provided in Section 4.2.3.2 is that they cannot be directly used for iterative decoding, since they cannot exploit the *a-priori* information $L[x_i(k)]$, which was generated from the *extrinsic* information gleaned from the other decoder. The rules of iterative source and channel decoding were derived in [87, 98]. To make use of the *a-priori* information $L[x_i(k)]$, the combined bit-wise log-likelihood information⁵ may be utilized as [87]

$$\gamma_i(x_i) = \exp \sum_{k=0}^{m-1} \frac{\bar{x}_i(k)}{2} \cdot \{L[x_i(k)] + L[y_i(k) | x_i(k)]\} \quad , \quad (4.15)$$

⁴As detailed in Section 4.2.1, each soft pixel y_i consists of m floating values $y_i(0), \dots, y_i(m-1)$, representing the confidences of the original m bits $x_i(0), \dots, x_i(m-1)$.

⁵Similar to Eq. (A.4), a constant normalization factor is neglected since it will be cancelled during the calculation.

where the symbol-based m -bit information γ is the combination of the bit-wise log-likelihood *a-priori* information $L[x_i(k)]$ and of the channel information $L[y_i(k)|x_i(k)]$. We note in this context that γ of Eq. (4.15) contains more valuable information than the channel information χ . By replacing χ of Eq. (A.8) with γ of Eq. (4.15), we have the following formula:

$$L[x_i(k)|y_0^t] = \ln \frac{\sum_{\substack{x_i \in X_m \\ x_i(k)=0}} \beta_i(x_i) \cdot \gamma_i(x_i) \cdot \alpha_i(x_i)}{\sum_{\substack{x_i \in X_m \\ x_i(k)=1}} \beta_i(x_i) \cdot \gamma_i(x_i) \cdot \alpha_i(x_i)}, \quad (4.16)$$

where the forward and backward recursion components β, α are defined as follows

$$\begin{aligned} \alpha_i(x_i) &= \sum_{x_{i-1} \in X_m} \gamma_{i-1}(x_{i-1}) \cdot p(x_i|x_{i-1}) \cdot \alpha_{i-1}(x_{i-1}) \\ \beta_i(x_i) &= \sum_{x_{i+1} \in X_m} \beta_{i+1}(x_{i+1}) \cdot \gamma_{i+1}(x_{i+1}) \cdot p(x_{i+1}|x_i). \end{aligned} \quad (4.17)$$

Moreover the symbol-based *a-posteriori* probability of the first and last m -bit patterns may be expressed as follows

$$\begin{aligned} p(x_0 \wedge y_0^t) &= \beta_0(x_0) \cdot \gamma_0(x_0) \cdot p(x_0) \\ p(x_t \wedge y_0^t) &= \gamma_t(x_t) \cdot \alpha_t(x_t). \end{aligned} \quad (4.18)$$

Similar to the BCJR decoding technique of classic turbo codes [78], the bit-based *a-posteriori* LLR $L[x_i(k)|y_0^t]$ may be split into three components, namely the *a-priori* information $L[x_i(k)]$, the channel information $L[y_i(k)|x_i(k)]$ and the extrinsic information $L_e[x_i(k)]$. Specifically, the bit-based *a-posteriori* information $L[x_i(k)|y_0^t]$ may be formulated as

$$\begin{aligned} L[x_i(k)|y_0^t] &= L[x_i(k)] + L[y_i(k)|x_i(k)] + \ln \frac{\sum_{\substack{x_i \in X_m \\ x_i(k)=0}} \beta_i(x_i) \cdot \gamma_i^{[ext]}[x_i(k)] \cdot \alpha_i(x_i)}{\sum_{\substack{x_i \in X_m \\ x_i(k)=1}} \beta_i(x_i) \cdot \gamma_i^{[ext]}[x_i(k)] \cdot \alpha_i(x_i)} \\ &= L[x_i(k)] + L[y_i(k)|x_i(k)] + L_e[x_i(k)], \end{aligned} \quad (4.19)$$

where the extrinsic information component $\gamma_i^{[ext]}[x_i(k)]$ may be expressed as

$$\gamma_i^{[ext]}[x_i(k)] = \exp \sum_{l=0, l \neq k}^{m-1} \frac{\bar{x}_i(l)}{2} \cdot \{L[x_i(l)] + L[y_i(l)|x_i(l)]\}. \quad (4.20)$$

4.2.3.4 Complexity Analysis

The complexity of our proposed first-order Markov process based decoder can be attributed to the calculation of $\gamma_i(x_i)$ in Eq. (4.15), $\alpha_i(x_i)$ in Eq. (A.5), $\beta_i(x_i)$ in Eq. (A.6) and $L[x_i(k)|y_0^t]$ in Eq. (4.16). As shown in Figure 4.4, the trellis size is $(2^m \cdot t)$, where t is the length of the Markov process. Similar to the BCJR decoding rules proposed in [183], the decoding of the $(2^m \cdot t)$ -state trellis of Figure 4.4 may be generalized into the following two stages:

- Calculation of γ , α and β : These operations are carried out across the entire Markov trellis of Figure 4.4, which imposes the complexity of m , 2^m , 2^m for each trellis state, as suggested by Eqs. (4.15), (A.5), (A.6), respectively. Hence the associated computational costs are on the order of $O(2^m \cdot t \cdot m)$, $O(2^{2m} \cdot t)$ and $O(2^{2m} \cdot t)$ for γ , α and β , respectively.
- Calculation of $L[x_i(k)|y_0^t]$: This operation is carried out for all the $t \cdot m$ bits of the t -pixel Markov process, which imposes a complexity of 2^m for each bit. Hence the computational cost is on the order of $O(2^m \cdot t \cdot m)$.

Therefore, the overall complexity imposed by our proposed decoder is

$$O(2 \cdot 2^{2m} \cdot t + 2 \cdot 2^m \cdot t \cdot m), \quad (4.21)$$

when decoding the t -pixel Markov process.

Similarly, for the conventional Markov process based decoding technique of Eq. (4.9) [87], the associated computational costs are on the order of $O(2^m \cdot t \cdot m)$, $O(2^{2m} \cdot t)$, $O(2^{2m} \cdot t)$ for γ , α , β , respectively, where the relevant definitions of γ , α , β [87] are not included in this treatise for the sake of space economy. However, the inner summation $\sum p(x_i|x_{i-1}) \cdot \alpha_{i-1}(x_{i-1})$ in Eq. (4.9) imposes $2^{2m} \cdot t$ more operations than Eq. (4.16), which is calculated in advance. Therefore the complexity of the conventional Markov process based decoding algorithm is on the order of $O(3 \cdot 2^{2m} \cdot t + 2 \cdot 2^m \cdot t \cdot m)$.

Based on the discussions above, the percentage of complexity reduction achieved by our proposed decoder may be expressed as $2^m/(3 \cdot 2^m + 2 \cdot m)$, which increases upon increasing m . Specifically, complexity reductions of 25% and 33% may be

achieved for $(m = 1)$ and $(m = 8)$, respectively. Note that the decoding complexity of the first-order Markov process increases exponentially with the number of bits per symbol m . Hence, a quantizer may be employed for striking a tradeoff between the complexity imposed and the attainable performance.

4.2.3.5 Parameter Training for Markov Processes

At the receiver, the first-order Markov process is utilized for modelling the correlation within multiple video frames. However, the Markov Model's State Transition Table (MMSTT) must be appropriately trained for the sake of reflecting the correlations inherent in a specific video sequence. Let us now detail the training process using a gray-scale video sequence. The same process may be readily applied to color sequences. Let us commence by assuming that the training video sequence contains f frames, each of which carries $(W \times H)$ m -bit pixels, where W is the width and H is the height of a figure. To train the MMSTT parameters for the horizontal and the vertical Markov processes, we firstly initialize the $(2^m \times 2^m)$ -element MMSTT $T[0 : 2^m - 1, 0 : 2^m - 1]$ to zero values. Then we scan all the H horizontal scanlines from left to right and the W vertical scanlines from top to bottom in each frame of the training video sequence. For all scanlines, when pixel s_{i-1} and pixel s_i are scanned, the corresponding element $T[s_{i-1}, s_i]$ in the MMSTT is increased by 1. Finally, by normalizing the summation of all rows in the MMSTT $T[0 : 2^m - 1, 0 : 2^m - 1]$, the first-order Markov transition probabilities $p(s_j | s_i)$ can be obtained, where we have $s_i, s_j \in [0, 2^m)$. Similarly, a total of $(W \cdot H)$ scanlines along the time axis are used for training the parameters of the inter-frame Markov processes, which are constituted by the consecutive video frames. Each of the sequences contains f pixels, which are extracted from the same position of f consecutive video frames. This inter-frame Markov process may be utilized for exploiting the inherent inter-frame correlation, as it will be detailed in Section 4.3.

Hence there are two MMSTTs constructed after the training process. They may be approximated by the Laplace distribution and used at the receiver. The parameter training process employed in this paper simply requires the evaluation, which is a low-complexity off-line process. Alternatively, the correlation may be evaluated from the stored previously transmitted video, which is more representative of the signal transmitted. The same process may be invoked at the receiver, provided that error probability is low. Furthermore, as we stated in the manuscript, the Markov transition table can be readily approximated using the analytical Laplace distribution. Hence, the parameter training process does not increase the decoding time and does not require a high signaling overhead. Alternatively, they may be

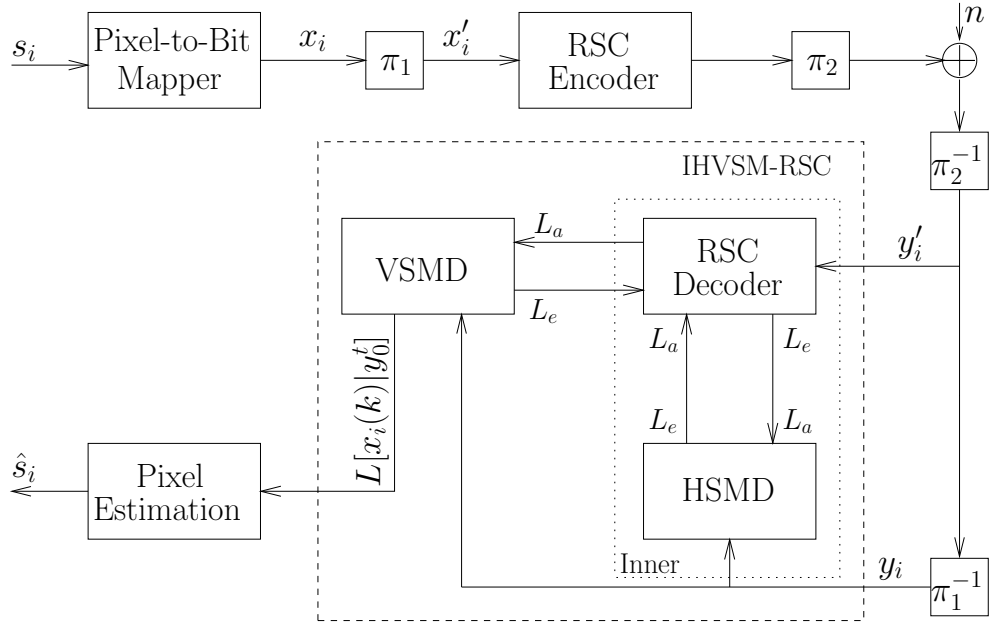


Figure 4.7: System architecture of IHVSM-RSC for communication of RSC encoded uncompressed video through wireless channel.

trained on a long and sufficiently diverse video sequence. In this treatise, we do not focus on this issue in more depth. Note that the accuracy of transition probability tables substantially affects the performance of the system.

4.2.4 ISCD Aided Uncompressed Video Transmission

Note that the peak to-signal-noise ratio (PSNR) metric is employed for quantifying the reconstructed video quality in this section. To avoid having infinite PSNR values when a video is perfectly reconstructed, we artificially set the total averaged mean squared error (MSE) value between the reconstructed and the original frame to a minimum value of 1. This is justified, since the same technique is employed in the H.264 reference software JM. Hence the maximum unimpaired video PSNR that may be obtained at the receiver is about 48.1 dB.

In this section, we consider the transmission of RSC encoded uncompressed video through a Rayleigh channel. The demodulator estimates the softbit information at the receiver, thereby meeting the essential prerequisites of Section 4.2.1 for the application of our proposed iterative source decoding technique. Furthermore, the IHVSM and the RSC codec will exchange extrinsic information for concealing the errors within the video frames.

	Akiyo	Coastguard	Football
Representation	YUV 4:2:0	YUV 4:2:0	YUV 4:2:0
Format	QCIF, CIF	QCIF, CIF	QCIF, CIF
Bits Per Pixel	8	8	8
FPS	30	15	15
Number of Frames	100	100	100
Bitrate (Mbps)	9.1, 36.5	4.6, 18.2	4.6, 18.2
“Natural” Code Rate	1/8.4, 1/7.5	1/2.4, 1/2.2	1/1.8, 1/2.1
Channel	Unc-Ray	Unc-Ray	Unc-Ray
Modulation	BPSK	BPSK	BPSK

Table 4.2: Parameters employed for the video sequences used for the characterization of the system seen in Figure 4.7. *Unc-Ray* stands for Uncorrelated Rayleigh.

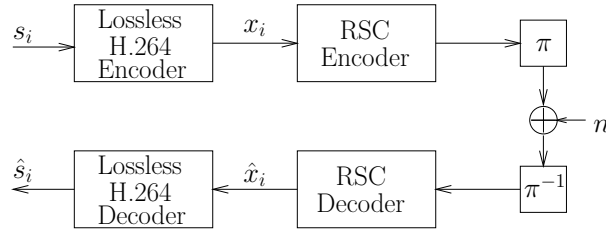


Figure 4.8: Architecture of the Lossless-H.264-RSC system, where the H.264 codec operates in the near-lossless encoding mode.

4.2.4.1 System Configuration

In this section, we consider the scenario of uncompressed video transmission, which may be employed for home networking [168]. The system’s architecture is displayed in Figure 4.7. At time instant i , the transmitter has to convey a video pixel x_i , which is mapped to the m -bit pattern $x_i \binom{m-1}{0}$. This pixel-to-bit mapper may include a quantization function [1]. We treat $(t + 1)$ consecutive and hence correlated m -bit patterns eg. x_0, \dots, x_t as a frame. Let us consider the first 2D video frame for example, which will be interleaved by a bit-based interleaver of length $(t + 1) \cdot m$. Then the signals are RSC encoded and transmitted to the receiver through a Rayleigh channel using BPSK modulation. At the receiver, the demodulator will generate the softbit information y_0, \dots, y_t of the video pixels, which will be input to our proposed ISCD model, namely to the “IHVSM-RSC” and “pixel estimation” blocks of Figure 4.7. Then, the HSMD, VSMD component decoders of our IHVSM and the RSC

	Dim- ension	Bits Num to decode	SI	Code Rate	Delay (frames)	Col (Tr- ans./Rec.)
MMSE-HD	1	8	None	NCR	0	low/low
Unc.-RSC	1	10000	None	$\frac{1}{2} \times \text{NCR}$	0	low/low
L-H.264-RSC	1	10000	None	$\frac{1}{2}$	8	high/low
FOMM-SBSD	1	704	$1 \times \text{MMSTT}$	NCR	0	low/high
FOMM-RSC	1	50688	$1 \times \text{MMSTT}$	$\frac{1}{2} \times \text{NCR}$	0	low/high
IHVSM	2	50688	$1 \times \text{MMSTT}$	NCR	0	low/high
IHVSM-RSC	2	50688	$1 \times \text{MMSTT}$	$\frac{1}{2} \times \text{NCR}$	0	low/high

Table 4.3: Comparison of iterative horizontal-vertical scanline model using recursive systematic convolutional (IHVSM-RSC) code and the benchmarks: hard decoded minimum mean square error (MMSE-HD) [84], Uncompressed-RSC (Unc.-RSC), Lossless-H.264-RSC, first-order Markov modelling aided softbit source decoding (FOMM-SBSD) [87], FOMM-RSC [87], IHVSM, where MMSE estimator is employed.

codec will perform a typical three-stage [41] iterative source-channel decoding for concealing the error effects imposed on the video frames. Specifically, in Figure 4.7 the RSC-HSMD consists of the inner decoder, while the VSMD is the outer decoder. The reconstructed hard-decision-based pixel \hat{x}_i can then be acquired after the pixel estimation stage of Figure 4.7. Although the system employs an RSC channel encoder, for our design-example, arbitrary systematic FEC codecs may be utilized.

We compare the performance of our scheme recorded for the video sequences Akiyo, Coastguard and Football against that of the existing system model. Video sequences stored in (176×144) -pixel quarter common intermediate format (QCIF) or in (352×288) -pixel CIF and 4:2:0 YUV representation are employed. Moreover, the horizontal and vertical decoders perform iterative decoding based on (88×72) -pixel blocks. For the video sequences represented in QCIF, each QCIF luminance frame is divided into (2×2) blocks and each QCIF chroma frame is transmitted in a single block. Note that for simplicity, the uncompressed video bits are transmitted through an uncorrelated non-dispersive Rayleigh channel using BPSK modulation. These parameters are summarized in Table 4.2. The MMSTT of the first-order Markov model was derived from the original video pixels, which may be utilized by the horizontal and vertical source decoders as side-information for improving the



Figure 4.9: Comparison of the first QCIF frames recorded at E_b/N_0 of 12.3 dB, 3.8 dB and 2.5 dB for the Akiyo, Coastguard and Football sequences, decoded by MMSE-HD [84], Uncompressed-RSC, Lossless-H.264-RSC, FOMM-RSC [87] and IHVSM-RSC. MMSE-based pixel estimation is employed. The schematic of Figure 4.7 configured with the parameters in Table 4.2 was used.

achievable error resilience. We mainly rely on two types of curves for characterizing the video quality, namely the PSNR versus the channel SNR per bit, namely E_b/N_0 curves and the bit error ratio (BER) ⁶ versus E_b/N_0 curves.

Shannon's channel capacity theorem [82] was proposed for the transmission of i.i.d source. Hence, to be in line with the channel capacity theory, we have to consider the true entropy of the video sequence, when calculating the energy efficiency per bit. More explicitly, any redundancy inherent in the encoded sequence has to be taken into account by shifting the BER vs E_b/N_0 curves to the right, regardless, whether the redundancy is natural source redundancy or whether it was artificially imposed by channel coding. In our case, substantial redundancy resides in the video source signal, since here we do not employ any video encoder. Hence the true amount of non-redundant information transmitted to the receiver is given by the entropy of the video sequence. Assuming that the total uncompressed size of a video file is S_r bits and the entropy of this video source file is S_e , we might interpret the raw video file as being "naturally" losslessly encoded from S_e i.i.d bits, to generate S_r bits where the code rate is $r = S_e/S_r$. According to Shannon's source coding theorem [82], the entropy of the video source file represents the lowest achievable rate at which the source may be losslessly represented. Hence in our simulations the E_b/N_0 (dB) value

⁶For the uncompressed video scenario, BER defines the bit error ratio of the reconstructed video compared to the original video. For the H.264 compressed bitstream, BER defines the bit error ratio of the bitstream decoded at the receiver.

is calculated as $E_b/N_0 = 10 \log_{10} \frac{E_b S_r}{N_0 S_e}$. However, no widely recognized technique exists for quantifying the entropy of a realistic video source. As a practical solution, we opted for using the near-lossless coding mode of the H.264 codec [1, 14] to encode the source video for the sake of approximating its entropy. The "natural" code rates (NCR) of the Akiyo, Coastguard and Football sequences used in our simulations are listed in Table 4.2. Quantitatively, we found that the "natural" code rates of the three sequences were 1/8.4, 1/2.4 and 1/1.8 for the QCIF scenario, which corresponds to the maximum achievable compression ratios of 8.4, 2.4 and 1.8, respectively. The corresponding parameters for all the scenarios considered are shown in Table 4.2, where the PSNR values correspond to the maximum quality of the error-freely source decoded video at the receiver.

We will benchmark the performance of our system against three schemes, namely against that of the MAP-based and MMSE-based hard decoding schemes (MAP-HD/MMSE-HD), where no softbit source decoding is employed, as well as against the FOMM-SBSD scheme, where no IHVSM decoding is employed. In the latter case only one of the two decoders is activated. A brief comparison of the four schemes is shown in Table 4.3.

4.2.4.2 Benchmarkers

In order to provide sufficiently deep insights into the performance of our proposed system, let us now describe the benchmarkers. Firstly, we benchmark our IHVSM-RSC scheme against the RSC aided uncompressed video transmission system, where no source correlation is exploited at the receiver. We refer to this as Uncompressed-RSC scheme for simplicity.

Then, to analyze the benefits of the RSC codec in the IHVSM-RSC scheme, the results of the IHVSM scheme will also be provided, which is a non-channel-encoded version of the IHVSM-RSC regime. We will also benchmark the performance recorded for the three video sequences against that of the MAP-based and MMSE-based hard decoding schemes (MAP-HD/MMSE-HD), where no source correlation is exploited by the receiver. As further benchmarkers, both the FOMM-SBSD and the FOMM-SBSD aided RSC scheme (FOMM-RSC) were also invoked.

Finally, the system employing the near-lossless H.264 codec of Figure 4.24 is invoked. Specifically, the H.264 codec [14] is configured using the smallest quantization index. Furthermore, both predicted frames (P-frames) and bidirectional predicted frames (B-frames) are enabled. More specifically, the 30-frame sequences were encoded into an intra-coded frame (I-frame), followed by the periodically repeated PBBBBBBB frames. Again, this enables the H.264 codec to generate a near-lossless

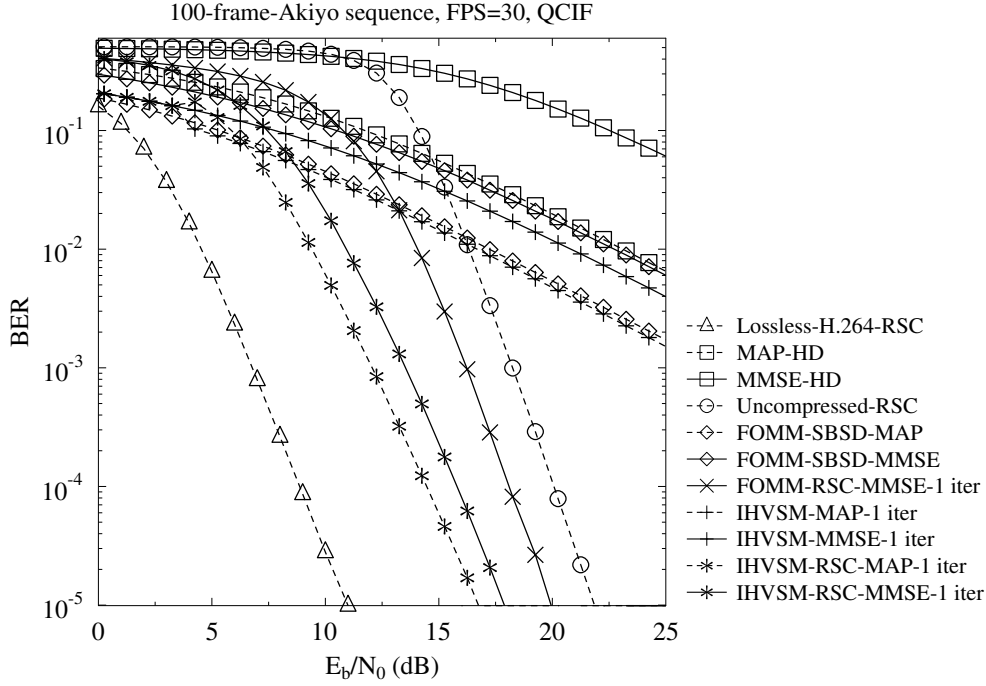


Figure 4.10: BER vs E_b/N_0 comparison of MAP- and MMSE-based pixel estimation for a Rayleigh channel. The schematic of Figure 4.7 configured with the parameters in Table 4.2 was used. *Akiyo*, QCIF

video bitstream. However, a delay of 8 frames was introduced by the employment of B-frames. As shown in Figure 4.24, the RSC codec of the IHVSM-RSC scheme was utilized as our FEC codec for protecting the losslessly encoded bitstream. We refer to this system as the Lossless-H.264-RSC arrangement for simplicity. Note that the Lossless-H.264-RSC system imposes a high complexity at the transmitter, but a low complexity at the receiver. By contrast, our system imposes low complexity at the transmitter and a high complexity at the receiver.

A brief comparison of all the benchmark schemes is shown in Table 4.3, where L-H.264-RSC represents the Lossless-H.264-RSC scheme. The Row Col (Trans./Rec.) compares the complexity imposed at the transmitter and receiver, respectively.

4.2.4.3 Numerical Results

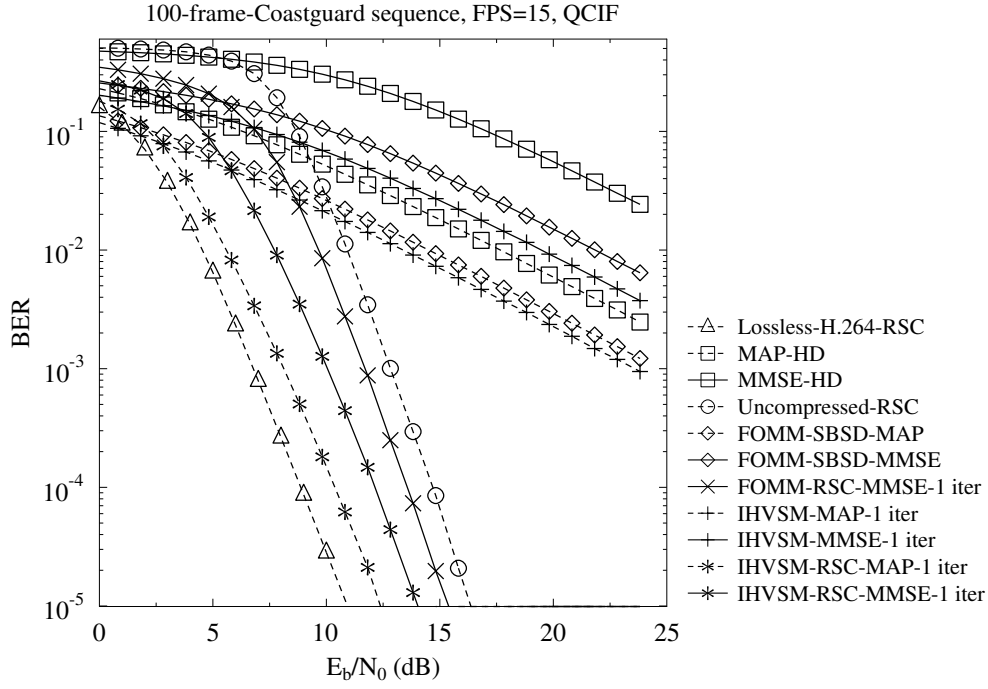


Figure 4.11: BER vs E_b/N_0 comparison of MAP- and MMSE-based pixel estimation for a Rayleigh channel. The schematic of Figure 4.7 configured with the parameters in Table 4.2 was used. *Coastguard*, QCIF

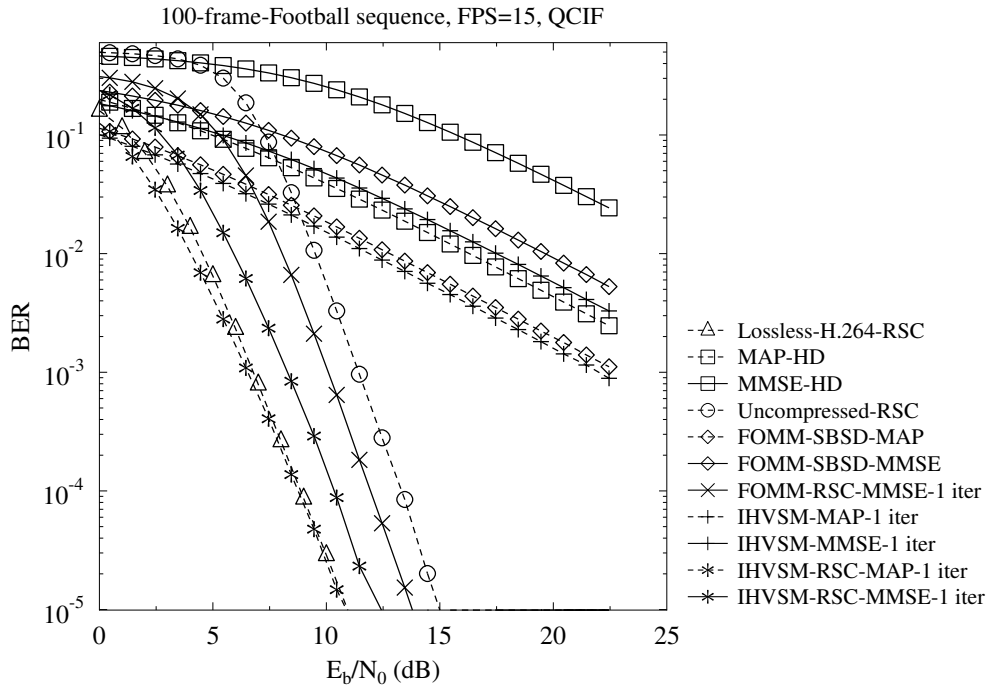


Figure 4.12: BER vs E_b/N_0 comparison of MAP- and MMSE-based pixel estimation for a Rayleigh channel. The schematic of Figure 4.7 configured with the parameters in Table 4.2 was used. *Football*, QCIF

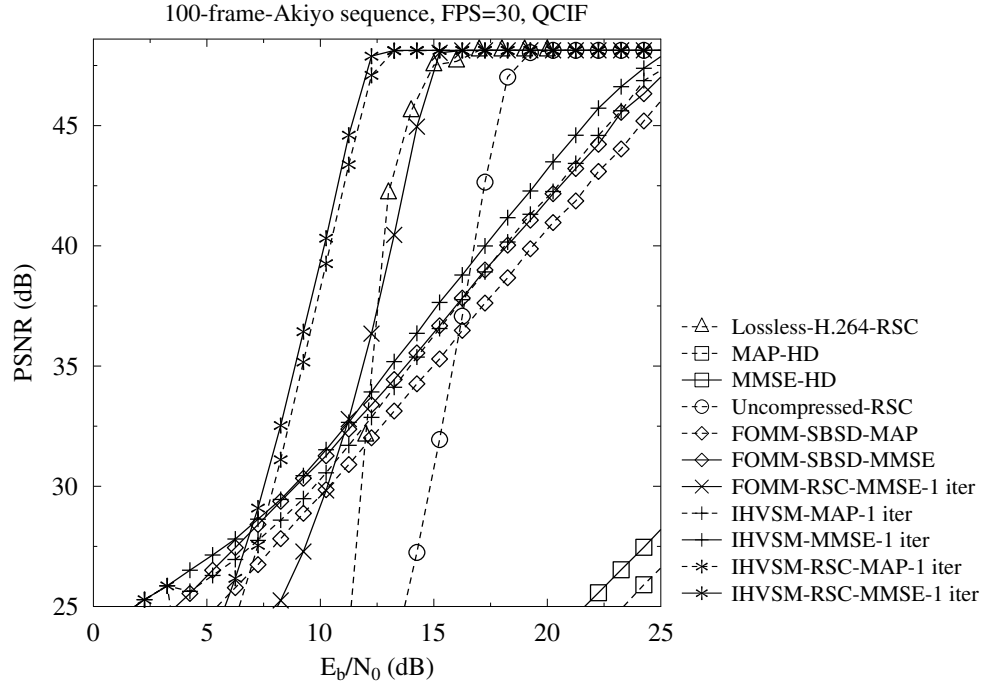


Figure 4.13: PSNR vs E_b/N_0 comparison of MAP- and MMSE-based pixel estimation for a Rayleigh channel. The schematic of Figure 4.7 configured with the parameters in Table 4.2 was used. *Akiyo*, QCIF

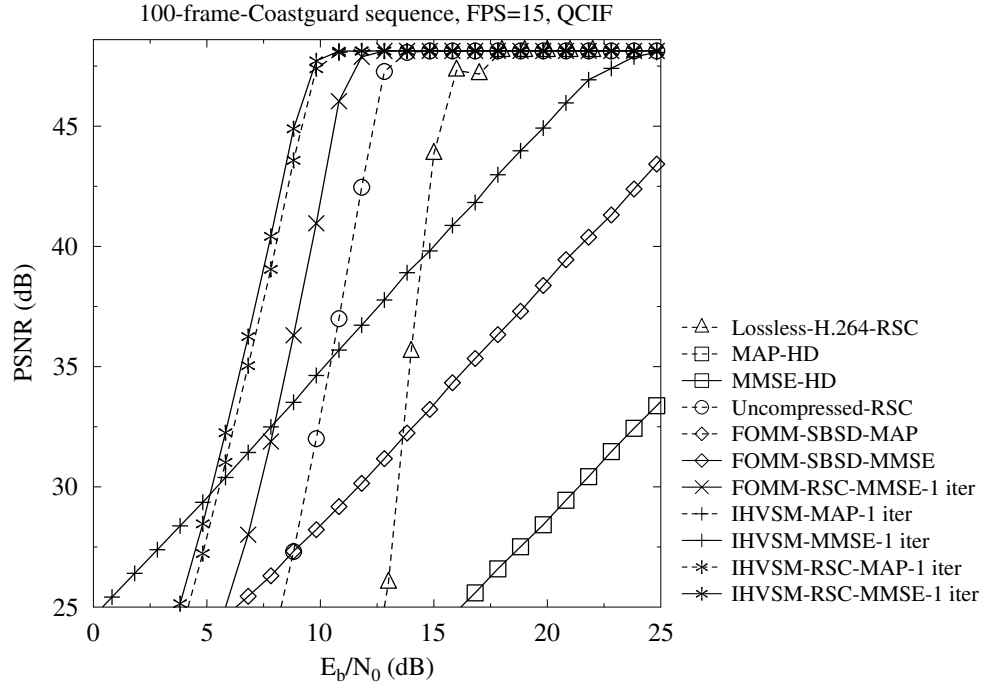


Figure 4.14: PSNR vs E_b/N_0 comparison of MAP- and MMSE-based pixel estimation for a Rayleigh channel. The schematic of Figure 4.7 configured with the parameters in Table 4.2 was used. *Coastguard*, QCIF

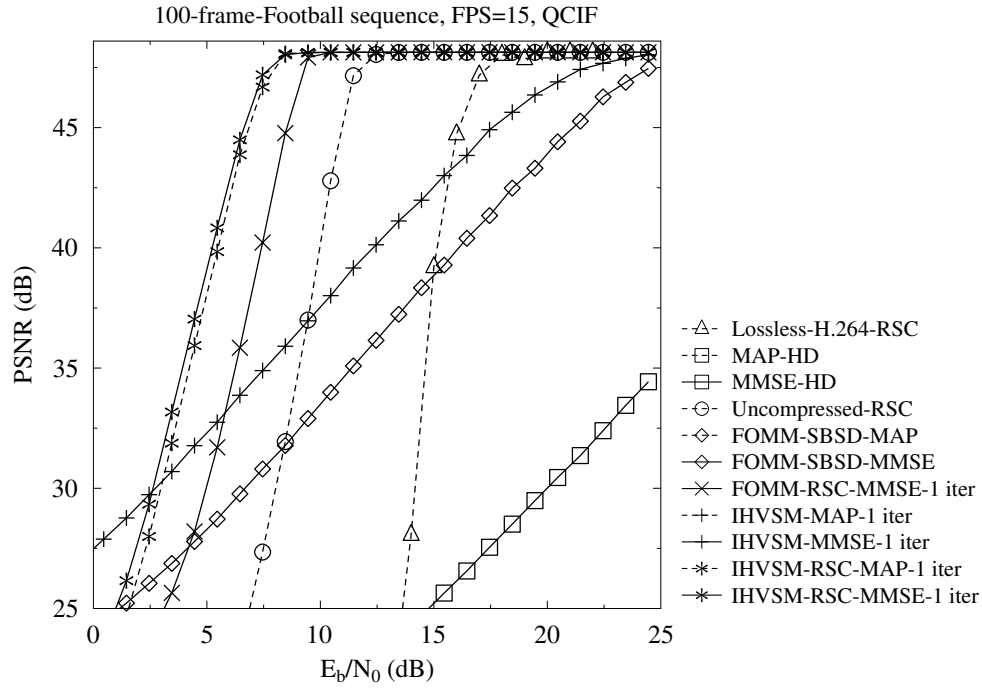


Figure 4.15: PSNR vs E_b/N_0 comparison of MAP- and MMSE-based pixel estimation for a Rayleigh channel. The schematic of Figure 4.7 configured with the parameters in Table 4.2 was used. *Football*, QCIF

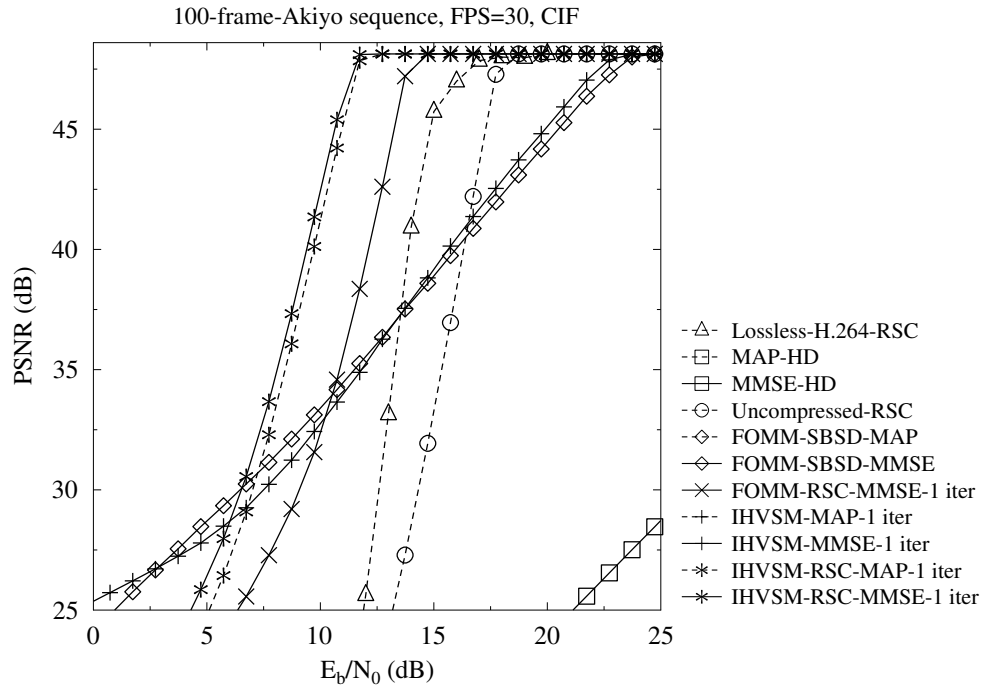


Figure 4.16: PSNR vs E_b/N_0 comparison of MAP- and MMSE-based pixel estimation for a Rayleigh channel. The schematic of Figure 4.7 configured with the parameters in Table 4.2 was used. *Akiyo*, CIF

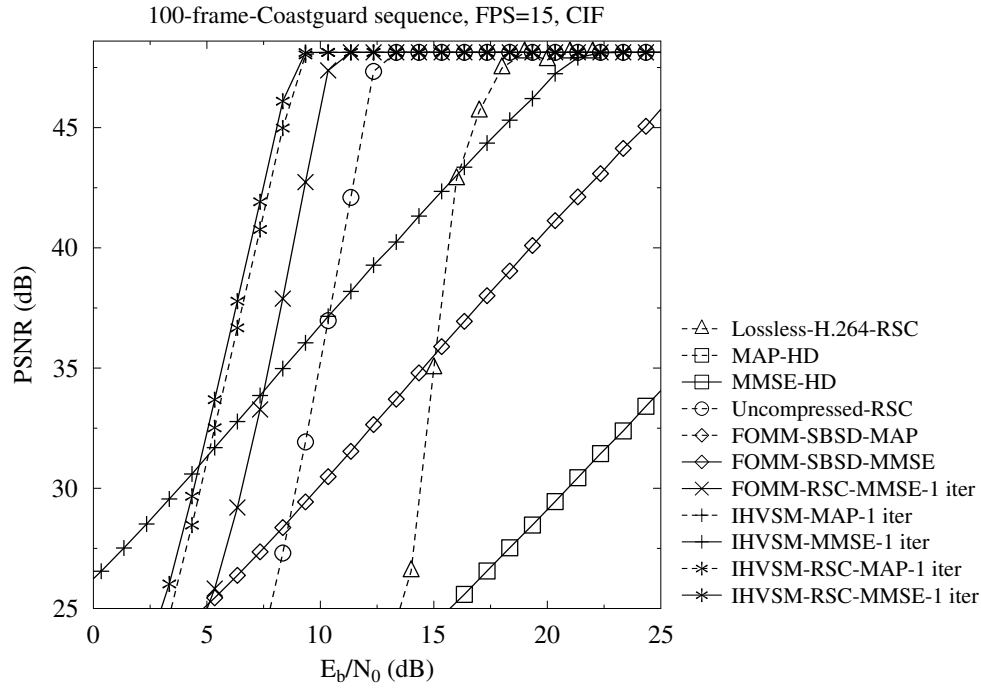


Figure 4.17: PSNR vs E_b/N_0 comparison of MAP- and MMSE-based pixel estimation for a Rayleigh channel. The schematic of Figure 4.7 configured with the parameters in Table 4.2 was used. *Coastguard*, CIF

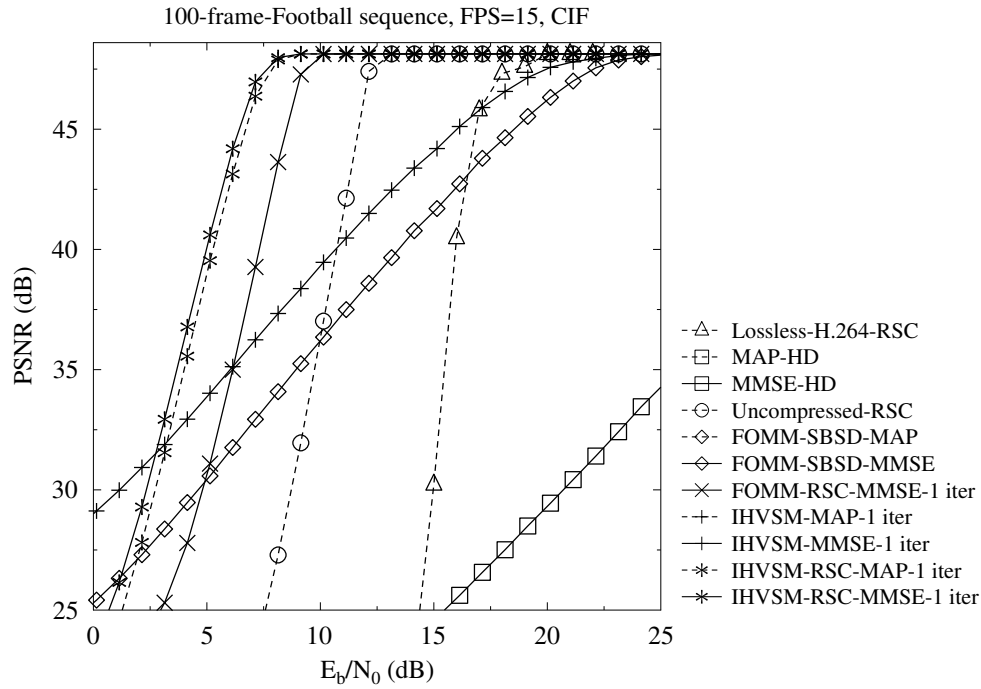


Figure 4.18: PSNR vs E_b/N_0 comparison of MAP- and MMSE-based pixel estimation for a Rayleigh channel. The schematic of Figure 4.7 configured with the parameters in Table 4.2 was used. *Football*, CIF

Firstly, the BER versus E_b/N_0 performance comparison of the MAP based and MMSE based pixel estimation using the Akiyo-QCIF sequence is presented in Figure 4.10, while the corresponding Y-PSNR versus E_b/N_0 results are presented in Figure 4.13. As suggested by Figure 4.10, the IHVSM-RSC scheme substantially outperforms both the FOMM-SBSD and the Uncompressed-RSC schemes. Specifically, at a BER of 10^{-5} the IHVSM-RSC-MMSE system outperforms the FOMM-RSC-MMSE and the Uncompressed-RSC systems by about 2 dB and 4.6 dB, respectively. Moreover, even though the Lossless-H.264-RSC achieves the best BER performance, the compressed bits are rather sensitive to bit errors. The BER versus E_b/N_0 performance comparison of the MAP based and MMSE based pixel estimation using the Coastguard-QCIF and Football-QCIF sequences is presented in Figures 4.11 and 4.12, respectively. Similar trends to those seen in Figure 4.10 are observed. As suggested by Figure 4.13., at a PSNR of 47.5 dB the IHVSM-RSC-MMSE system outperforms the FOMM-RSC-MMSE, the Uncompressed-RSC and the Lossless-H.264-RSC systems by about 2.8 dB, 6.6 dB and 2.8 dB, respectively. In other words, at E_b/N_0 of 12.1 dB the IHVSM-RSC-MMSE system outperforms the FOMM-RSC-MMSE and the Lossless-H.264-RSC systems by about 12.1 dB and 15 dB, in terms of the video PSNR achieved respectively.

Observe from Figure 4.10 that the MAP-based systems outperform the schemes employing the MMSE estimator in terms of the BER attained. On the other hand, Figure 4.13 suggests that the MMSE-based systems outperform the schemes employing the MAP estimator in terms of the PSNR attained. We may conclude that the MAP estimator is capable of achieving a lower BER, while the MMSE estimator may achieve an increased PSNR. Hence, to achieve an increased PSNR, the MMSE based pixel estimation should be employed.

To provide further insights, we present PSNR versus E_b/N_0 results in Figures 4.14 and 4.15 for the high-dynamic Coastguard-QCIF and Football-QCIF sequences, respectively. Again, the MMSE based estimator is capable of achieving a higher PSNR than the MAP based estimator. We mainly present the Y-PSNR versus E_b/N_0 curves in Figures 4.14 and 4.15 using the MMSE based estimator. As suggested by Figures 4.14 and 4.15, the IHVSM-RSC-MMSE substantially outperforms the FOMM-RSC-MMSE, the Uncompressed-RSC and the Lossless-H.264-RSC schemes. Specifically, at a Y-PSNR of 47.5 dB the IHVSM-RSC-MMSE outperforms the FOMM-RSC-MMSE, the Uncompressed-RSC and the Lossless-H.264-RSC schemes by about 1.8 dB, 3.3 dB and 6.4 dB, respectively, when considering the Coastguard-QCIF sequence. Similar trends are observed for the Football-QCIF sequence. Our visual comparisons of

the decoded frames for the Akiyo, Coastguard and Football sequences are shown in Figure 4.9.

The corresponding PSNR vs E_b/N_0 results recorded for CIF-sized video clips are presented in Figures 4.16, 4.17 and 4.18 for the Akiyo, Coastguard and Football sequences, respectively. Similar to the trends observed in Figures 4.13, 4.14 and 4.15, Figures 4.16 to 4.18 suggest that the IHVSM-RSC-MMSE arrangement substantially outperforms the FOMM-RSC-MMSE, the Uncompressed-RSC and the Lossless-H.264-RSC schemes.

4.3 Spatio-Temporal Iterative Source-Channel Decoding

4.3.1 System Overview

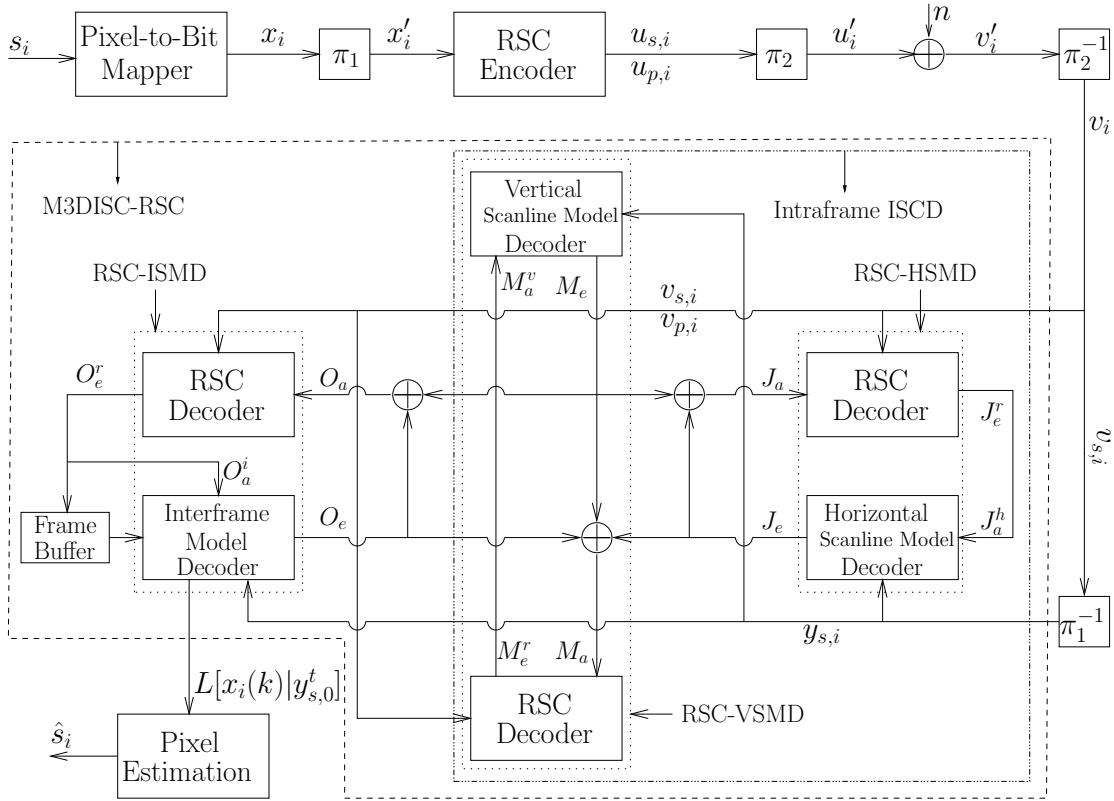


Figure 4.19: System architecture of the M3DISC-RSC, where R represents reordering of the video pixels, while π represents the bit-interleaver.

A one-dimensional iterative system model was proposed and analyzed in [84,83,98, 87] in the context of audio signals. In [184], we conceived a system for iterative source-and-channel concealment in two-dimensional video frames, which exploited the intra-frame correlation of practical video signals. In this section, we will detail our system model designed for ISCD exploiting the 3D correlation inherent in uncompressed video streaming. The system model of 3D iterative ISCD is displayed in Figure 4.19,

where R represents reordering of the video pixels, while π denotes the bit-interleaver. Assuming that our algorithm performs soft decoding on a (8×8) -pixel macroblock, all the soft pixels are ordered into 8 horizontal soft Markov processes, each of which will be input into a horizontal scanline model decoder. Then these 8 horizontal soft Markov processes will be reordered into 8 vertical soft Markov processes, each of which consists of 8 soft pixels and will be input into a vertical scanline model decoder. Note that the dimension of the video is the only parameter of the reordering process, hence given this parameter, the receiver can readily carry out the reordering. More details of the reordering are provided in [184]. The subscripts “s,” “p” denote the systematic and parity bits of a coded symbol, respectively. For instance, a systematic convolutional coded (RSC) symbol v_i consists of the systematic component $u_{s,i}$ and the parity component $u_{p,i}$. The subscripts “a,” “e” represent the *a-priori* extrinsic information, respectively, whereas “J,” “M,” and “O” indicate their relevance to the inner, middle and outer decoders, respectively. Furthermore, the superscripts “h,” “v,” “r,” and “i” indicate that the information is related to the horizontal scanline model decoder, vertical scanline model decoder, the RSC and the inter-frame model decoder, respectively. We will further detail the system model below.

4.3.1.1 Transmitter

Since we consider uncompressed video communication in this treatise, we do not employ any video encoder at the transmitter. Hence, the uncompressed video pixels are simply converted to bits by the pixel-to-bit mapper shown in Figure 4.19. Then the uncompressed bits are encoded by a RSC code. Specifically, at time instant i , the transmitter has to convey a video pixel s_i , which is mapped into the bit pattern x_i . This pixel-to-bit mapper may include the classic quantization operation [1]. Let us now assume that the m -bit pattern x_i is expressed as $\{x_i(0), \dots, x_i(m-1)\} = x_i \binom{m-1}{0}$ as well as that N consecutive and hence highly correlated m -bit patterns, such as x_0, \dots, x_{N-1} may be treated as a frame. Consider the first 2D video frame for example, which is mapped to a bit sequence $x'_0 \binom{m-1}{0}, \dots, x'_{N-1} \binom{m-1}{0}$ by a bit-based interleaver of length $N \cdot m$. Then the interleaved bit sequence is encoded by a channel codec, where a RSC is employed in our case. After this stage, the RSC coded bitstream consisting of the systematic bit sequence $u_{s,i}$ and the parity bit sequence $u_{p,i}$, will be interleaved by another interleaver before transmission over the channel. Then the signals are transmitted to the receiver through a Rayleigh channel using BPSK modulation. Note that the transmitter scans and transmits each video frame on a block by block basis, and similarly the receiver will reconstruct each video frame on a block basis.

4.3.1.2 Receiver

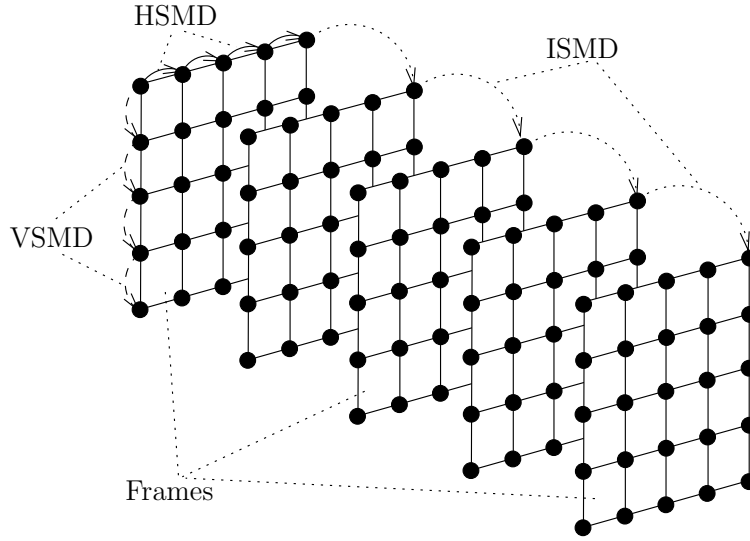


Figure 4.20: Exemplified Markov processes of HSMD, VSMD and ISMD in Figure 4.19 using five consecutive (5×5) -sized frames.

At the receiver, the softbit source decoding [84] principle is employed for mitigating the effects of the error-infested bit sequence $v_0 \binom{m-1}{0}, \dots, v_{N-1} \binom{m-1}{0}$, which is deinterleaved from the received bit sequence $v'_0 \binom{m-1}{0}, \dots, v'_{N-1} \binom{m-1}{0}$. Two decoding stages are involved in the softbit source channel decoding process, namely the M3DISC-RSC and the related pixel estimation. Firstly the received signal $v'_{s,i}$ will be deinterleaved at the receiver to generate the bit sequence $v_{s,i}$ and $v_{p,i}$, where $v_{s,i}$ indicates the systematic part of the deinterleaved bit sequence, while $v_{p,i}$ indicates the parity part. Then the systematic information $v_{s,i}$ will be further deinterleaved by interleaver π_1 ⁷, hence the error-infested version of signal x_i namely $y_{s,i}$ can be obtained, as seen in Figure 4.19.

At the first decoding stage of Figure 4.19 four decoders are employed, namely the RSC channel decoder as well as the three source decoders, the Horizontal Scanline Model Decoder (HSMD) operating in the horizontal direction, the Vertical Scanline Model Decoder (VSMD) proceeding in the vertical direction and the Inter-frame Scanline Model Decoder (ISMD). The Markov processes of HSMD, VSMD and ISMD are exemplified in Figure 4.20, where five consecutive (5×5) -sized frames are employed. However, each of the source decoders is paired with the RSC thereby forming three decoder pairs, namely the RSC-HSMD, the RSC-VSMD and the RSC-ISMD decoder pairs as seen in Figure 4.19. The reason for this design is that the three source decoders jointly improve the attainable video peak signal-to-noise ratio (PSNR) owing to the improved bit error ratio (BER), while this was not possible in the previously

⁷In Figure 4.19, the reordering and interleaving operations are ignored for the sake of simplifying our system architecture, which are straightforward to add.

proposed iterative EC system using the Iterative Horizontal-Vertical Scanline Model (IHVSM) of [184]. Hence the RSC codec is employed for providing increasingly improved extrinsic information from one source decoder to another for reducing the BER and hence improving the PSNR. Furthermore, we choose the RSC codec as the channel decoder, since the systematic information bits namely $v_{s,i}$ of Figure 4.19 may be readily exploited by the source decoders. However, arbitrary channel codecs may also be employed. In our system, the three decoder pairs of Figure 4.19 are treated as three amalgamated decoders for the sake of performing the three-stage decoding [41], while a certain number of iterations is performed between the two decoders within each decoder pair to generate the extrinsic information during the integrated three-stage decoding process. Note that in our scenario, both the systematic part $v_{s,i}$ and the parity part $v_{p,i}$ can be directly exploited by the RSC decoder, while the three source decoders can only directly utilize the deinterleaved systematic information $y_{s,i}$.

Let us now continue by detailing the decoding process at the receiver by assuming the following scenarios:

- The f^{th} frame is being transmitted, which implies that the $(f - 1)$ th frame has already been received.
- H horizontal scanlines and V vertical scanlines of the current frame have been received. This is equivalent to saying that a $(H \times V)$ -line block of the f^{th} frame has been received, which is represented by the $(H \cdot V)$ -bit patterns $v_{s,i}/y_{s,i}$, $v_{p,i}$.

Two different types of iterative decoding processes are invoked at the receiver, namely the iterative decoding process within the amalgamated three decoder pairs and the iterative decoding process exchanging extrinsic information among the three integrated decoder pairs. Below we now detail them separately.

4.3.1.2.1 Iterative Decoding within the Decoder Pairs Again there are three amalgamated decoder-pairs in Figure 4.19, where the RSC components accept *a-priori* information from the other two decoder-pairs, while the source decoders are responsible for generating the extrinsic information. The inner RSC-HSMD decoder pair of Figure 4.19 performs decoding on a block of $(H \cdot V)$ bit-patterns, namely H horizontal and V vertical scanlines. The RSC codec accepts both the systematic information $v_{s,i}$ and the parity information $v_{p,i}$ of the $(H \cdot V)$ -bit patterns, which are deinterleaved by the interleaver π_2 of Figure 4.19. The HSMD source decoder accepts the systematic information $y_{s,i}$ as its input, which is deinterleaved from $v_{s,i}$ by the interleaver π_1 . For each decoding iteration within a specific decoder-pair, the RSC decoder will take both the channel information v_i and the *a-priori* information

J_a of Figure 4.19 as its input to generate the extrinsic information J_e^r , which will be deinterleaved by interleaver π_1 . Then the deinterleaved extrinsic information will be reordered by R_1 to generate H horizontal scanlines of *a-priori* LLR information J_a^h , which can be exploited by the HSMD. Conversely, the extrinsic information J_e generated by the HSMD in the format of the H horizontal scanlines will be reordered and deinterleaved for further exploitation by the RSC, as a part of the *a-priori* information. Finally, after a preset number of iterations exchanging extrinsic information between the RSC and the HSMD, the extrinsic information J_e generated by the HSMD will be output as the extrinsic information of the inner decoder-pair as seen in Figure 4.19. Similarly, the middle RSC-VSMD decoder pair of Figure 4.19 performs decoding on a block of $(H \cdot V)$ bit-patterns. However, reordering is employed for the extrinsic information between the RSC and the VSMD. In contrast to the inner and intermediate decoder pairs, the RSC of the outer decoder pair of Figure 4.19 performs decoding on a block of $(H \cdot V)$ bit-patterns, while the ISMD performs decoding on f -bit patterns, which are from the same frame position of f consecutive video frames. Firstly, the RSC takes the extrinsic information O_a from the ISMD and the intermediate decoder pair of Figure 4.19 as *a-priori* information to generate the extrinsic information O_e^r , which will be deinterleaved by interleaver π_1 and reordered by R_3 . Then the reordered information will be stored in the “frame buffer” of Figure 4.19, which will output $H \cdot V$ independent scanlines for the f consecutive frames. Each of the scanlines carries f -bit patterns, which obey a Markov process and this is exploited by the ISMD in order to perform decoding. Note that f is a flexible system parameter, which depends on the particular application. Specifically, for non-realtime video streaming applications, f may be set to a higher number, which will induce a maximal delay of $(f - 1)$ video frames. However, for realtime applications, only a more limited range of the previously received frames may be utilized for decoding the current frame. Finally, after a certain number of iterative decoding iterations, the extrinsic information generated by the ISMD will be output as the extrinsic information O_e generated by this particular decoder-pair.

4.3.1.2.2 Iterative Decoding Exchanging Information Among Decoder Pairs

When considering iterative decoding exchanging information among the three decoder-pairs of Figure 4.19, each source-channel decoder pair will be treated as an amalgamated decoder. Generally, the extrinsic information exchange rules follow the three stage decoding rules stated in [41]. The inner source-channel decoder iterations exploit the intra-frame correlation within a video frame, while the outer decoder exploits the inter-frame correlation. As seen in Figure 4.19 there are two inputs to

the inner RSC-HSMD decoder pair, namely the channel information v_i and the *a-priori* information $J_a = J_e + M_e$, where M_e is the extrinsic information generated from the intermediate RSC-VSMD decoder pair, while J_e is generated by the inner decoder-pair itself. By contrast, the *a-priori* information forwarded to the intermediate RSC-VSMD decoder pair can be expressed as $M_a = M_e + J_e + O_e$, where O_e is the extrinsic information generated from the outer RSC-ISMD decoder pair. Similarly, the *a-priori* information provided for the outer RSC-ISMD decoder-pair of Figure 4.19 may be expressed as $O_a = O_e + M_e$. Note that the extrinsic information must be appropriately deinterleaved and reordered to be exploited as the *a-priori* information by another decoder. However, in order to simplify our discussions, we ignored this interleaver and reordering operation in the above expressions. The final *a-posteriori* information may be generated by a hard-decision at the output of the outer decoder-pair of Figure 4.19.

As a matter of fact, more extrinsic information may be gleaned from any of the six decoders of Figure 4.19. For example, J_e^r can also be utilized by VSMD as part of the *a-priori* information. However, since M_e^r generated by the RSC of the intermediate decoder pair may be viewed as a more reliable version of J_e^r , J_e^r was excluded from the *a-priori* information provided for the VSMD.

In the discussions of Section 4.3.1.2.1 and Section 4.3.1.2.2, a number of extrinsic information exchanges are involved, namely J_a , J_e , J_e^r , J_a^h , M_a , M_e , M_e^r , M_a^v , O_a , O_e , O_e^r , O_a^i . Among them, we have $J_a^h = J_e^r$, $M_a^v = M_e^r$ and $O_a^i = O_e^r$, which are generated by the RSC decoder [183]. The extrinsic information terms J_e , M_e and O_e are generated by the horizontal, vertical and inter-frame model decoder, respectively, whose extrinsic information derivation rule is given by Eq. (4.10). After the first stage decoding, the relevant *a-posteriori* LLR information $L[x_i(k)|y_{s,0}^t]$ is generated, which may be exploited by either the bit-based MMSE or the bit-based MAP estimator for estimating the m -bit pattern x_i as well as for outputting the original pixel \hat{s}_i at the parameter estimation stage, which may be formulated as [84, 184]

- MAP estimator

$$\hat{x}_i = \arg \max_{\forall x_i \in X_m} \prod_{k=0}^{m-1} p[x_i(k)|y_0^t]; \quad (4.22)$$

- MMSE estimator

$$\hat{x}_i = \sum_{k=0}^{m-1} 2^k \cdot p[x_i(k) = 1|y_0^t]. \quad (4.23)$$

Finally, the original video source pixel \hat{s}_i may be obtained from the estimated m -bit pattern \hat{x}_i by using the inverse operations of the source encoder.

4.3.2 Performance Analysis

In this section, we will analyze the performance of our proposed system introduced in Section 4.3.1. Firstly, in Section 4.3.2.1 we will introduce the scenario considered in our experiments. Then our EXIT-chart analysis will be presented in Section 4.3.2.2, followed by a couple of benchmarkers discussed in Section 4.3.2.3. Finally, we will benchmark the performance of our system.

4.3.2.1 Scenario

	Akiyo	Foreman	Coastguard
Representation	YUV 4:2:0	YUV 4:2:0	YUV 4:2:0
Format	QCIF	QCIF	QCIF
Bits Per Pixel	8	8	8
FPS	30	15	15
Number of Frames	30	30	30
Bitrate	524 kbps	1579 kbps	1924 kbps
“Natural” Code Rate	1/8.7	1/2.89	1/2.37

Table 4.4: Features of the video sequences, Akiyo, Foreman and Coastguard used for the system seen in Figure 4.19.

Generator of RSC	[11,13,13,15]	Modulation	BPSK
Channel Code Rate	1/2	Pair Iteration	2
Channel	Unc-Ray	Inner Iteration	2

Table 4.5: Table of parameters employed for simulations of the system seen in Figure 4.19. *Unc-Ray* stands for Uncorrelated Rayleigh.

In this section, we present our experimental parameters used for characterizing the convergence behavior of and benchmarking the proposed M3DISC-RSC scheme introduced in Section 4.3.1. Three 30-frame video sequences, Akiyo, Foreman and Coastguard, represented in (176×144) -pixel quarter common intermediate format (QCIF) and 4:2:0 YUV representation are employed. Moreover, the iterative intra-frame EC scheme of Figure 4.19 operates on the basis of (8×8) -pixel blocks. Each QCIF luminance frame is divided into (22×18) 8×8 -pixel blocks and each QCIF chroma frame is divided into (11×9) 8×8 -pixel blocks. The uncompressed video bits are transmitted through an uncorrelated non-dispersive Rayleigh channel using BPSK modulation. We employ a RSC encoder having a rate of $R = 1/2$ and generator polynomials of $g_1 = 1011$, $g_2 = 1101$, $g_3 = 1101$, $g_4 = 1111$, which are represented as $G = [1, g_2/g_1, g_3/g_1, g_4/g_1]$, where g_1 is a feedback input and g_2, g_3, g_4 are the feedforward outputs. Moreover, the puncturing matrix $[10; 01; 10; 01]$ is employed, where the four rows correspond to the output of the systematic bit of the g_2, g_3 and g_4 , respectively. For the 3D iterative decoding, two iterations are employed for iterative decoding within each decoder-pair is employed, as well as for the inner/intra-frame iterative information exchange between decoder pairs. The parameters employed are listed in Table 4.4 and Table 4.5. The parameters of the first-order Markov model MMSTTs were trained using the original video sequences according to the process detailed in Section 4.2.3.5, which are utilized by the HSMD, VSMD and ISMD for improving the achievable error resilience.

In our simulations, the 'natural' code rate detailed in Section 4.2.4.1, the 'natural' code rate principle is introduced for calculating the the true entropy of the video sequence. Specifically, the E_b/N_0 (dB) value is calculated as

$$E_b/N_0 = 10 \log_{10} \frac{E_b S_r}{N_0 S_e}. \quad (4.24)$$

The “*natural*” code rates of the three video sequences used in our simulations are listed in Table 4.4. Quantitatively, we found that the “*natural*” code rates (NCR) of the Akiyo, Foreman and Coastguard clips were $1/8.7$, $1/2.89$ and $1/2.37$, respectively for the scenario considered, which corresponds to the maximum lossless compression ratios of 8.7, 2.89 and 2.37.

4.3.2.2 Three-Dimensional EXIT Charts

In this section, we characterize the convergence behavior of the proposed M3DISC-RSC scheme introduced in Section 4.3.1 using the Akiyo sequence. For the ISMD, we divide the 30 frames of Akiyo into two $f = 15$ -frame groups for decoding, which

imposes a maximal delay of $(f - 1)/FPS = 933$ millisecond (ms)⁸ in video transmission.

For the sake of analyzing the iterative decoding convergence of the M3DISC-RSC scheme seen in Figure 4.19, each of the three source-channel decoder-pairs is treated as an integrated decoder component. Again, the channel information v_i can be exploited by all of the three decoder-pairs. Hence all the EXIT functions of the inner, intermediate and outer decoder-pairs depend on E_b/N_0 , which can be expressed as $I(J_e) = f_j[I(\tilde{J}_e), I(M_e), E_b/N_0]$, $I(M_e) = f_m[I(J_e), I(\tilde{M}_e), I(O_e), E_b/N_0]$ and $I(O_e) = f_o[I(M_e), I(\tilde{O}_e), E_b/N_0]$ [132, 156], respectively. Furthermore, \tilde{J}_e , \tilde{M}_e and \tilde{O}_e indicate the relevant extrinsic information previously generated by the inner, intermediate and outer decoder-pairs, respectively. However, in order to draw the EXIT functions in the three-dimensional (3D) space, we approximate the EXIT functions as $I(J_e) = f_j[0, I(M_e), E_b/N_0]$, $I(M_e) = f_m[I(J_e), 0, I(O_e), E_b/N_0]$ and $I(O_e) = f_o[I(M_e), 0, E_b/N_0]$ in the simulations.

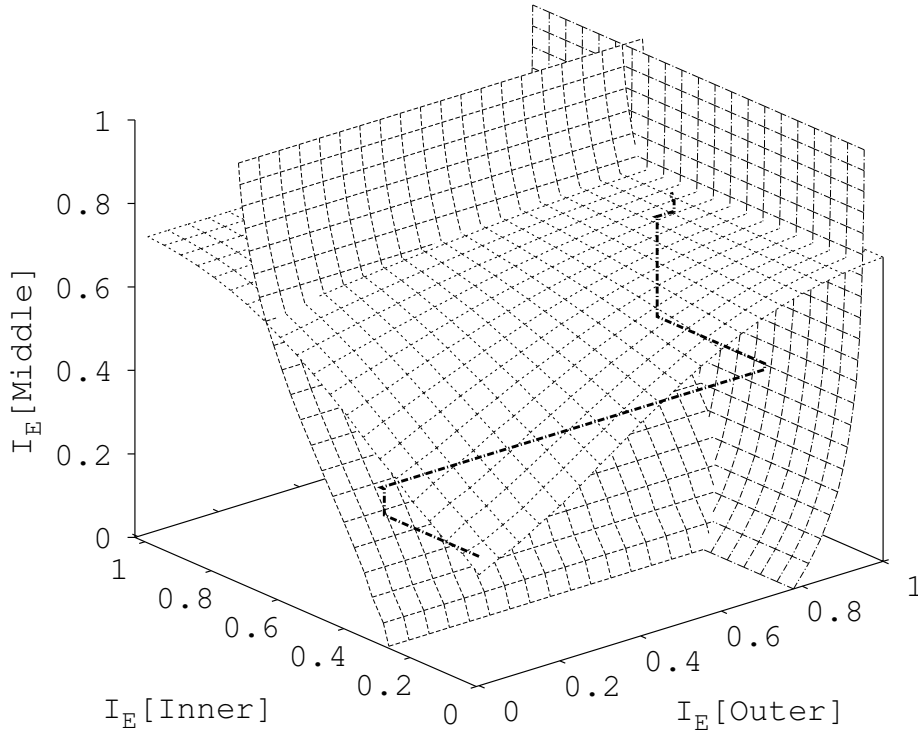


Figure 4.21: Three dimensional EXIT chart of the M3DISC-RSC, when communicating over uncorrelated Rayleigh channel with E_b/N_0 of 5.4 dB. The schematic of Figure 4.19 configured with the parameters in Tables 4.4 and 4.5 was used.

⁸This delay of ~ 1 sec is unsuitable for lip-synchronized interactive video applications. However in Section 4.3.2.5, we will optimize our system and characterize its performance for reduced delays.

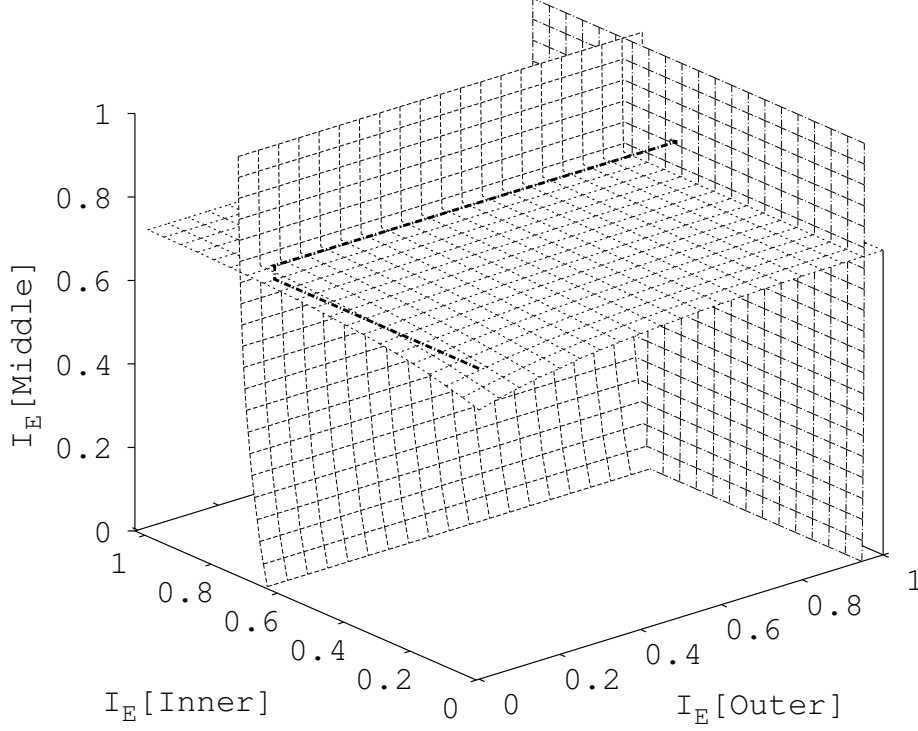


Figure 4.22: Three dimensional EXIT chart of M3DISC-RSC, when communicating over an uncorrelated Rayleigh channel at $E_b/N_0 = 9.4$ dB. The schematic of Figure 4.19 configured with the parameters in Tables 4.4 and 4.5 was used.

We present the 3D EXIT charts recorded at the E_b/N_0 values of 5.4 dB and 9.4 dB, which are shown in Figure 4.21 and Figure 4.22, respectively. Observe from the figures that the inner and intermediate decoders generate similar mutual information (MI) with the aid of the same *a-priori* MI, which is due to the fact that the horizontal and vertical scanlines carry similar amount of correlations. The outer decoder generates substantially higher MI than the inner and the middle decoders at the same *a-priori* MI, since a higher amount of correlation is associated with the consecutive frames within a intra-frame. Observe from Figure 4.21 and Figure 4.22 that the three EXIT surfaces intersect at the points (0.94,0.62,0.73) and (0.95,0.69,0.73), respectively. Hence, the Monte-Carlo simulation based decoding trajectory is unable to reach the point (1,1,1) at an E_b/N_0 of 5.4 or 9.4 dB. However, we will demonstrate in Section 4.3.2.4 that we can still attain a high video quality at $E_b/N_0 = 9.4$ dB despite having a non-negligible BER.

4.3.2.3 Benchmarks

In order to provide sufficiently deep insights on the performance of our proposed system, let us now describe the benchmarks.

Firstly, we benchmark our M3DISC-RSC scheme against the RSC aided uncompressed video transmission system, where no source correlation is exploited at the receiver. We refer to this as RSC scheme for simplicity.

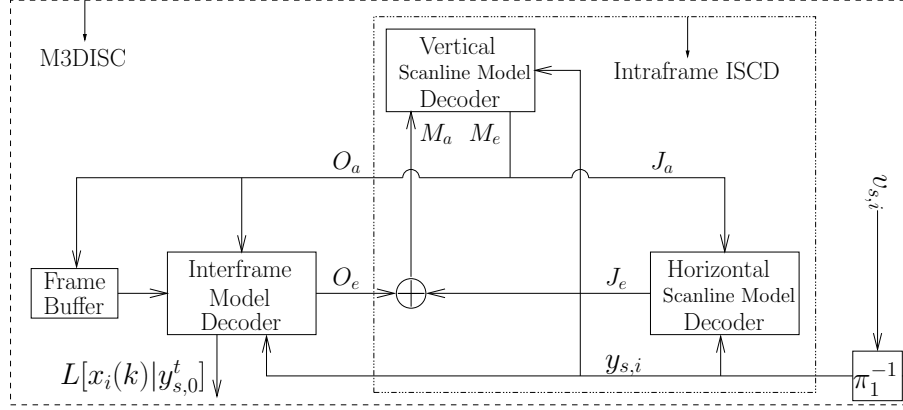


Figure 4.23: Architecture of the M3DISC, where no-channel encoder is employed.

Then, to analyze the benefits of the RSC codec in the M3DISC-RSC scheme, we will benchmark it against the M3DISC arrangement, which is a non-channel-encoded version of the M3DISC-RSC regime. The architecture of the M3DISC scheme is portrayed in Figure 4.23. We will also benchmark the performance recorded for the three video sequences against that of the MMSE-based hard decoder (MMSE-HD) [184], where no source correlation is exploited by the receiver. As further benchmarkers, both the first-order Markov modeling based SBS (FOMM-SBS) relying on a one-dimensional Markov process is employed at the receiver and the IHVSM of [184] were also invoked.

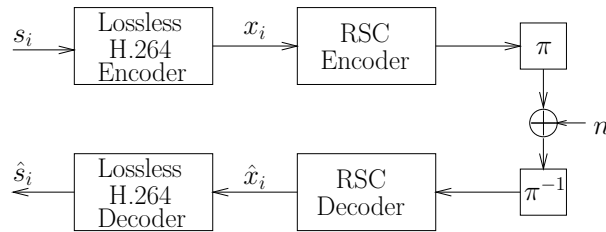


Figure 4.24: Architecture of the Lossless-H.264-RSC system, where the H.264 codec operates in the near-lossless encoding mode.

Finally, the system employing the near-lossless H.264 codec of Figure 4.24 is invoked. Specifically, the H.264 codec [14] is configured using the smallest quantization index. Furthermore, both predicted frames (P-frame) and bidirectional predicted (B-frames) are enabled. More specifically, the 30-frame sequences were encoded into an intra-coded frame (I-frame), followed by the periodically repeated PBBBBBBB frames. Again, this enables the H.264 codec to generate a near-lossless video bit-stream. However, a delay of 8 frames delay was introduced by the employment of

B-frames. As shown in Figure 4.24, the SC codec of the M3DISC-RSC scheme was utilized as our FEC codec for protecting the losslessly encoded bitstream. We refer to this system as the Lossless-H.264-RSC arrangement for simplicity. Note that the Lossless-H.264-RSC system imposes a high complexity at the transmitter, but a low complexity at the receiver. By contrast, our system imposes low complexity at the transmitter and a high complexity at the receiver.

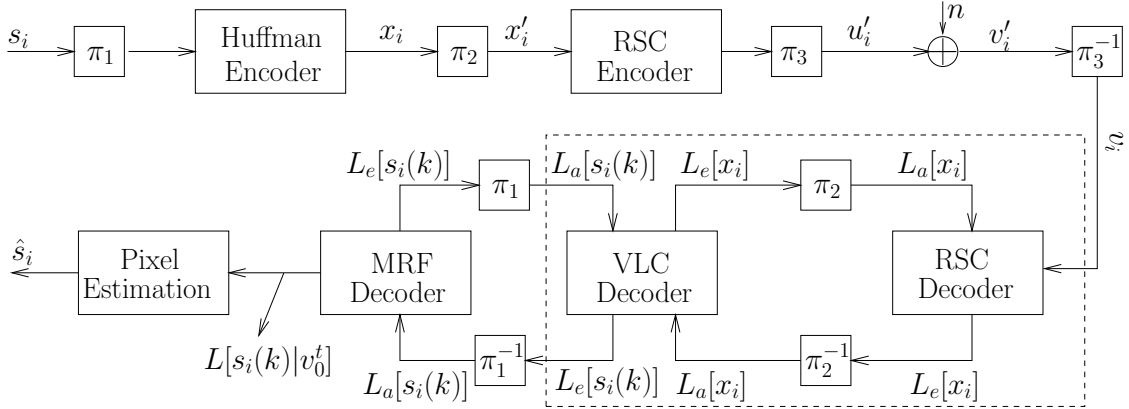


Figure 4.25: Architecture of the three-stage VLC-MRF-RSC system [185], where the soft VLC decoder and RSC decoder consist the inner decoding stage. π_1 is a pixel-level interleaver [186].

Finally, to show the beneficial effects of our proposed system, the benchmark system of Figure 4.25 was also considered. At the transmitter, the original video signals are encoded by a Huffman code. Then the compressed bitstream is encoded by the same RSC codec as that employed by the M3DISC-RSC system. At the receiver, the three decoders, namely the soft variable length codec (VLC) decoder [187], the soft MRF decoder [86] and the RSC decoder, perform three-stage decoding relying on joint source-channel decoding, where the MRF decoder was proposed for exploiting the correlation among adjacent pixels. In this system, the three Huffman codebooks (CB) designed for the YUV components have to be signaled to the receiver by the transmitter. Here we chose this system as a benchmark, since this system imposes similar complexity characteristics to those of our system. We refer to this system as VLC-MRF-RSC for simplicity.

A brief comparison of all the benchmarking schemes is shown in Table 4.6, where L-H.264-RSC represents the Lossless-H.264-RSC scheme. The Row Col (Trans./Rec.) compares the complexity imposed at the transmitter and receiver, respectively, while f represents the number of buffered frames invoked for decoding.

	Dim- ension	Bits Num to decode	SI	Code Rate	Delay (frames)	Col (Tr- ans./Rec.)
MMSE-HD	1	8	None	NCR	0	low/low
RSC	1	10000	None	$\frac{1}{2} \times \text{NCR}$	0	low/low
L-H.264-RSC	1	10000	None	$\frac{1}{2}$	8	high/low
FOMM-SBSD	1	64	$1 \times \text{MMSTT}$	NCR	0	low/high
IHVSM	2	512	$1 \times \text{MMSTT}$	NCR	0	low/high
M3DISC	3	$512 \sim f \times 512$	$2 \times \text{MMSTT}$	NCR	$0 \sim f - 1$	low/high
VLC-MRF- RSC	2	512	$3 \times \text{CB}$	$(\frac{1}{2} \times \text{NCR}, \frac{1}{2})$	0	low/high
M3DISC-RSC	3	$512 \sim f \times 512$	$2 \times \text{MMSTT}$	$\frac{1}{2} \times \text{NCR}$	$0 \sim f - 1$	low/high

Table 4.6: Comparison of three-dimensional iterative source-channel decoding using a recursive systematic convolutional (M3DISC-RSC) codec and the benchmarks: hard decoded minimum mean square error (MMSE-HD) [84], RSC, Lossless-H.264-RSC, first-order Markov modelling aided softbit source decoding (FOMM-SBSD) [87], [87], IHVSM [184], M3DISC, variable length coding aided Markov random field using RSC (VLC-MRF-RSC) [86].

4.3.2.4 Numerical Results

In this section, we present our simulation results for benchmarking the scheme introduced in Section 4.3.1 using Akiyo, Foreman and Coastguard sequences. We rely on two types of curves for characterizing the attainable video quality, namely the PSNR versus E_b/N_0 curves and the bit error ratio (BER) versus E_b/N_0 curves. Since the MMSE-based estimator outperforms the MAP-based estimator in terms of the PSNR video quality [184], we only present the simulation results, where the MMSE-based estimator is employed for pixel estimation. Note that to avoid having infinite PSNR values when a video frame is perfectly reconstructed, we artificially set the total averaged mean squared error (MSE) value between the reconstructed and the original frame to a minimum value of 1. This is justified, since the same technique is employed in the H.264 reference software JM. Hence the maximum unimpaired video PSNR that may be obtained at the receiver is about 48.1 dB.

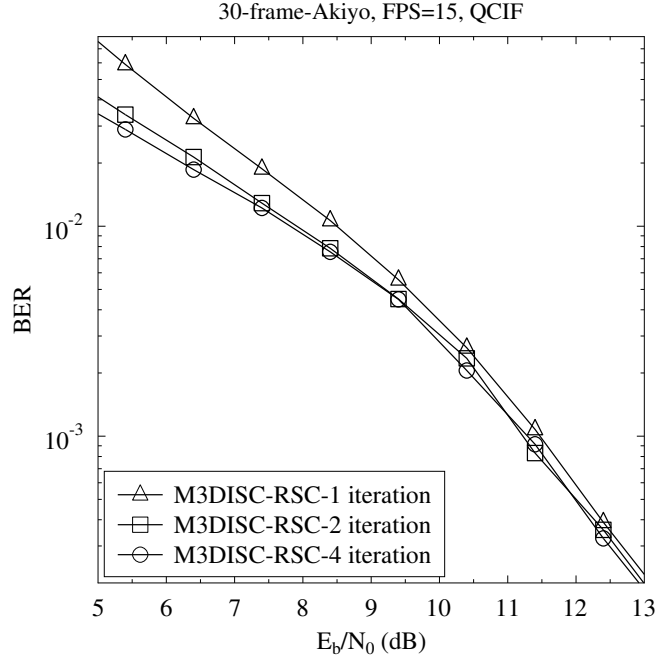


Figure 4.26: BER vs E_b/N_0 for a Rayleigh channel, when the MMSE-based pixel estimation is employed. The schematic of Figure 4.19 configured with the parameters in Tables 4.4 and 4.5 was used.

Firstly, we present the BER versus E_b/N_0 performance of the M3DISC-RSC scheme of Figure 4.19 in Figure 4.26 using the Akiyo sequence when tolerating a maximal delay of $(f - 1)/FPS = 933$ ms by setting $f = 15$, while the relevant Y-PSNR versus E_b/N_0 performance is displayed in Figure 4.27. Observe from the two figures that we can achieve a BER of about 8×10^{-3} and a Y-PSNR of about 40 dB at a E_b/N_0 of 8.4 dB using 4 iterations. Furthermore, we observe in Figure 4.27 that at a E_b/N_0 of 10.4 dB the M3DISC-RSC using 4 iterations performs slightly worse than after a single iteration. This may due to the fact that the parameters of Markov processes trained using the Akiyo video sequence does not exactly match the distribution of some of the blocks in specific frames, as exemplified by the boundaries of objects, where the pixel values may change drastically. Another reason for this phenomenon is that we employ a short interleaver of only 512 bits, which cannot entirely prohibit error propagation during the decoding process.

Let us now present our performance comparison of the M3DISC-RSC scheme with a delay of $(f - 1) = 14$ frames and contrast it to the benchmarks, where $I_{outer} = 1$ (outer) iteration is employed for all schemes. Since the BER metric is less relevant than the PSNR metric in reflecting the perceptual video quality, here we present the BER vs E_b/N_0 curves in Figure 4.28 only for the Akiyo sequence. More specifically, $I_{inner} = 1$ inner iteration is employed for the M3DISC, since it outperforms the ones with more iterations [184]. Observe in Figure 4.28 that at a BER of 5×10^{-3} , the

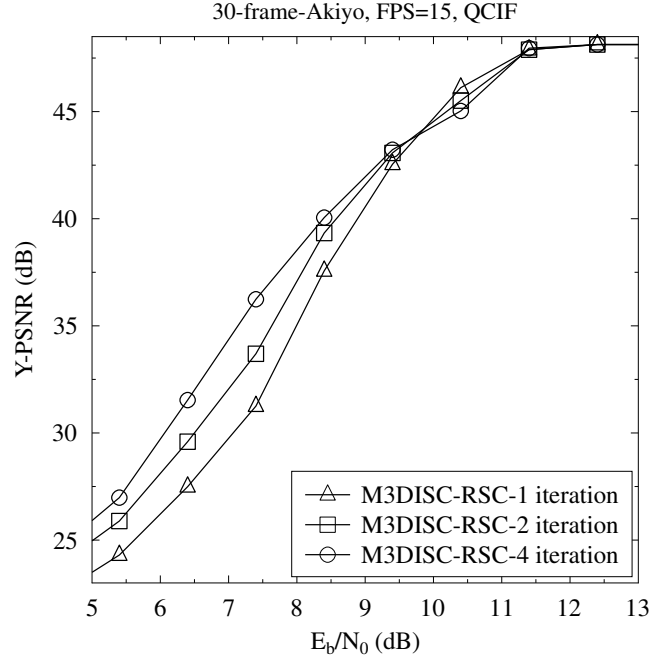


Figure 4.27: Y-PSNR vs E_b/N_0 for a Rayleigh channel when the MMSE is employed for pixel estimation. The schematic of Figure 4.19 configured with the parameters in Tables 4.4 and 4.5 was used.

M3DISC-RSC scheme of Figure 4.19 outperforms the IHVSM, FOMM-SBSD and VLC-MRF-RSC schemes by about 14.7 dB, 17.1 dB and 3.5 dB, respectively, while the M3DISC scheme achieves a power reduction of 7.5 dB compared to the IHVSM. Even though the Lossless-H.264-RSC achieves the best BER performance, its bits are extremely sensitive to bit errors.

Moreover, the PSNR vs E_b/N_0 curves are recorded in Figure 4.29 for the Akiyo, Foreman and Coastguard sequences. As seen in Figure 4.29 for the Akiyo sequence at a Y-PSNR of 46 dB⁹, the M3DISC-RSC scheme outperforms the IHVSM, FOMM-SBSD, Lossless-H.264-RSC and VLC-MRF-RSC arrangements by about 12.4 dB, 14.8 dB, 3 dB and 5.1 dB in terms of the required transmission power, respectively, while the M3DISC scheme attains a power reduction of 8.6 dB compared to the IHVSM. In other words, the M3DISC-RSC scheme outperforms the IHVSM, FOMM-SBSD, Lossless-H.264-RSC and VLC-MRF-RSC arrangements in terms of its reconstructed video quality by 12.9 dB, 15 dB, more than 20 dB and more than 20 dB of Y-PSNR at an E_b/N_0 level of 9.4 dB, respectively.

Viewing Figure 4.30 from a different perspective, we observe for the Foreman sequence, that at a Y-PSNR of 46 dB, the M3DISC-RSC scheme outperforms the IHVSM, FOMM-SBSD, Lossless-H.264-RSC and VLC-MRF-RSC arrangements by

⁹Here we are interested in this high video quality, since this treatise considers the quality-sensitive applications.

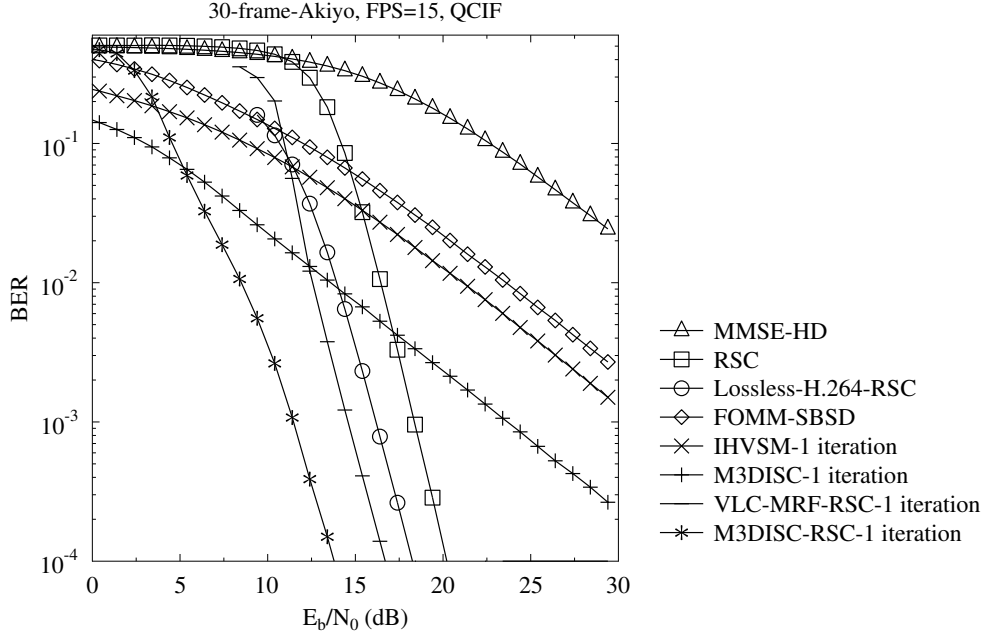


Figure 4.28: BER comparison of M3DISC-RSC and the benchmarks: MMSE-HD [84], RSC, Lossless-H.264-RSC, FOMM-SBSD [87], IHVSM [184], M3DISC, VLC-MRF-RSC [86]. The schematic of Figure 4.19 configured with the parameters in Tables 4.4 and 4.5 was used. Akiyo sequence

about 11.8 dB, 14.8 dB, 6.7 dB and 3.4 dB in terms of the required transmission power, respectively, while the M3DISC scheme achieves a power reduction of 3 dB compared to the IHVSM. In other words, it becomes explicit from Figure 4.30 that the M3DISC-RSC scheme attains a Y-PSNR improvement of about 13.4 dB and 17.2 dB at a E_b/N_0 of 7.5 dB compared to the IHVSM and the FOMM-SBSD, respectively. As seen in Figure 4.31 for the Coastguard sequence, when considering a Y-PSNR of 46 dB, the M3DISC-RSC scheme achieves a power reduction of about 13.2 dB, 7.8 dB and 3.3 dB compared to the IHVSM, the Lossless-H.264-RSC and the VLC-MRF-RSC, while the M3DISC scheme outperforms the IHVSM by about 3.1 dB. Alternatively, Figure 4.31 suggests that the M3DISC-RSC scheme outperforms the IHVSM and the FOMM-SBSD by about 14 dB and 15.2 dB in terms of the attainable Y-PSNR at an E_b/N_0 of 7.5 dB, respectively.

From the above discussions, we may conclude that our proposed M3DISC-RSC system substantially outperforms the IHVSM, FOMM-SBSD, Lossless-H.264-RSC and VLC-MRF-RSC schemes in terms of the Y-PSNR video quality achieved. Even though the Lossless-H.264-RSC has the best BER performance, its error-sensitive bits reduce the robustness of the streamed video signals. Furthermore, according to the Y-PSNR results of Figures 4.29, 4.30 and 4.31, we may attain an improved power reduction by employing the M3DISC-RSC scheme for the video sequences exhibiting

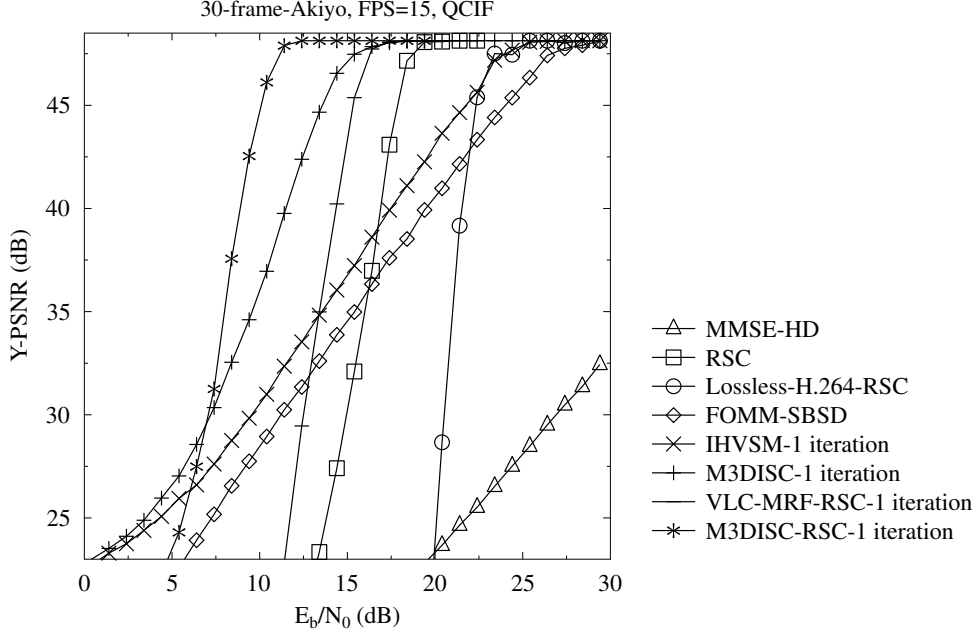


Figure 4.29: Reconstructed video quality of M3DISC-RSC and of the benchmarkers: MMSE-HD [84], RSC, Lossless-H.264-RSC, FOMM-SBSD [87], IHVSM [184], M3DISC, VLC-MRF-RSC [86]. The schematic of Figure 4.19 configured with the parameters in Tables 4.4 and 4.5 was used. Akiyo sequence

dynamic motions compared to the Lossless-H.264-RSC system, since the Lossless-H.264 codec substantially reduces the robustness of the system while only achieving a modest compression ratio.

A subjective comparison of the decoded Akiyo sequence at $E_b/N_0 = 9.4$ dB is displayed in Figure 4.32, where $I_{outer} = 1$ iteration is employed for all the benchmarkers. Observe from Figure 4.32 that the proposed M3DISC-RSC scheme is capable of recovering the error-infested video substantially better than the benchmarkers.

4.3.2.5 System Optimization

In this section, we characterize the M3DISC-RSC scheme associated with different delays using the Akiyo sequence. In the simulations of Section 4.3.2.4, we always buffered f frames for joint decoding, which induces a delay of $(f - 1)$ frames. However, in practical scenarios, we may buffer a reduced number of $d + 1$ ($\leq f$) frames and utilize the preceding $(f - d - 1)$ frames previously reconstructed at the receiver. In this case, we can perform decoding using f frames, which consist of $(d + 1)$ newly buffered softbit frames and $(f - d - 1)$ reconstructed hardbit frames. Correspondingly, we impose a delay of d frames. The PSNR vs E_b/N_0 curves using $f = 15$ and variable value of d is shown in Figure 4.33. Observe that only a moderate gain of about $E_b/N_0 = 1$ dB can be achieved upon increasing the delay from $d = 0$ to 14

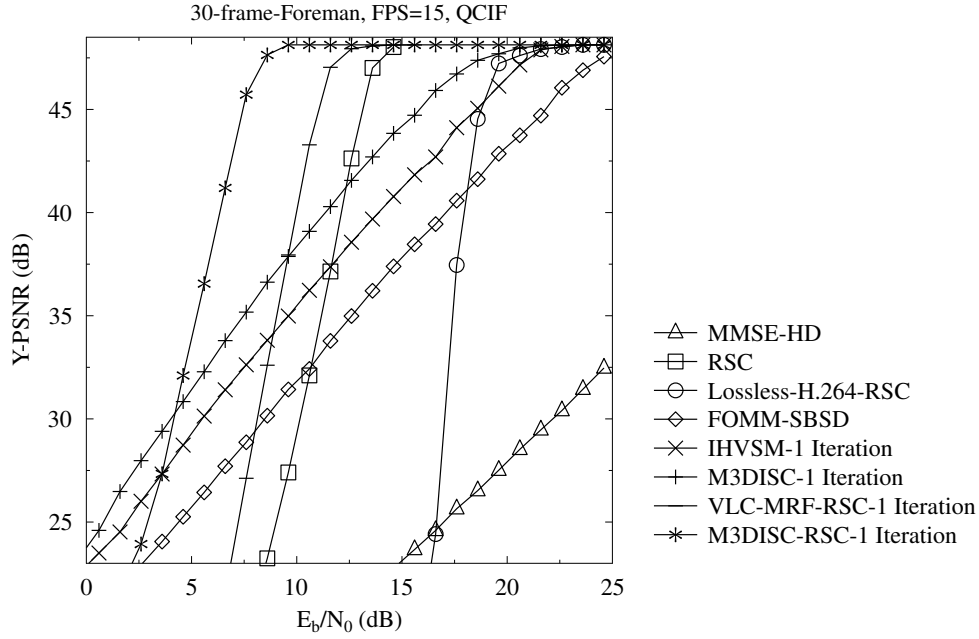


Figure 4.30: Reconstructed video quality of M3DISC-RSC and of the benchmarkers: MMSE-HD [84], RSC, Lossless-H.264-RSC, FOMM-SBSD [87], IHVSM [184], M3DISC, VLC-MRF-RSC [86]. The schematic of Figure 4.19 configured with the parameters in Tables 4.4 and 4.5 was used. Foreman sequence

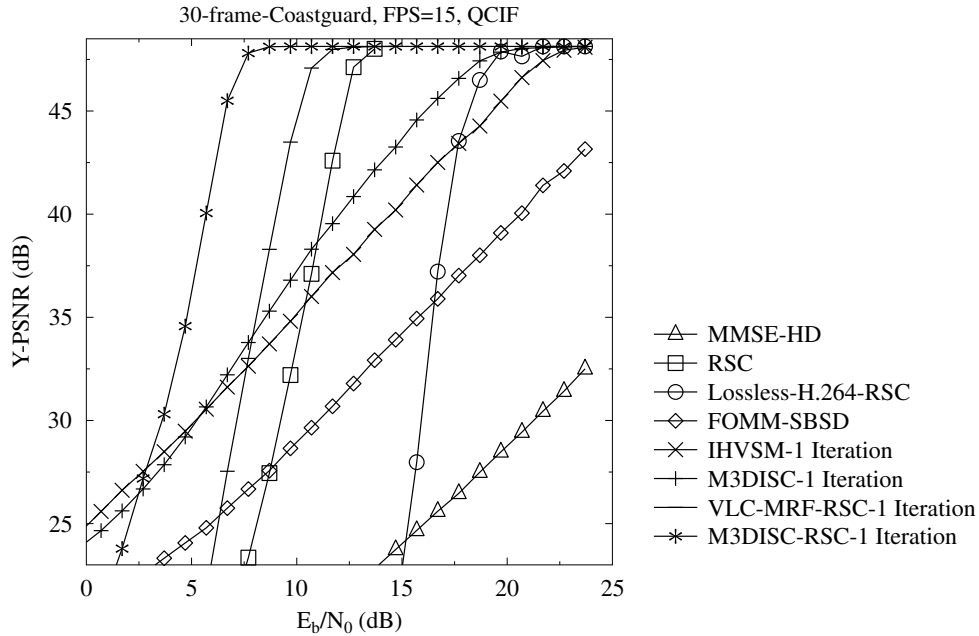


Figure 4.31: Reconstructed video quality of M3DISC-RSC and of the benchmarkers: MMSE-HD [84], RSC, Lossless-H.264-RSC, FOMM-SBSD [87], IHVSM [184], M3DISC, VLC-MRF-RSC [86]. The schematic of Figure 4.19 configured with the parameters in Tables 4.4 and 4.5 was used. Coastguard sequence

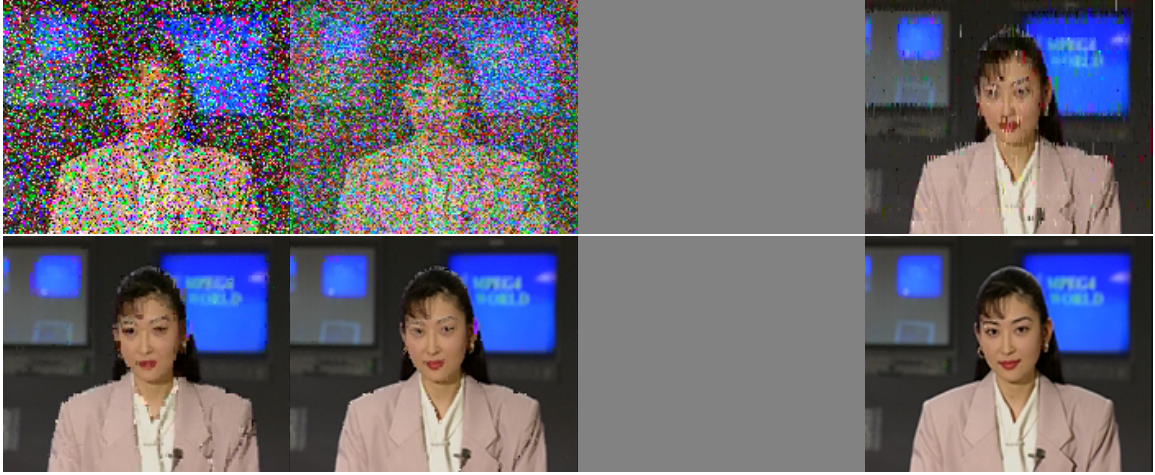


Figure 4.32: A frame comparison of the decoded Akiyo sequence at $E_b/N_0 = 9.4$ dB. The frames are reconstructed by MMSE-HD [84], RSC, Lossless-H.264-RSC, FOMM-SBSD [87], IHVSM [184], M3DISC, VLC-MRF-RSC [86], M3DISC-RSC, respectively. The schematic of Figure 4.19 configured with the parameters in Tables 4.4 and 4.5 was used.

frames, which suggests that we may decrease the delay of our system by appropriately tuning the decoder at an acceptable PSNR performance degradation.

Again, our system operates on a block-by-block basis. In Section 4.3.2.4, a constant block size of (8×8) -pixels was employed. Below, we present the performance of the M3DISC-RSC scheme configured for variable block sizes, ranging from (8×8) to (88×72) , using $f = 15$, as well as different number of iterations. The corresponding PSNR vs E_b/N_0 performance is displayed in Figure 4.34 using the Akiyo sequence, where the system's performance substantially improved upon increasing the size of the block. This may be attributed to the fact that both the source decoders and the RSC decoder may benefit from increasing the length of the interleavers.

4.4 Summary and Conclusions

This chapter summarized our iterative source decoding schemes conceived for video-quality enhancements, which may be combined with arbitrary FEC codecs. Furthermore, a reduced-complexity first-order Markov model based source decoder was derived. Iterative decoding was performed by exchanging extrinsic information among the FEC codec and two source decoders, namely the HSMD and the VSMD.

The simulation results of Figures 4.13, 4.14 and 4.15 demonstrated that the proposed IHVSM-RSC-MMSE system substantially improves the achievable system performance and may facilitate a transmit power reduction in excess of 2.8 dB in uncompressed video coding. More specifically, the attainable E_b/N_0 gains are listed

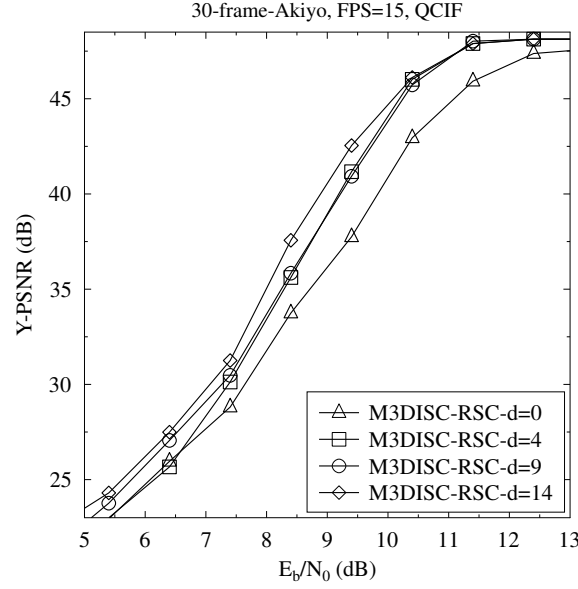


Figure 4.33: Performance of M3DISC-RSC with different delay tolerance, where d represents the delay expressed in terms of number of frames. The schematic of Figure 4.19 configured with the parameters in Tables 4.4 and 4.5 was used.

in Table 4.7, which were obtained by the schemes of Section 4.2.4 compared to the Lossless-H.264-RSC benchmark scheme of Section 4.24.

Furthermore, we detailed the first-order Markov process aided three-dimensional iterative source-channel decoding concept relying on an RSC codec for uncompressed video transmissions, where both the horizontal and vertical intra-frame correlations as well as the inter-frame correlations were exploited by relying on first-order Markov processes. The proposed technique is capable of exploiting both the intra-frame and inter-frame correlations for iterative source-channel decoding. Furthermore, a single RSC codec was combined with three independent source decoders for forming three decoder-pairs, for three-stage decoding, where the RSC was utilized for improving the source decoder's convergence behavior.

The simulation results of Figures 4.29, 4.30 and 4.31 showed that the proposed M3DISC-RSC system substantially improves the achievable system performance of uncompressed video systems. More specifically, the attainable E_b/N_0 gains are listed in Table 4.8, which are achieved by our schemes of Section 4.3.2 compared to the Lossless-H.264-RSC benchmark scheme. Furthermore, the M3DISC-RSC system of Figure 4.19 was refined by employing larger block sizes and more iterations, as displayed in Figure 4.34, while the corresponding E_b/N_0 gains are listed in Table 4.9.

The core of the algorithms introduced in this chapter is the Markov process based decoding assisted by the trained MMSTTs of Section 4.2.3.5. In some applications

	Akiyo (dB)	Coastguard (dB)	Football (dB)	
MAP-HD	<-10	na	na	Eq. (4.2)
MMSE-HD	<-10	<-7.6	<-7.6	Eq. (4.3)
Uncompressed-RSC	-3.7	4.3	5.5	Figure 4.7
FOMM-SBSD-MAP	<-10	na	na	[87]
FOMM-SBSD-MMSE	<-10	<-7.6	-7.1	[87]
FOMM-RSC-MMSE	-0.2	5.7	8.2	[87]
IHVSM-MAP	<-10	na	na	Figure 4.1
IHVSM-MMSE	-9.4	-5.7	-4.4	Figure 4.1
IHVSM-RSC-MAP	2.3	7.3	9.4	Figure 4.7
IHVSM-RSC-MMSE	2.8	<7.6	9.6	Figure 4.7

Table 4.7: E_b/N_0 gains obtained by the different schemes compared to the Lossless-H.264-RSC Figure 4.8 at a PSNR of 47.5 dB. The schematic of Figure 4.7 configured with the parameters in Table 4.2 was used for Akiyo-QCIF, Coastguard-QCIF and Football-QCIF sequence

	Akiyo (dB)	Foreman (dB)	Coastguard (dB)	
MMSE-HD	<-15	<-8.7	<-8	Eq. (4.3)
RSC	-4.9	1.7	2.5	Section 4.3.2.3
FOMM-SBSD	-12.6	-8.7	<-8	Figure 4.1
IHVSM	-10.2	-5.3	-6	Figure 4.1
M3DISC	-1.3	-3.2	-3.1	Figure 4.23
VLC-MRF-RSC	-2.3	3.8	4.5	Figure 4.25
M3DISC-RSC	2.9	7.4	8.2	Figure 4.19

Table 4.8: E_b/N_0 gains obtained by different benchmarks compared to the Lossless-H.264-RSC of Figure 4.8 at a PSNR of 47.5 dB. The schematic of Figure 4.19 configured with the parameters in Tables 4.4 and 4.5 was used for Akiyo, Foreman and Coastguard sequences.

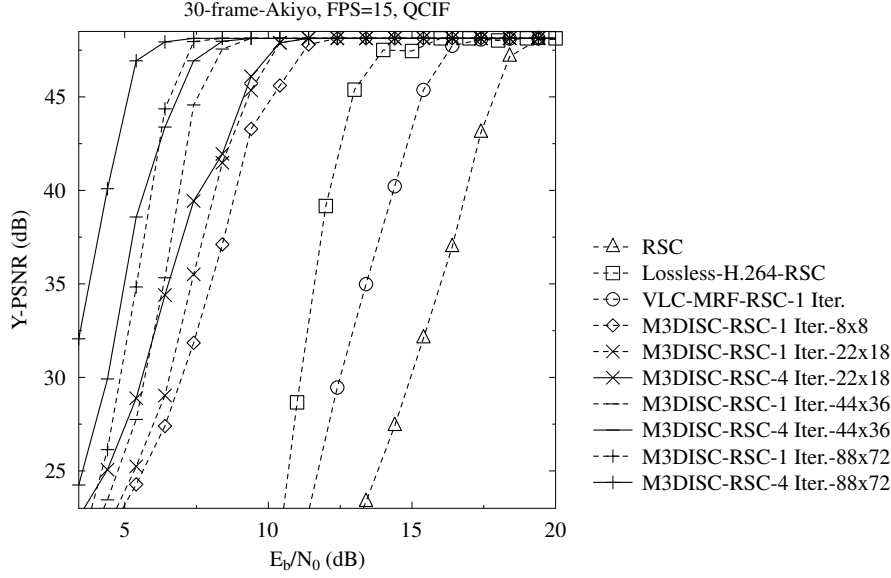


Figure 4.34: Performance of M3DISC-RSC with different block size and constant $f = 15$, where 8×8 , 22×18 etc. represent size of block. The performance of RSC, Lossless-H.264-RSC and VLC-MRF-RSC are included for benchmarking. The schematic of Figure 4.19 configured with the parameters in Tables 4.4 and 4.5 was used.

we might need 8-bit transition probabilities and, additionally the video may contain dynamic high-motion scenes. A realistic option is to scan the difference between the adjacent pixels in order to approximate the related difference probability table (DPT), which generally obeys a Laplacian distribution [188]. Hence, what we have to transmit is not a (256×256) -element matrix containing the MMSTT, but a one-dimensional table of $(255 \times 2 + 1)$ probabilities indicating histogram of the differences between adjacent pixels, which may be further simplified by sending just two parameters for characterizing the corresponding Laplacian distribution model. At the receiver, the DPT will be used for generating the 2D MMSTT. Specifically, provided that difference $-255 \leq d \leq 255$, the difference probability obeys $p(u, u + d) = p(u + d, u) = p(d), 0 \leq u \leq 255$. Performance comparison of these two cases are provided in Figure 4.35 using the Akiyo-QCIF sequence. Furthermore, another option for obtaining the required side information is that of using long-term offline training. Alternatively, in most applications, we can also estimate the above-mentioned side information at the receiver from the past decoded frames. Naturally, the accuracy of the transition probabilities substantially affects the achievable performance. These comments motivated us to embark on further investigations on estimating the Markov-state transition probability table.

	Akiyo (dB)	
RSC	-4.7	Section 4.3.2.3
VLC-MRF-RSC	-2.3	Figure 4.25
M3DISC-RSC-1 Iter.-8x8	2.8	Figure 4.19
M3DISC-RSC-1 Iter.-22x18	3.8	Figure 4.19
M3DISC-RSC-4 Iter.-22x18	3.8	Figure 4.19
M3DISC-RSC-1 Iter.-44x36	5.6	Figure 4.19
M3DISC-RSC-4 Iter.-44x36	6	Figure 4.19
M3DISC-RSC-1 Iter.-88x72	6.7	Figure 4.19
M3DISC-RSC-4 Iter.-88x72	7.9	Figure 4.19

Table 4.9: E_b/N_0 gains compared to the Lossless-H.264-RSC Figure 4.8 at a PSNR of 47.5 dB obtained by different benchmarks, which may be configured with different size of block and iterations. The schematic of Figure 4.19 configured with the parameters in Tables 4.4 and 4.5 was used for Akiyo sequence.

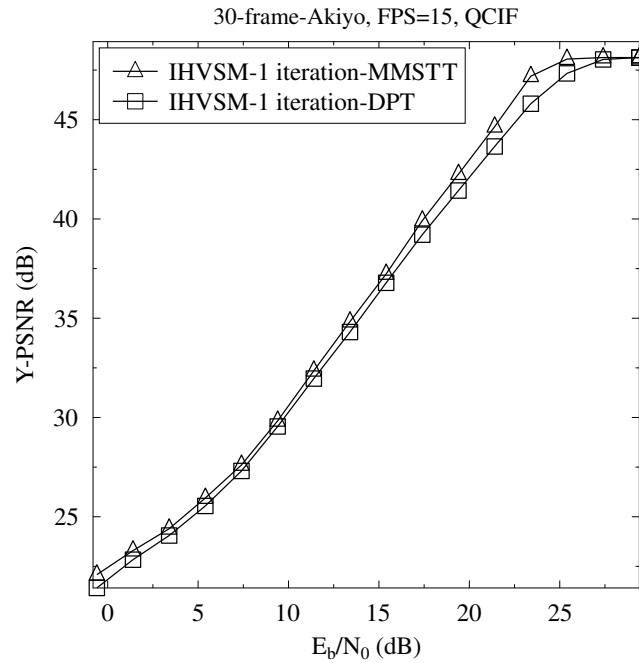


Figure 4.35: Performance comparison using MMSTT and DPT for Akiyo-QCIF and Football-QCIF sequences. MMSE estimator is employed. The schematic of Figure 4.7 configured with the parameters in Table 4.2 was used.

Iterative Source-Channel Decoding Aided Distributed Video Coding

5.1 Introduction

In the standardized MPEG or the ITU-T H.26x video codecs [1], the encoder exploits the correlation of the source signals. This principle has indeed been routinely exploited until the appearance of multimedia applications, where the transmitter is highly constrained in terms of its complexity, memory and power, dissipation, as exemplified by wireless sensor networks [127]. However, efficient compression can also be achieved by exploiting the source-correlation at the decoder side. This insightful result is the consequence of the information theoretic discoveries of Slepian and Wolf, which led to the concept of for distributed lossless coding [91]. The concept was later extended to lossy source coding by Wyner and Ziv [99]. The schemes that are based on these theorems are generally known as distributed coding algorithms, Slepian-Wolf (SW) coding and Wyner-Ziv (WZ) coding. The SW and WZ theorems will be further illustrated in Section 5.2.

Since the resource-limited encoder fails to remove all the redundancy inherent in the video signal in the distributed video coding (DVC) [103] applications. It is beneficial to further exploit the correlation at the decoder side for the sake of reducing the bitrate required for adequately communicating the video signal. Specifically, the traditional motion estimation [189] is deemed to be one of the most computational complex operation in video coding, which should be abandoned at the resource-limited transmitters. Hence, the correlation between adjacent frames is not removed at the encoder before transmission. Although a number of techniques [190, 105, 106,

103, 100, 111, 114, 115] were proposed for side information (SI) estimation of the so-called key frames [103], the correlation should be further exploited by modelling the intra-frame and inter-frame correlation for the sake of reducing the required bitrate. In Chapter 4, we proposed the two-dimensional intra-frame correlation modelling process. Since in distributed multiview video coding [111, 112] designed for wireless video sensor networks, the inter-view correlation cannot be removed by the encoders, because the transmitters of different views do not have access to other adjacent views, the inter-view correlation should be exploited at the receiver side for the sake of reducing the bitrate required.

Motivated by the principle of exploiting the source redundancy at the receiver, in Section 5.3 we study the application of ISCD conceived for DVC [191], where the video signal is modelled by a first-order Markov process decoded by the reduced-complexity formulas detailed in Chapter 4. The bit-based iterative horizontal-vertical decoding algorithm detailed in Chapter 4 will be utilized, where a horizontal and a vertical source decoder are employed for exchanging their information using the iterative decoding philosophy. This scheme may be combined with the classic FEC codecs employed in the state-of-the-art DVC systems. In Section 5.3.2, we characterize the attainable system performance against that of the existing pixel-domain Wyner-Ziv (PDWZ) video coding systems, detailed in Section 5.2.

Furthermore, we exploit the inter-view correlation with the aid of inter-view motion search in distributed multi-view video coding (DMVC) in Section 5.4. Firstly, we rely on the system architecture of WZ coding invoked for multi-view video. Then we construct a novel mesh-structured pixel-correlation model from the inter-view motion vectors (MVs) [192] and derive its decoding rules for joint source-channel decoding in Section 5.4.2. Finally, we benchmark the attainable system performance against the existing pixel-domain WZ (PDWZ) coding based DMVC scheme in Section 5.4.3, where the classic turbo codec is employed.

5.2 A Pixel-Domain Wyner-Ziv Architecture

In Figure 5.1, we introduce the general structure of a PDWZ system architecture, which will also be adopted in our system throughout this chapter. The structure of Figure 5.1 is developed from the WZ video coding system proposed in [105, 103]. Generally, this structure relies on a uniform quantizer, a turbo code based Slepian-Wolf codec, a frame estimation block, a video reconstruction block and a request-and-decode process. Below, we highlight the main processing blocks of this structure, while more details of this PDWZ system may be found in [188]. An example of the

PDWZ processing of Figure 5.1 is shown Figure 5.2, where the *Outdoor* sequence is used.

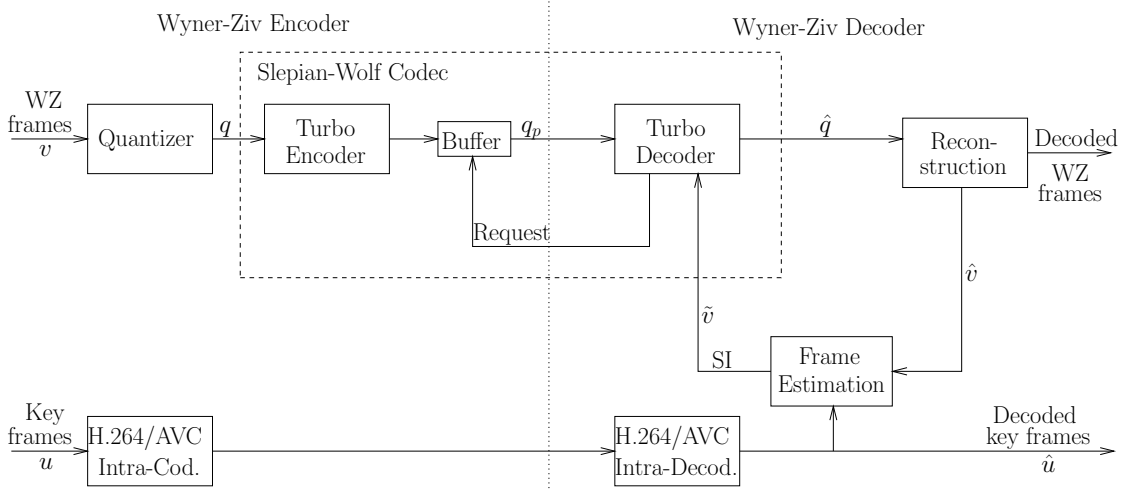


Figure 5.1: Architecture of a pixel-domain Wyner-Ziv encoder/decoder scheme, which is developed from the PDWZ system proposed in [105, 103].

5.2.1 Encoder Operations

1. Frame classification

At the input of the WZ codec of Figure 5.1, a video frame sequence is split into two types of frames, namely the so-called key frames and the WZ frames. The key frames will be encoded by a typical intra-frame encoder, such as the H.264/AVC intra-frame encoder of Figure 5.1. The WZ frames will be encoded by the WZ codec. Moreover, each GOP consists of a single key frame and a number of WZ frames. For example, a key frame is inserted after every WZ frame in the scenario of $\text{GOP}=2$. As exemplified in Figure 5.2, the frames with indices of 0 and 2 are classified as key frames, while the frame with an index of 1 is classified as a WZ frame.

2. Uniform Quantization

Each pixel of the WZ frames is uniformly quantized to one of the 2^m levels, hence generating frame q having m bit pixels, where m is the number of most significant bit (MSB) planes. This quantized frame q is exemplified in Figure 5.2 for $m = 3$. Then, the resultant bitplanes are grouped according to the significance of the bits, as exemplified in Figure 5.3. All the m bitplanes will be encoded independently by a turbo codec.

3. Turbo encoding

All the m bitplanes, as exemplified in Figure 5.3 will be encoded independently

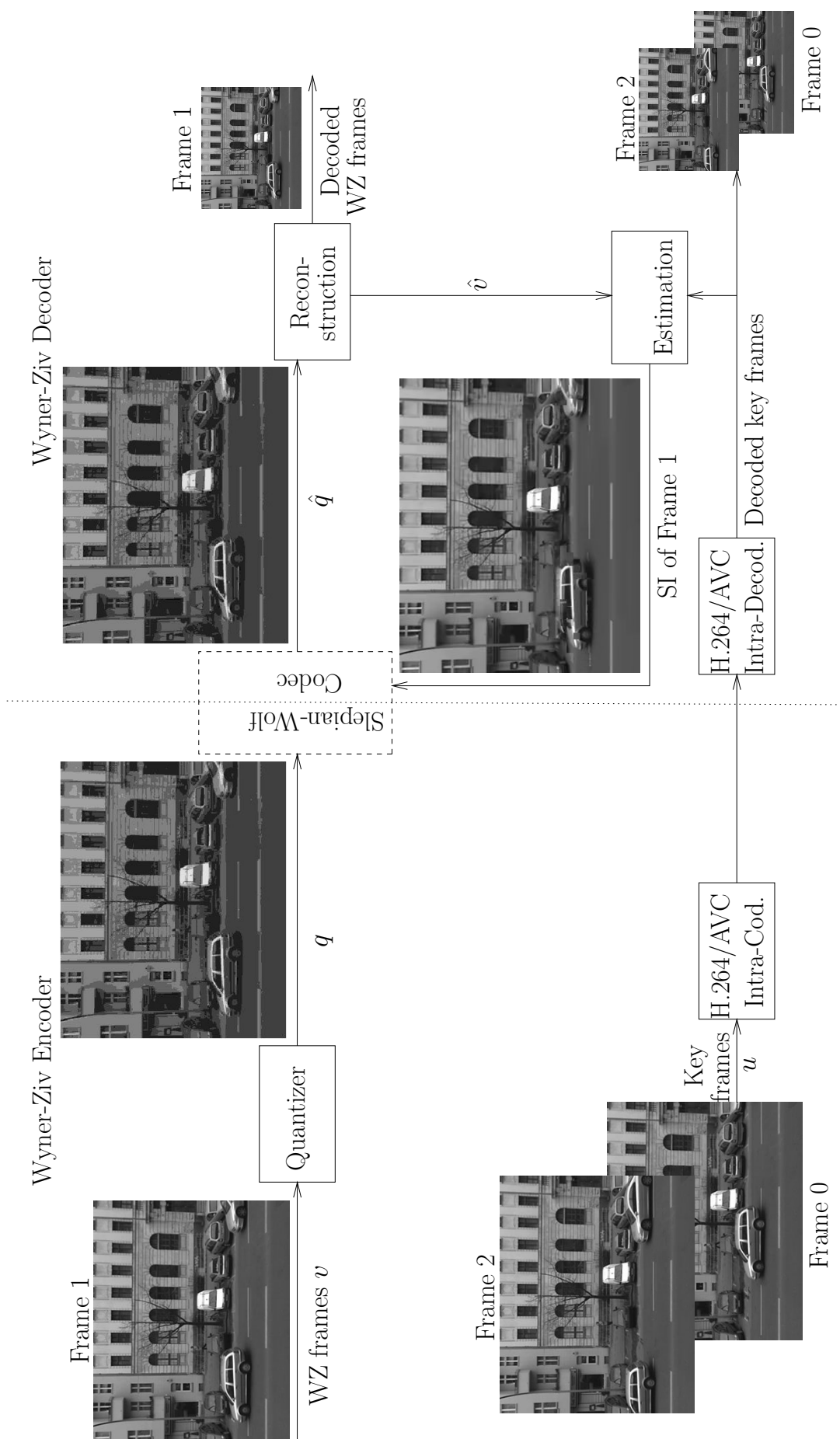
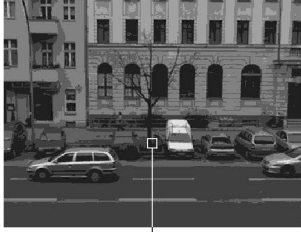


Figure 5.2: Example of the PDWZ processing of Figure 5.1, where the Ourdoor sequence was used with GOP=2.



$$\begin{bmatrix} 4 & 4 & 4 & 4 \\ 3 & 3 & 3 & 4 \\ 2 & 2 & 2 & 3 \\ 2 & 1 & 1 & 3 \end{bmatrix} = \begin{bmatrix} 100 & 100 & 100 & 100 \\ 011 & 011 & 011 & 100 \\ 010 & 010 & 010 & 011 \\ 010 & 001 & 001 & 011 \end{bmatrix} \\
 \downarrow m = 3 \\
 \overbrace{\begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix}}$$

Figure 5.3: A (4×4) macroblock from q of Figure 5.2 is split into $m = 3$ MSB planes.

by the turbo codec. Afterwards, the parity bits generated will be buffered at the transmitter, which may be transmitted to the receiver upon request from the receiver. Noting that the turbo codec here is used for embedding the information of the m bitplanes into the parity bits, which can then be used by the turbo decoder for reconstructing the m bitplanes at the receiver side. Since no channel effects are considered in Figure 5.1, turbo coding is utilized as a tool for compressing the m bitplanes instead of combating realistic channel-induced errors. Moreover, arbitrary FEC codecs may be employed here. In general, near-capacity codecs, such as turbo and LDPC codes, are favored in the literature [103, 104].

5.2.2 Decoding Operations

1. Frame Estimation

The “Frame Estimation” block of Figure 5.1 estimates the side information (SI) of each WZ frame utilizing both the nearest previous and past decoded frames. For example, in the scenario of $\text{GOP}=2$, if the index of the current WZ frame is i , the frames with indices of $(i - 1)$ and $(i + 1)$ will be used for the SI estimation. An SI frame is exemplified in Figure 5.2, which contains frame 1’s soft information indicating the reliability of each pixel. Note that the accuracy of this SI estimation process significantly affects the performance of the WZ coding scheme, since less of the parity bits of the WZ frames generated by the

turbo FEC code are required for correcting the resultant more accurate SI of the WZ frame.

2. Request-and-decode

Given the estimated SI of the current WZ frame, the turbo decoder of Figure 5.1 will attempt to recover all the m bitplanes of the quantized WZ frame. If the BER of decoded bitplanes is deemed to be higher than a preset threshold, the receiver will send a request to the transmitter to send more parity bits for this specific WZ frame through the feedback channel of Figure 5.1. Then the receiver will attempt to decode the bitplanes again. The above process will be repeated, until the BER of the decoded bitplanes becomes lower than the threshold. Once all the m bitplanes of the quantized frame has been successfully turbo decoded, the “request-and-decode” process of Figure 5.1 terminates and the receiver starts decoding the next WZ frame.

3. Video Reconstruction

Given all the m bitplanes of the current WZ frame, the original WZ frame may be reconstructed with the aid of inverse quantization. After all the key frames and the WZ frames have been reconstructed, the entire GOP may be readily recovered by appropriately reassembling the key frames and WZ frames.

5.3 IHVSM aided Wyner-Ziv Monoscopic Video Coding

In the Wyner-Ziv video coding schemes of Girod *et al.* [103], as well as of Brites and Pereira [100], the side information (SI) of the WZ frame is estimated from the so-called key frames, which are intra-coded by standard codecs, such as H.264 for example. Furthermore, a part of the parity bits of WZ frames is transmitted to the receiver upon request for the sake of assisting the receiver to recover the WZ frames. However, the pixel-correlation within WZ frames is not exploited. In this section, we will combine our IHVSM with the turbo-coding aided WZ codec [100], in order to iteratively conceal the softbit errors returned by the turbo decoder, thereby implicitly reducing the requested bitrate. Note that both the key frames and the WZ frames are transmitted to the receiver through a perfect channel. Hence the errors of the WZ frames are purely introduced by the SI estimation process at the receiver, where our proposed IHVSM model will exploit the intra-frame correlation of the WZ frames for correcting the errors, thereby reducing the amount of parity bits required for the WZ frame.

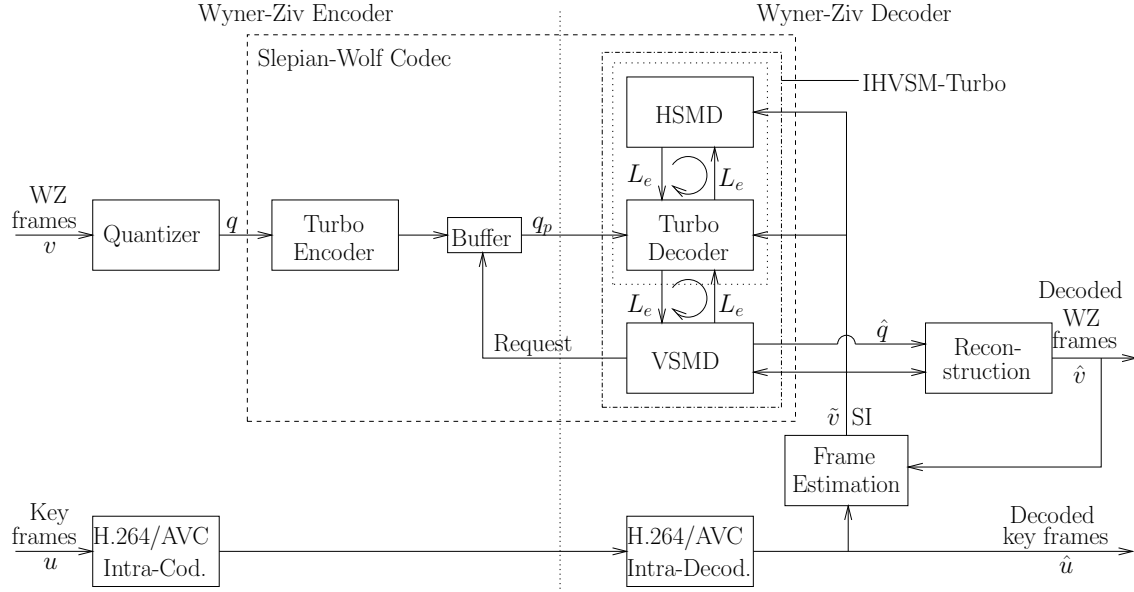


Figure 5.4: System architecture of our proposed IHVSM-PDWZ system, which is based on the PDWZ system of Figure 5.1. The IHVSM-Turbo decoder is employed in contrast to the turbo decoder employed in the PDWZ system of Figure 5.1.

	Hall	Foreman	Coastguard
Representation	YUV 4:2:0	YUV 4:2:0	YUV 4:2:0
Format	CIF	CIF	CIF
Bits Per Pixel	8	8	8
FPS	30	30	30
GOP	2	2	2
Number of Frames	300	300	300
Channel	Perfect	Perfect	Perfect

Table 5.1: Table of parameters employed for the *Hall*, *Foreman* and *Coastguard* sequences, as represented in Figure 5.5, for the schematic of Figure 5.4.

5.3.1 System Configuration

Generally, two types of techniques exist in the literature for encoding the WZ frames, namely pixel-domain WZ (PDWZ) [103, 100] and transform-domain WZ (TDWZ) [100] video coding. The PDWZ codec imposes a lower complexity on the encoder than the TDWZ scheme, hence - as expected - it exhibits a worse rate-distortion (RD) performance. In this section, we apply the proposed IHVSM algorithm in the context of a PDWZ scheme for improving the attainable RD performance. The Wyner-Ziv coding scheme advocated in [100, 188] is based on the structure proposed in [190, 106],



Figure 5.5: Frames for the *Hall*, *Foreman* and *Coastguard* sequences represented in CIF.

with a number of further refinements proposed. We opted for the PDWZ system of [100] as our benchmark, while some further details are illustrated in [188]. Note that the proposed IHVSM may also be employed in other WZ video coding schemes provided that the video decoders at the receiver are capable of estimating the softbit information of the video pixels.

Let us now proceed by detailing the turbo-coded (IHVSM-Turbo) PDWZ system of Figure 5.4. At the transmitter, the key frame u is intra-coded by the H.264/AVC video codec, whilst the WZ frame v is uniformly quantized for generating the 2D signal q . Then the bit-pattern q is encoded by the classic turbo codec [121], where only the parity bits q_p will be buffered and transmitted to the receiver. At the receiver, the key frame is reconstructed using the H.264/AVC intra-decoder. The decoded key frame \hat{u} will then be utilized by the frame estimator¹ of Figure 5.4 for estimating the side information (SI) frame \tilde{v} . Then, our IHVSM-Turbo decoder of Figure 5.4 will perform three-stage iterative decoding [41] using the estimated SI frame \tilde{v} and the received parity bits q_p . For the IHVSM-Turbo scheme, the SI frame \tilde{v} may be used as the systematic information of the corresponding WZ frame v , which will be exploited by all the three decoding components, namely the HSMD, the turbo codec, and the VSMD of Figure 5.4. For the HSMD and the VSMD components, the technique introduced in Section 4.2.1 is utilized for iterative decoding. Furthermore, the HSMD and the turbo codec constitute the inner decoding component, while the VSMD is the outer one. Similarly to most proposals found in the literature [103,100], the "request-and-decode"²process of [103] is repeated, until an acceptable probability of symbol error is reached. Finally, the quantized WZ frame is recovered as \hat{q} , which

¹More details of the frame estimation process can be found in [100]. Here we focus on our proposed IHVSM-Turbo scheme.

²When failing in decoding a WZ frame, the receiver sends a request to the transmitter, which will transmit additional parity bits to the receiver. Then the receiver re-decodes the WZ frame upon the accumulated parity bits.

will be utilized together with the estimated SI frame \tilde{v} for reconstructing the WZ frame \hat{v} .

In the following, we benchmark the proposed IHVSM-PDWZ scheme's performance against that of the PDWZ system of Brites and Pereira [100], where the traditional turbo codec was employed. Moreover, as seen in most of the state-of-the-art contributions [103], the corresponding results recorded both for the JPEG and for the H.264/AVC schemes are also provided as baseline benchmark curves in Section 5.3.2. More specifically, all the frames were intra-coded frame (I-frame) coded by the JPEG and H.264 codecs, as well as by the H.264/AVC NoMotion mode, where all the frames may be coded as I-frames and predicted frames (P-frames) without inter-frame motion estimation. Three sequences are tested, namely the Hall monitor, the Foreman and the Coastguard sequences, whose parameters are shown in Table 5.1. In all the experiments, the group of picture (GOP) parameter was set to 2; the bitrate/peak to-signal-noise ratio (PSNR) of both the WZ and the key frames was also taken into account. The RD results recorded for both the PDWZ and IHVSM-PDWZ schemes in Figure 5.4 were parameterized by the number of WZ coded bitplanes, which was set to $m = 1, 2, 3$ or 4 most significant bit (MSB) planes, as usual in the pixel-domain WZ video coding literature. This was arranged by invoking the uniform quantizer shown in Figure 5.4. In [188], each bitplane of the MSB planes was transmitted separately and each bitplane was then refined based on the previously decoded bitplanes [193]. However, in our system, all bitplanes of the MSB planes were transmitted together, which allowed us to reduce the number of "request-and-decode" processes defined in [103]. The remaining parameters of our system were identical to those in [100, 188]. Note that for the IHVSM-PDWZ schemes, the size of the Markov Model's State Transition Table (MMSTT) was of $(2^m \times 2^m)$ elements. Specifically, for the Hall sequence associated with $m = 4$, the size of the MMSTT side information was (16×16) floating-point values for 300 CIF frames³

5.3.2 Numerical Results

³In our simulations, the transmitter scans all adjacent pixel pairs and records their difference, which may be modelled by a Laplacian distribution. The required MMSTT is then generated at the receiver using the received Laplace parameters. Furthermore, since the key frames are available at the decoder, an alternative technique is to estimate the MMSTT from the adjacent key frames.

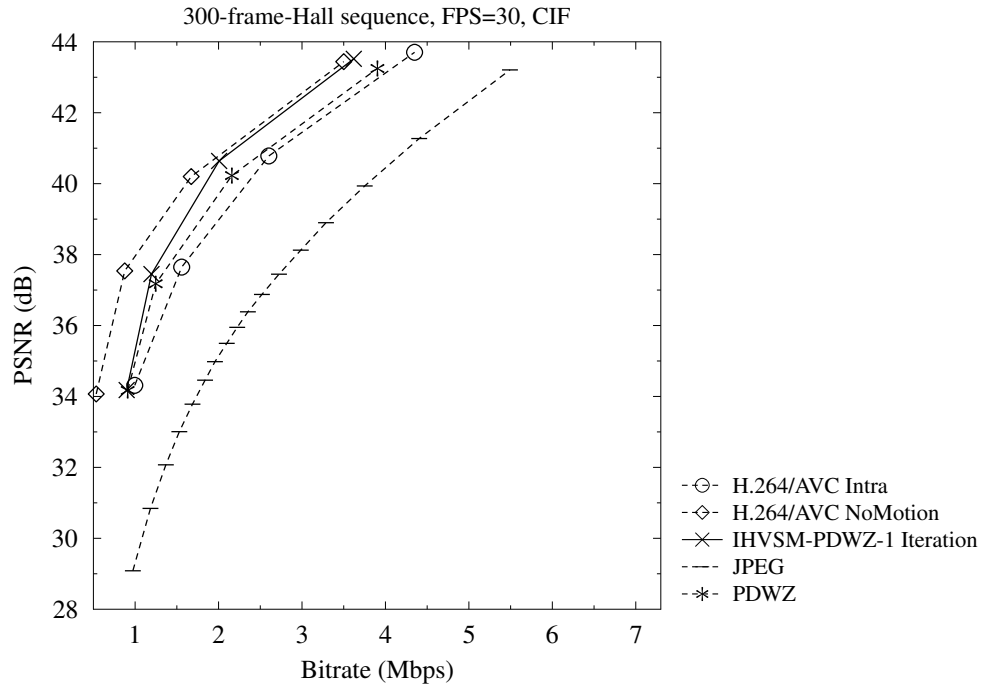


Figure 5.6: Rate-distortion performance comparison of the IHVSM-PDWZ codec and the benchmarks: JPEG, H.264/AVC Intra, H.264/AVC NoMotion, PDWZ [100]. The schematic of Figure 5.7 configured with the parameters in Table 5.1 was used. *Hall*, CIF

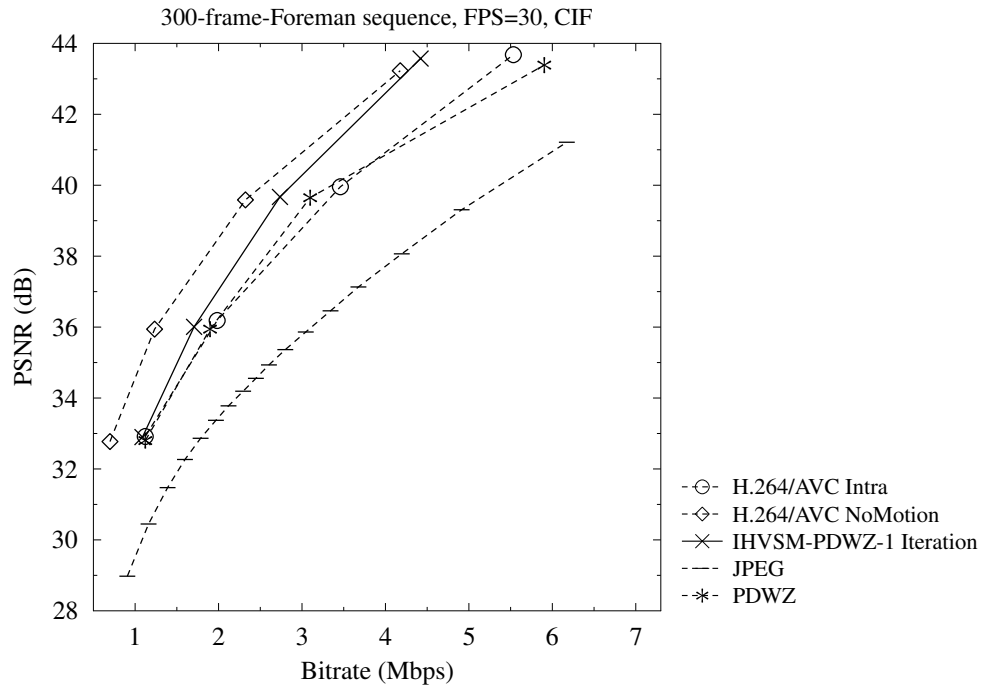


Figure 5.7: Rate-distortion performance comparison of the IHVSM-PDWZ codec and the benchmarks: JPEG, H.264/AVC Intra, H.264/AVC NoMotion, PDWZ [100]. The schematic of Figure 5.7 configured with the parameters in Table 5.1 was used. *Foreman*, CIF

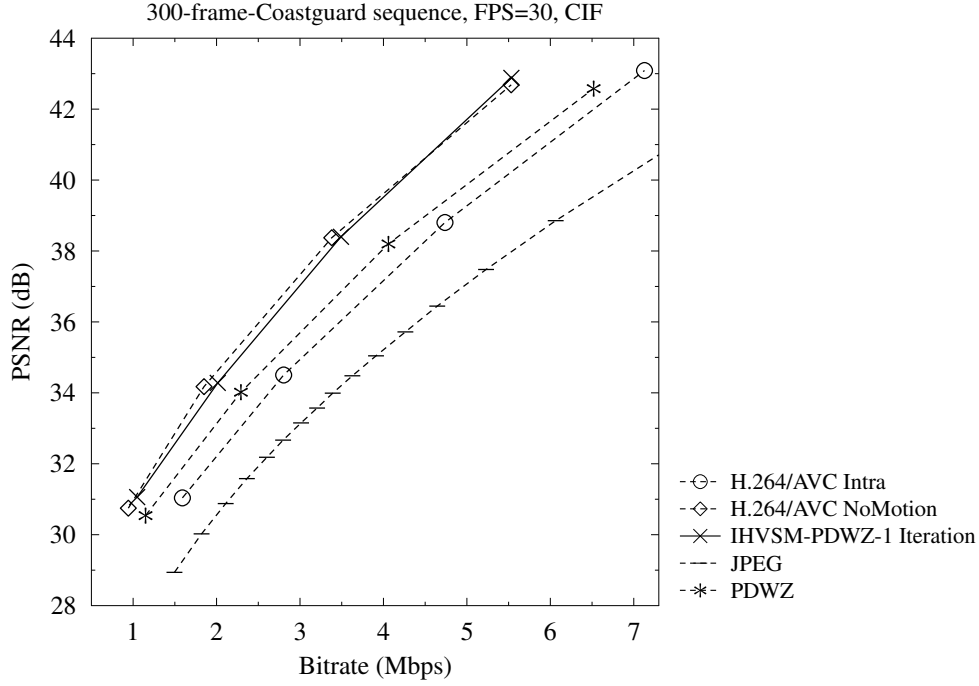


Figure 5.8: Rate-distortion performance comparison of the IHVSM-PDWZ codec and the benchmarks: JPEG, H.264/AVC Intra, H.264/AVC NoMotion, PDWZ [100]. The schematic of Figure 5.7 configured with the parameters in Table 5.1 was used. *Coastguard*, CIF

The simulation results recorded for the three sequences are displayed in Figures 5.6, 5.7 and 5.8. Observe from Figure 5.6 that for the Hall-CIF sequence the IHVSM-PDWZ using a single iteration reduces the bitrate by about 0%, 11%, 14.7% and 15% in comparison to the PDWZ scheme for $m = 1, \dots, 4$ MSB planes, respectively. This is due to the fact that our IHVSM scheme is capable of estimating and correcting the extrinsic information gleaned from the turbo codec during each iteration. Similarly, observe from Figure 5.7 that for the Foreman-CIF sequence the IHVSM-PDWZ using $I = 1$ iteration reduces the bitrate by 3%, 10.1%, 11.6%, 25.1% for $m = 1, \dots, 4$ MSB planes, respectively. For the Coastguard-CIF sequence, the IHVSM-PDWZ using $I = 1$ iteration reduced the bitrate by 8.7%, 12.1%, 14%, 15.1% for $m = 1, \dots, 4$ MSB planes, respectively.

The results of Figures 5.6, 5.7 and 5.8 are summarized in Table 5.2 for the three sequences of Table 5.1, namely for the Hall, Foreman and Coastguard sequences. Generally, more substantial improvements may be obtained for the more dynamic higher-motion sequences, than for the lower-motion sequences. For all the test sequences, the bitrate reduction increased with the number of MSB planes considered. The reason for this is that the receiver is more unlikely to be able to estimate the SI of WZ frame accurately from the received key frames, while our IHVSM scheme is capable of effectively correcting the errors in the turbo decoding process. Moreover,

Sequence	ΔR (%)			
	$m = 1$	$m = 2$	$m = 3$	$m = 4$
Hall	0.0	11.0	14.7	15.0
Foreman	3.0	10.1	11.6	25.1
Coastguard	8.7	12.1	14.0	15.1

Table 5.2: Bitrate comparison of the IHVSM-PDWZ versus the PDWZ schemes for the considered sequences of Table 5.1. “R” indicates “Rate”. The parameter m indicates the number of MSB planes. The schematic of Figure 5.7 configured with the parameters in Table 5.1 was used.

as seen in Figures 5.6, 5.7 and 5.8, the IHVSM-PDWZ system of Figure 5.4 is comparable to the H.264/AVC-NoMotion system at sufficiently high bitrates corresponding to high PSNRs. For example, in the PSNR range in excess of 41 dB, the IHVSM-PDWZ slightly outperforms the H.264/AVC-NoMotion system, when employing the Coastguard-CIF sequence.

5.4 Motion-Aware Mesh-Structured Correlation Modelling

In this section, we firstly introduce the system model of the proposed Wyner-Ziv coding scheme conceived for multi-view video. Then, the intricacies of our motion aware mesh-structured correlation modeling will be detailed, which models the inter-view pixel correlation within the Wyner-Ziv frames. Finally, we will provide simulation results for characterizing our system.

5.4.1 Wyner-Ziv Coding for Multi-view Video: System Model

Again, WZ compression techniques designed for monoscopic video have attracted substantial research attention [190, 105, 107, 106, 103, 104, 101, 113, 110, 100, 114, 188, 115]. A number of contributions have also been proposed for DMVC [108, 129, 111, 112]. In this section, we will detail the WZ compression philosophy shown in Figure 5.9, which is invoked for distributed MVC employing our proposed MSSM-Turbo decoder. In the system of Figure 5.9, there are N cameras capturing N views, respectively. We consider an array of N cameras, which employ N identical low-complexity video encoders for encoding the N camera-views independently at the transmitter and a potentially high-complexity video decoder for jointly decoding the N camera-views at the receiver. Each group of pictures (GOP) of each of the N camera views consists of a single I-frame followed by a fixed number of WZ frames. The encoding structure of the WZ MVC encoder is displayed in Figure 5.10.

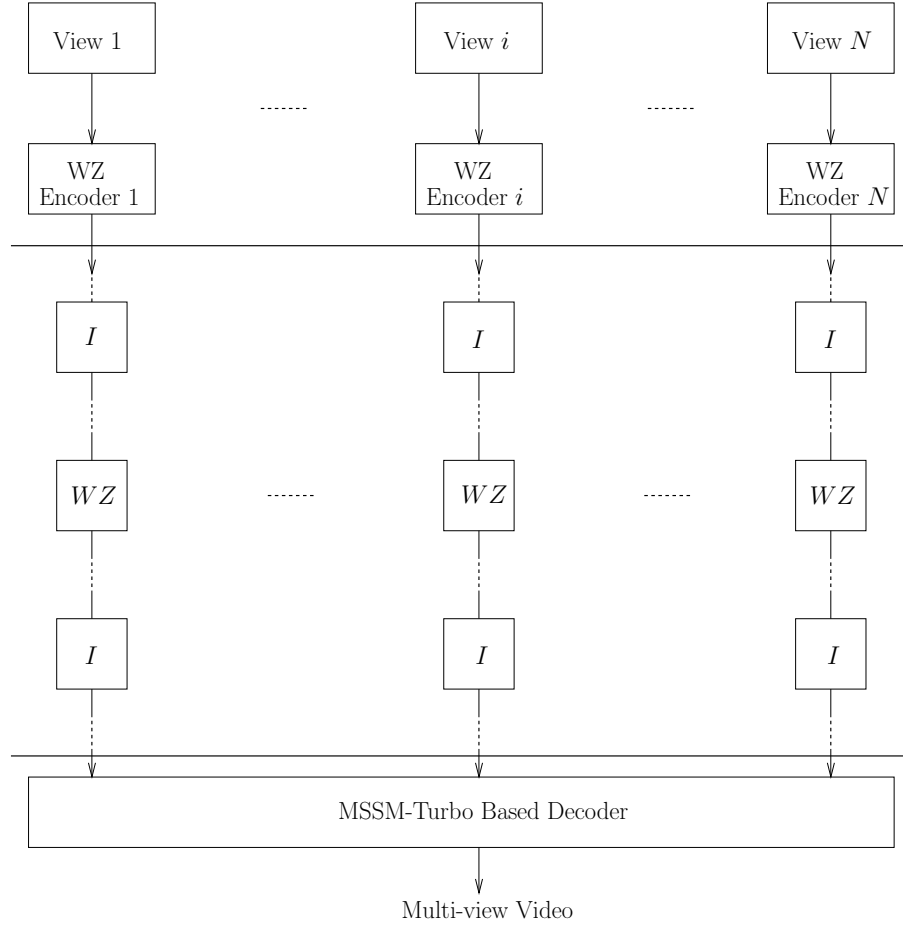


Figure 5.9: The schematic compression process of Wyner-Ziv coding conceived for multi-view video

Generally, this treatise focuses on the MSSM-Turbo decoder of Figure 5.10, while the rest of the techniques, including the WZ encoder, the motion-compensated frame interpolation (MCFI) etc., are detailed in [188]. Below, we will briefly introduce the system of Figure 5.10, while the MSSM-Turbo decoder will be detailed in Section 5.4.2.

5.4.1.1 Transmitter

For each view, the frames are classified into two categories, namely the so-called key frames U_1, \dots, U_N of Figure 5.10 and the WZ frames V_1, \dots, V_N of Figure 5.10. The key frames U_1, \dots, U_N , also referred to as I-frames, are intra-frame-coded by the H.264/AVC encoder and then transmitted to the receiver. Below we consider the monoscopic WZ frame V_i ($1 \leq i \leq N$) for introducing the encoding process of the WZ frames in Figure 5.10. The encoding process is encapsulated in the following steps

- The monoscopic frame V_i is quantized by a uniform quantizer at the transmitter of Figure 5.10 generating the resultant m -bit monoscopic frame q_i .

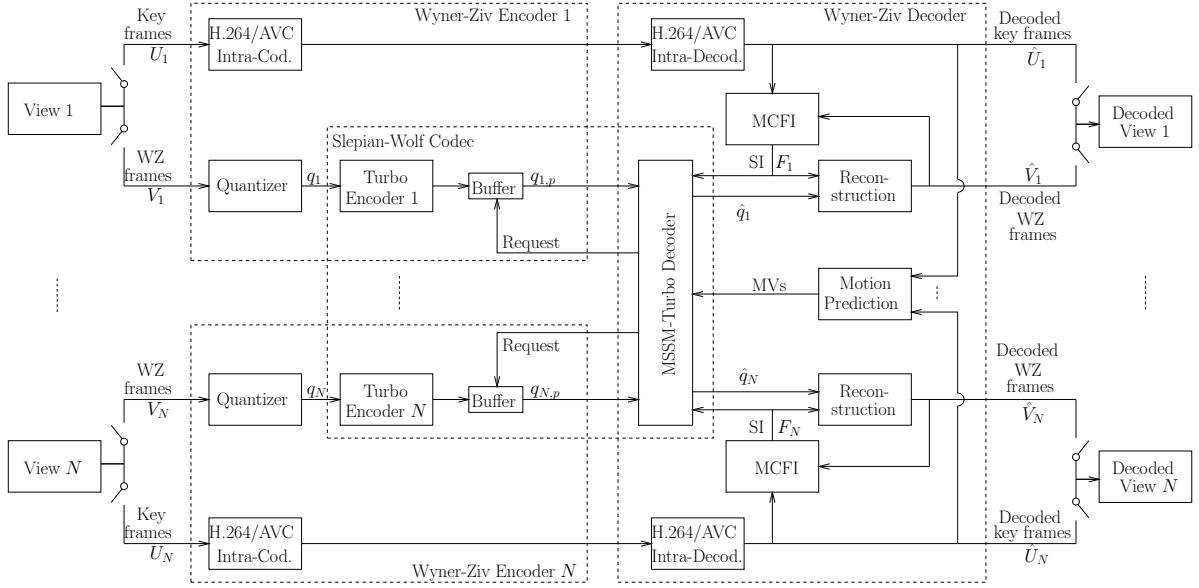


Figure 5.10: System architecture of Wyner-Ziv compression invoked for MVC using MSSM-Turbo decoding. The MSSM-Turbo decoder is employed for multi-view sequences, as opposed to the PDWZ systems of Figures 5.1 and 5.4 used for monoscopic video.

- Each pixel of the m -bit quantized monoscopic frame q_i is decomposed into m bits. Then the bits from the same position of q_i constitute the most significant bit (MSB) plane resulting in m MSB planes. Afterwards, each of the m MSB planes will be considered as a block input, which is encoded by the turbo encoder i of Figure 5.10.
- The systematic output of the turbo encoder i is directly transmitted to the receiver. The parity bits $q_{1,p}, \dots, q_{N,p}$ ⁴ of Figure 5.10 generated by the turbo encoder i for the different MSB planes of the WZ frame V_i may be independently buffered at the transmitter, which will then be transmitted to the receiver upon its request.

5.4.1.2 Receiver

The behavior of the receiver is described in Figure 5.11, which will be further detailed in Section 5.4.2. At the receiver, the received bitstreams of the key frames representing the N camera views may be independently decoded for reconstructing the key frames $\hat{U}_1, \dots, \hat{U}_N$, which will then be utilized for estimating the SI for the corresponding WZ frames. Then the decoding process is listed as follows

- For each monoscopic WZ frame, two temporally adjacent monoscopic key frames may be utilized for predicting the related soft-bit information. A number of algorithms have been proposed for this estimation process, such as for example

⁴The subscript p indicates that $q_{1,p}, \dots, q_{N,p}$ are parity bits of quantized pixels q_1, \dots, q_N .

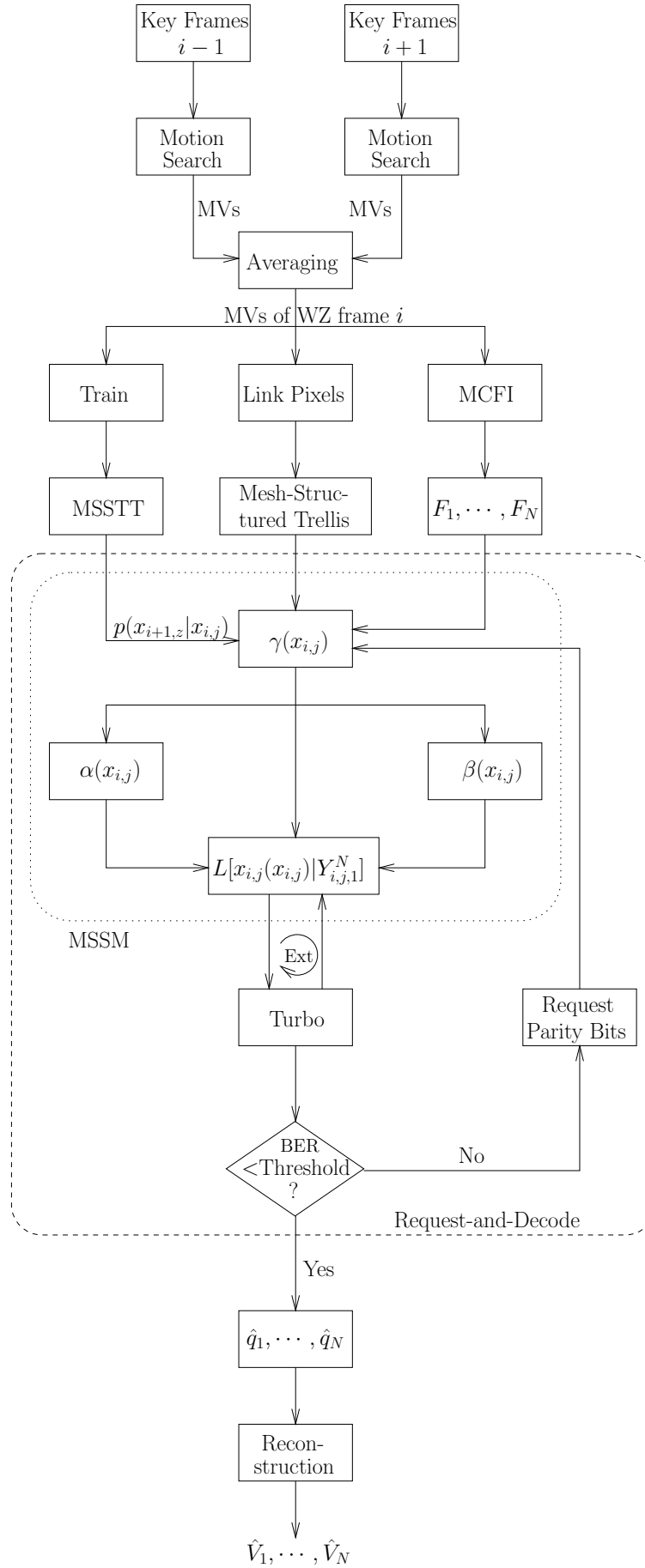


Figure 5.11: Flow-chart of the receiver.

the MCFI framework [188] shown in Figure 5.10, which is invoked in our system for predicting the SI in the temporal direction.

- As observed from Figure 5.11, the SI generated for the WZ frames and the received parity bits of the MSB planes related to the WZ frames will be used by the MSSM-Turbo decoder of Figure 5.10 for reconstructing the WZ frames $\hat{V}_1, \dots, \hat{V}_N$.
- The flow-chart of Figure 5.11 shows that when the MSSM-Turbo decoder fails to perfectly recover a MSB plane, the receiver will send a feedback flag to the transmitter for requesting more parity bits for this particular MSB plane. Again, this process is referred to as "request-and-decode"⁵ process [105] in Figure 5.11. The MSSM-Turbo decoder will be detailed in Section 5.4.2.
- As observed from Figure 5.11, the "request-and-decode" process will terminate, when the bit error ratio (BER) of the MSB plane becomes lower than a preset threshold. Once all MSB planes have been decoded, the current multi-view frame may be readily reconstructed.

5.4.2 Inter-view Correlation Modelling

The maximum *a-posteriori* probability (MAP) determination technique conceived for first-order Markov processes was detailed in Section 4.2, where the intra-frame correlation was exploited using both horizontal and vertical Markov processes. However, the first-order Markov processes cannot be readily applied for modelling the inter-view correlation. In this section, we introduce the techniques invoked for exploiting the inter-view correlation of the N cameras by designing a novel trellis representation and derive its decoding rules. Below we focus on a specific multi-view WZ frame, which consists of N monoscopic video frames, whose SI is estimated by the MCFI of Figure 5.10. Moreover, the trellis generation, MSSM-Turbo decoding and "request-and-decode" processes are illustrated in the flow-chart of Figure 5.11. Let us commence by introducing the following notations:

- N : the number of camera views in the multi-view video;
- V_i : the original monoscopic frame of the camera view i ;
- F_i : the SI of the frame V_i , which is estimated by the MCFI block of Figure 5.10;
- $x_{i,j}$: the j^{th} pixel⁶ in V_i , namely the pixel at position (i, j) ;
- $x_{i,j}(k)$: the k^{th} bit of the pixel $x_{i,j}$;
- $y_{i,j}$: the SI of the pixel $x_{i,j}$, which is at position j of F_i ;

⁵Rate controller at the encoder side [114] may be investigated to avoid this feedback channel.

⁶The pixels of a 2D frame are indexed using a one-dimensional formulation, assuming that the 2D image is scanned into a single-dimensional vector.

- m : the number of bits contained in the pixel $x_{i,j}$;
- $X_m = \{0, 1, \dots, 2^m - 1\}$: the set of legitimate values of an m -bit pattern $x_{i,j}$;
- $L_{i,j,k}$: the set of the original pixels linked with $x_{i,j}$ in camera view k .
- $L_{i,j,1}^N$: notation for the set $\bigcup_{t=1}^N L_{i,j,t}$;
- $Y_{i,j,k}$: the corresponding SI of $L_{i,j,k}$;
- $Y_{i,j,1}^N$: notation for the set $\bigcup_{t=1}^N Y_{i,j,t}$.

5.4.2.1 Mesh-Structured Trellis Representation

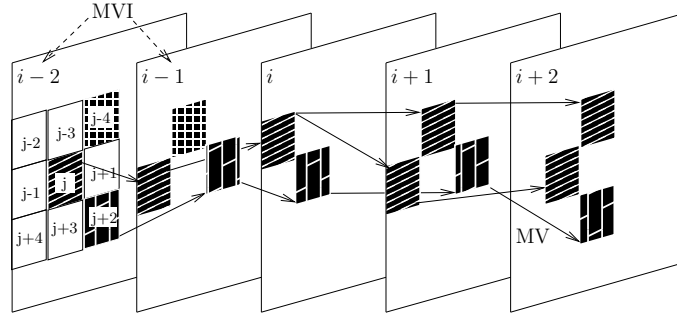


Figure 5.12: Motion based inter-view correlation, where $(j - 4) \dots (j + 4)$ indicate the indices of the (1×1) -pixel MBs.

Again, each multi-view frame consists of N monoscopic camera view frames. Since the current multi-view WZ frame is not available at the receiver, the inter-view MVs of the current WZ frame are estimated from the adjacent key frames. Note that all the key frames are available at the receiver after the "H.264/AVC intra decoder" block of Figure 5.10. Here we consider the scenario of $\text{GOP}=2$, but this technique may be readily extended to larger GOP scenarios. Specifically, the $(i - 1)^{st}$ and $(i + 1)^{st}$ key frames are utilized for estimating the MVs of the i^{th} WZ frame, as illustrated in the flow-chart of Figure 5.11. The inter-view MVs of the WZ multi-view frames are generated as follows

- The inter-view MVs of the $(i - 1)^{st}$ and $(i + 1)^{st}$ key frames are readily estimated using the traditional macroblock (MB) based motion search techniques [194], which is indicated by the "Motion Search" blocks of Figure 5.11.
- The inter-view MVs of the i^{th} WZ frame is averaged based on that of the $(i - 1)^{st}$ and $(i + 1)^{st}$ key frames, as illustrated by the "Averaging" block of Figure 5.11.

The structure of the inter-view "Motion Prediction" of Figure 5.10 is shown in Figure 5.12, where the MBs of the specific view associated with the Monoscopic View Index (MVI) i ($2 < i \leq I$) are estimated from the views $(i - 1)$ and $(i + 1)$. Note that in Figure 5.12 we employ (1×1) -pixel "MBs" for the sake of simplifying the relevant descriptions, which the classic (8×8) -pixel MBs may be readily extended to.

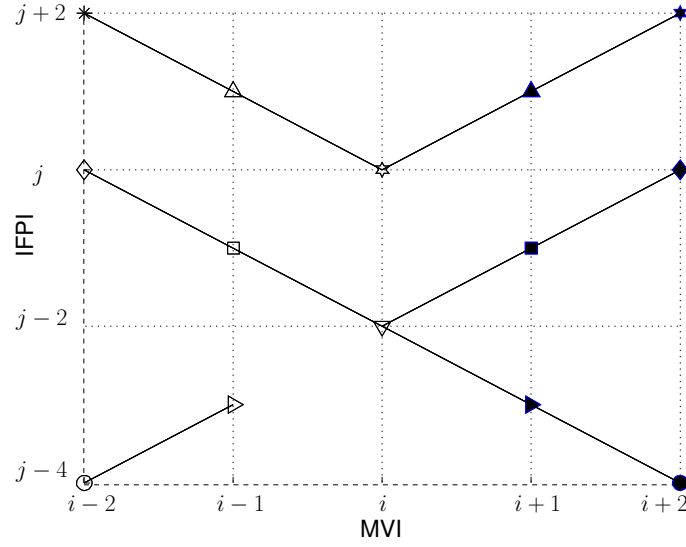


Figure 5.13: The inter-view pixel correlation graph of Figure 5.12, where each marker indicates a different pixel.

Following the inter-view motion estimation process at the receiver, the pixels of a given MB of Figure 5.12 may be linked to other pixels in the same position of the predicted MBs in the other views, as exemplified in the inter-view pixel correlation graph of Figure 5.13. Specifically, the graph of Figure 5.13, indicated by the “Link Pixels” block of Figure 5.11, may be created using the following steps:

- The pixels of Figure 5.12 are scanned into one-dimensional Intra-Frame Pixel Indices (IFPI), namely $(j-4), \dots, (j+4)$ in view $(i-2)$ of Figure 5.12. For example, each monoscopic frame having (352×288) -pixels will be indexed by $352 \cdot 288 = 72864$ one-dimensional scan-line indices.
- The coordinate axes of Figure 5.13 are created by arranging for the x axis and y axis to indicate the MVI and the IFPI, respectively. Specifically, the MVI of Figure 5.12 are in the range of $(i-2, \dots, i+2)$. Furthermore, we only consider the colored pixels of Figure 5.12 for simplifying the related descriptions, resulting in IFPI of $(j-4, \dots, j+2)$.
- Connect all the correlated pixels portrayed in Figure 5.13 using a link, where the presence of highest correlation is indicated by the MVs of Figure 5.12. For example, in Figure 5.13 the pixel at position (i, j) is correlated with the pixels at positions of $(i-1, j+1)$ and $(i+1, j+1)$ in Figure 5.13, as indicated by the MVs of Figure 5.12.

Based on the two-dimensional Markov-modelling based trellis representation developed in [184], the mesh-structured trellis of Figure 5.14 may be derived from the correlation graph of Figure 5.13. This trellis generation process is also indicated in Figure 5.11, which is completed using the following steps:

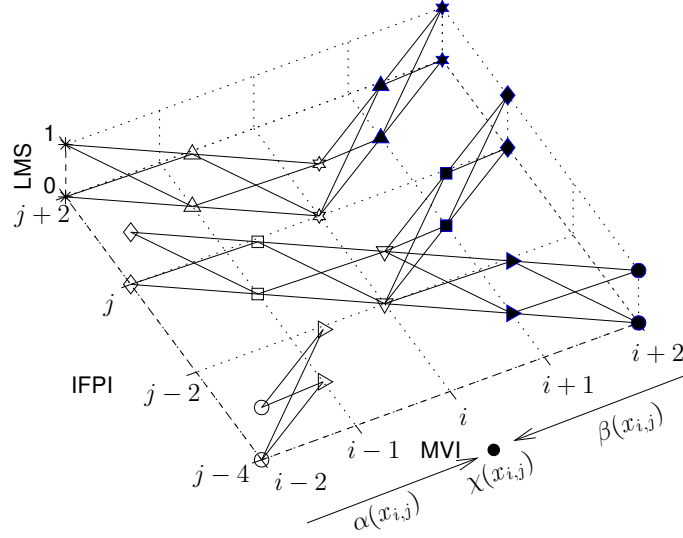


Figure 5.14: Mesh-structured trellis representation of inter-view correlations for $m = 1$ -bit pixels, where the pixels of view $(i - 2)$ to view $(i + 2)$ are displayed.

- Each m -bit pixel has a value in the range of $[0, 2^m)$, which we refer to as the Legitimate Markov States (LMS). Then each pixel in Figure 5.13 has 2^m LMS. Hence by introducing the z axis indicating the LMS, Figure 5.14 may be created, where the x axis and y axis indicate the MVI and IFPI, respectively.
- Each pair of pixels connected by a direct link in Figure 5.13 represents a correlated Markov-state transition. Hence we have to incorporate $(2^m \times 2^m)$ links indicating the $(2^m \times 2^m)$ possible Markov-state transitions for these pixels in Figure 5.13. Specifically, for the pixel $x_{i,j}$ within view i , the transition probability from state $x_{i,j}$ to the correlated successor state $x_{i+1,j+1}$ within view $(i + 1)$ is represented by $p(x_{i+1,j+1}|x_{i,j})$, which is the state transition of the related Markov process.

Hence all nodes in Figure 5.14 associated with identical MVI belong to the same view, while the nodes having both an identical MVI and IFPI values are Markov states for a same corresponding pixel. Furthermore, the trellis representation seen in Figure 5.14 may be readily generalized both for arbitrary m -bit pixel multi-view signals and for an arbitrary number of views, which leads to 2^m LMS.

5.4.2.2 Trellis Decoding

Section 5.4.1 detailed how the SI of a multi-view WZ frame can be generated. The SI consists of floating-point values indicating the reliability of specific pixel values, which is estimated by the MCFI block of Figure 5.10. For example, the SI of the bit

$x_{i,j}(k)$ may be expressed in the log-likelihood ratio (LLR) format as

$$L[x_{i,j}(k)] = \ln \frac{L[x_{i,j}(k) = 0]}{L[x_{i,j}(k) = 1]}, \quad (5.1)$$

while the reliability of the pixel $x_{i,j}$ can be represented by $L[x_{i,j}(0)], \dots, L[x_{i,j}(m-1)]$. In Section 5.4.2.1, the pixel correlations of a multi-view WZ frame are modelled by the mesh-structured trellis of Figure 5.14. Hence the SI of the multi-view WZ frame may be refined by decoding the trellis of Figure 5.14, where the classic Bahl-Cocke-Jelinek-Raviv (BCJR) [183] decoding principle may be applied.

Given the SI of the current multi-view WZ frame F_1, \dots, F_N , the pixel $x_{i,j}$ may be estimated by the *a-posteriori* probability (APP) $p(x_{i,j}|F_1, \dots, F_N)$. Furthermore, by employing our proposed MSSM detailed in Section 5.4.2.1, we have

$$p(x_{i,j}|F_1, \dots, F_N) \approx p(x_{i,j}|Y_{i,j,1}^N), \quad (5.2)$$

where $Y_{i,j,1}^N$ includes the SI of the pixels $L_{i,j,1}^N$, which are correlated with $x_{i,j}$. Consider the trellis of Figure 5.14 for example, where we have $L_{i,j,i-2} = \{x_{i-2,j+2}\}$, $L_{i,j,i-1} = \{x_{i-1,j+1}\}$, $L_{i,j,i} = \{x_{i,j}\}$, $L_{i,j,i+1} = \{x_{i+1,j+1}\}$ and $L_{i,j,i+2} = \{x_{i+2,j+2}\}$, the pixels $L_{i,j,i-2}^{i+2}$ can be readily derived as $L_{i,j,i-2}^{i+2} = \bigcup_{t=i-2}^{i+2} L_{i,j,t}$. In the Appendix, we show that the log-likelihood ratio (LLR) version of the APP $p(x_{i,j}(k)|Y_{i,j,1}^N)$ for the bit $x_{i,j}(k)$, $0 \leq k < m$ may be formulated as

$$L[x_{i,j}(k)|Y_{i,j,1}^N] = \ln \frac{\sum_{\substack{x_{i,j} \in X_m \\ x_{i,j}(k)=0}} \beta(x_{i,j}) \cdot \chi(x_{i,j}) \cdot \alpha(x_{i,j})}{\sum_{\substack{x_{i,j} \in X_m \\ x_{i,j}(k)=1}} \beta(x_{i,j}) \cdot \chi(x_{i,j}) \cdot \alpha(x_{i,j})}, \quad (5.3)$$

where $\beta(x_{i,j})$, $\chi(x_{i,j})$, $\alpha(x_{i,j})$ are defined in Eqs. (B.9), (B.3) (B.8). Specifically, $\beta(x_{i,j})$, $\alpha(x_{i,j})$ indicate the backward and forward oriented probability of the pixel $x_{i,j}$, while $\chi(x_{i,j})$ is the channel information of the pixel $x_{i,j}$.

5.4.2.3 Iterative MSSM-Turbo Decoding

A limitation of the formulas provided in Section 5.4.2.2 is that they cannot be directly used for iterative decoding, since they cannot exploit the *a-priori* LLR $L[x_{i,j}(k)]$, which was generated from the *extrinsic* information gleaned from the other decoder components. To make use of the *a-priori* LLR $L[x_{i,j}(k)]$, the combined bit-wise LLR

may be expressed as [87, 184]

$$\gamma_{i,j}(x_{i,j}) = \exp \sum_{k=0}^{m-1} \frac{\bar{x}_{i,j}(k)}{2} \cdot \{L[x_{i,j}(k)] + L[y_{i,j}(k)|x_{i,j}(k)]\}, \quad (5.4)$$

where the symbol-based m -bit information γ is the combination of the bit-wise *a-priori* LLR $L[x_{i,j}(k)]$ and of the channel information $L[y_{i,j}(k)|x_{i,j}(k)]$. We note in this context that γ of Eq. (5.4) contains more valuable information than the channel information χ . Hence Eq. (5.3) may be used for iterative joint source-channel decoding by replacing χ with γ in Eq. (5.4).

Similar to the BCJR decoding technique of classic turbo codes [78], the bit-based *a-posteriori* LLR $L[x_{i,j}(k)|Y_{i,j,1}^N]$ may be split into three components, namely the *a-priori* LLR $L[x_{i,j}(k)]$, the channel information $L[y_{i,j}(k)|x_{i,j}(k)]$ and the extrinsic information $L_e[x_{i,j}(k)]$. Specifically, the bit-based *a-posteriori* LLR $L[x_{i,j}(k)|Y_{i,j,1}^N]$ may be formulated as

$$\begin{aligned} & L[x_{i,j}(k)|Y_{i,j,1}^N] \\ &= L[x_{i,j}(k)] + L[y_{i,j}(k)|x_{i,j}(k)] + \ln \frac{\sum_{\substack{x_{i,j} \in X_m \\ x_{i,j}(k)=0}} \beta(x_{i,j}) \cdot \gamma^{[ext]}[x_{i,j}(k)] \cdot \alpha(x_{i,j})}{\sum_{\substack{x_{i,j} \in X_m \\ x_{i,j}(k)=1}} \beta(x_{i,j}) \cdot \gamma^{[ext]}[x(k)] \cdot \alpha(x_{i,j})}, \end{aligned} \quad (5.5)$$

where the extrinsic information component $\gamma^{[ext]}[x_{i,j}(k)]$ may be expressed as

$$\gamma^{[ext]}[x_{i,j}(k)] = \exp \sum_{l=0, l \neq k}^{m-1} \frac{\bar{x}_{i,j}(l)}{2} \cdot \{L[x_{i,j}(l)] + L[y_{i,j}(l)|x_{i,j}(l)]\}.$$

Based on the above, the MSSM-Turbo decoding architecture of Figure 5.10 is shown in Figure 5.15, which terminates after I_{iter} iterations of extrinsic information exchange. Moreover, the MSSM-turbo decoding process is also detailed in the flow-chart of Figure 5.11.

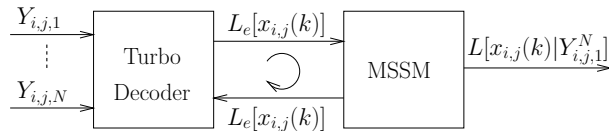


Figure 5.15: The MSSM-Turbo decoding architecture of Figure 5.10

5.4.2.4 Training for Mesh-Structured State Transition

Again, the mesh-structured trellis of Figure 5.14 is utilized for modelling the inter-view correlation at the receiver. Specifically, the component $p(x_{i+1,z}|x_{i,j})$ of Eqs. (B.8), (B.9) quantifies the inter-view correlation, where z and j indicate the pixel index of the views $(i-1)$ and i , respectively. Here we refer to $p(x_{i+1,z}|x_{i,j})$ as the Mesh-Structured State Transition Table (MSSTT) for simplicity. We initialize the $(2^m \times 2^m)$ -element MSSTT $T[0:2^m-1, 0:2^m-1]$ to zero values. Then we scan all the linked pixel pairs of Figure 5.13. For example, when the pixel pairs s_{i-1} and s_i are scanned, the corresponding element $T[s_{i-1}, s_i]$ in the MSSTT is increased by 1. Finally, by normalizing the summation of all rows in the MSSTT $T[0:2^m-1, 0:2^m-1]$, the transition probabilities $p(x_{i+1,z}|x_{i,j})$ can be obtained, where we have $x_{i,j}, x_{i+1,z} \in [0, 2^m)$.

5.4.2.5 Complexity Analysis

The complexity of our proposed MSSM can be attributed to the calculation of $\gamma(x_{i,j})$ in Eq. (5.4), $\alpha(x_{i,j})$ in Eq. (B.8), $\beta(x_{i,j})$ in Eq. (B.9) and $L[x_{i,j}(k)|Y_{i,j,1}^N]$ in Eq. (5.3). As shown in Figure 5.14, the trellis size is $(2^m \cdot S \cdot N)$, where S is the number of pixels in each monoscopic view. Similar to the BCJR decoding rules proposed in [183], the decoding of the $(2^m \cdot S \cdot N)$ -state trellis of Figure 5.14 may be generalized into the following two stages:

- Calculation of γ , α and β : These operations are carried out across the entire trellis of Figure 5.14, which imposes the complexity of m , 2^m , 2^m for each trellis state, as suggested by Eqs. (5.4), (B.8), (B.9), respectively. Hence the associated computational costs are on the order of $O(2^m \cdot S \cdot N \cdot m)$, $O(2^{2m} \cdot S \cdot N)$ and $O(2^{2m} \cdot S \cdot N)$ for γ , α and β , respectively.
- Calculation of $L[x_{i,j}(k)|Y_{i,j,1}^N]$: This operation is carried out for all the $(S \cdot N \cdot m)$ bits of a multi-view frame, which imposes a complexity of 2^m for each bit. Hence the computational cost is on the order of $O(2^m \cdot S \cdot N \cdot m)$.

Therefore, the overall complexity imposed by our proposed decoder is

$$O(2 \cdot 2^{2m} \cdot S \cdot N + 2 \cdot 2^m \cdot S \cdot N \cdot m), \quad (5.6)$$

when decoding a multi-view frame.

5.4.3 Performance Study

In this section, we present our simulation results for benchmarking the scheme introduced in Section 5.4.1. Firstly, in Section 5.4.3.1 we will introduce the parameters of the scenario considered in our experiment. Then we will discuss our numerical results in Section 5.4.3.2.

5.4.3.1 Scenario

Representation	YUV 4:2:0	Motion MB	8×8
Format	CIF	Generator of RSC	[11011,10011]
Bits Per Pixel	8	Turbo Code Rate	$1/2$
Number of Frames	100	Interleaver Length	352×288
GOP	2, 4	Puncturer	[1 1; 0 1; 1 0]
Number of Views	8	Intra-codec	H.264

Table 5.3: Table of parameters employed for the simulations of the schematic seen in Figure 5.10.

In this section, we present our experimental parameters used for characterizing the convergence behavior of the proposed scheme introduced in Section 5.4.1. Multi-view video sequences having 8 camera views represented in (352×288) -common intermediate format (CIF)⁷ and 4:2:0 YUV representation are employed. Moreover, the bitrate/PSNR of both the WZ and the key frames was taken into account in our average results. The distributed WZ coding scheme conceived for the multi-view video scheme of Figure 5.10 operates on the basis of (352×288) -pixel blocks. More specifically, a specific MSB plane of each view is input to the turbo encoder, which consists of (352×288) bits. In other words, the interleaver length of our turbo codec is (352×288) bits. In [188], each bitplane of the MSB was transmitted separately and each bitplane was then refined based on the previously decoded bitplanes [193]. However, in our system, all MSB planes were transmitted together, which allowed us to reduce the number of "request-and-decode" processes defined in [103]. We employ a recursive systematic convolutional (RSC) encoder relying on the generator polynomials of $g_1 = 11011$, $g_2 = 10011$, which are represented as $G = [1, g_2/g_1]$, where g_1 is a feedback input and g_2 is feed-forward output. Moreover, two identical RSC encoders are employed for the turbo codec and the puncturing matrix of [1 1; 0 1; 1 0]

⁷We converted the multi-view video sequences into CIF representation for the sake of speeding up our simulations.

is employed for the turbo code. The parameters employed are listed in Table 5.3. The remaining parameters of our system were identical to those in [100, 188].

	Turbo	MSSM-Turbo
Dimension	1	2
Number of Bits to Decode	352×288	$352 \times 288 \times 8 \times n_{\text{MSB}}$
Side Information	None	$1 \times \text{MMSTT}$

Table 5.4: Comparison of MSSM-Turbo in schematic of Figure 5.10 and the benchmark turbo for Wyner-Ziv coding of multiview video, where n_{MSB} indicates the number of MSB planes. $I_{\text{iter}} = 4$ is employed, which indicates the number of decoding iterations for the MSSM-Turbo decoder.

Let us now compare the proposed MSSM-Turbo decoder to the classic turbo decoder. The turbo decoder is invoked for each MSB plane of each monoscopic frame, which carries (352×288) bits, scanned into a one-dimensional vector. *Let n_{MSB} be the number of MSB planes.* The MSSM-Turbo decoder is invoked for all MSB planes of each multi-view frame, which carries $(352 \times 288 \times 8 \times n_{\text{MSB}})$ bits and it is arranged into $I = 8$ scanlines or vectors. Furthermore, the MSSM-Turbo scheme relies on the corresponding MSSTT, which has $(2^{n_{\text{MSB}}} \times 2^{n_{\text{MSB}}})$ elements. Specifically, for the simulations associated with $m = 4$, the size of the MSSTT SI was (16×16) floating-point values for 100 CIF frames. Our detailed comparisons are listed in Table 5.4. Note that the MSSTT may be estimated from the adjacent I-frames of the current WZ frame.

5.4.3.2 Numerical Results

In this section, we present our numerical results for benchmarking the proposed MSSM-PDWZ scheme's performance against that of the PDWZ system of [100] for four multi-view sequences associated with different motion features, namely the *Newspaper* sequence, the *Leaving-laptop* sequence [195], the *Outdoor* sequence [195] and the *Ballroom* sequence [196], scanned in FPSs of 30, 16.67, 16.67, 25, respectively. The rate-distortion (RD) results recorded for both the PDWZ and MSSM-PDWZ schemes in Figure 5.10 were parameterized by the number of WZ coded bitplanes for $m = 1, 2, 3$ or 4 MSB planes, because this configuration has been widely adopted in the pixel-domain WZ video coding literature [100]. This was arranged by invoking the uniform quantizers shown in Figure 5.10. The RD results of the test sequences

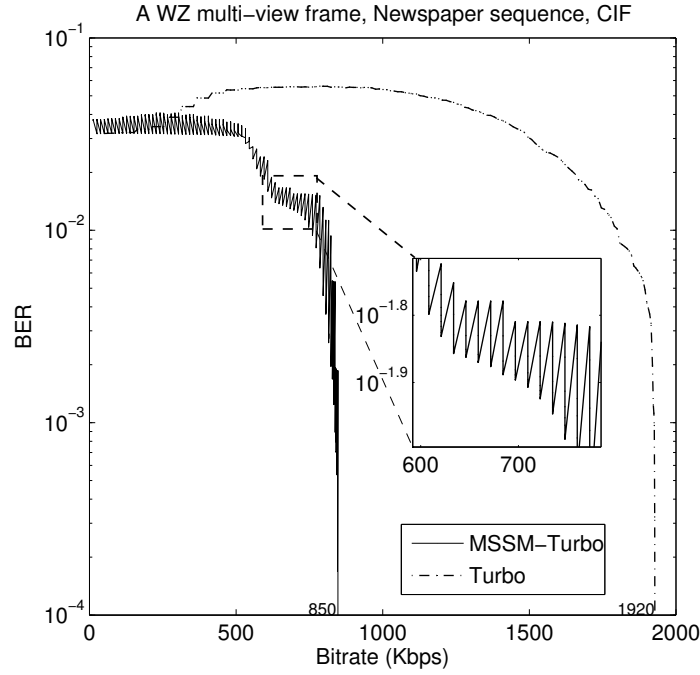


Figure 5.16: BER versus bitrate comparison of the MSSM-turbo and of turbo codec for a multi-view WZ frame, where only the bits of the WZ frame are taken into account as the bitrate. Using the *Newspaper* sequence and GOP=2. The schematic of Figure 5.10 configured with the parameters in Table 5.3 was used.

coded by the H.264/AVC codec [14] are provided below as usual, both in the associated intra-frame encoding mode and in the motion compensation dispensed mode in conjunction with GOP periods of 2 and 4. Both these modes were selected by appropriately adjusting the encoding parameters of the H.264/AVC reference software JM [197]. Furthermore, the performance of the multi-view video codec JMVC operating without motion estimation and using GOP periods of 2 and 4 was also provided.

5.4.3.2.1 BER Characteristics The BER comparison of the MSSM-turbo and turbo codec schemes is displayed in Figure 5.16, where the x axis represents the bitrate of the multi-view WZ frame. Note that the bitrate of a video sequence consists of the bitrate of the WZ frames and the bitrate of the key frames. Here we only count the bitrate of the WZ frames for the sake of providing further insights into our system's behavior.

For the MSSM-turbo decoder, the BERs of both the MSSM and of the turbo decoder components are provided. More specifically, multiple BER values are plotted for the MSSM-turbo decoder for each "Bitrate" abscissa value in Figure 5.16. This results in a wave-shaped, fluctuating curve for the MSSM-turbo decoder. Alternatively, for each "Bitrate" value, the MSSM-turbo is capable of further reducing

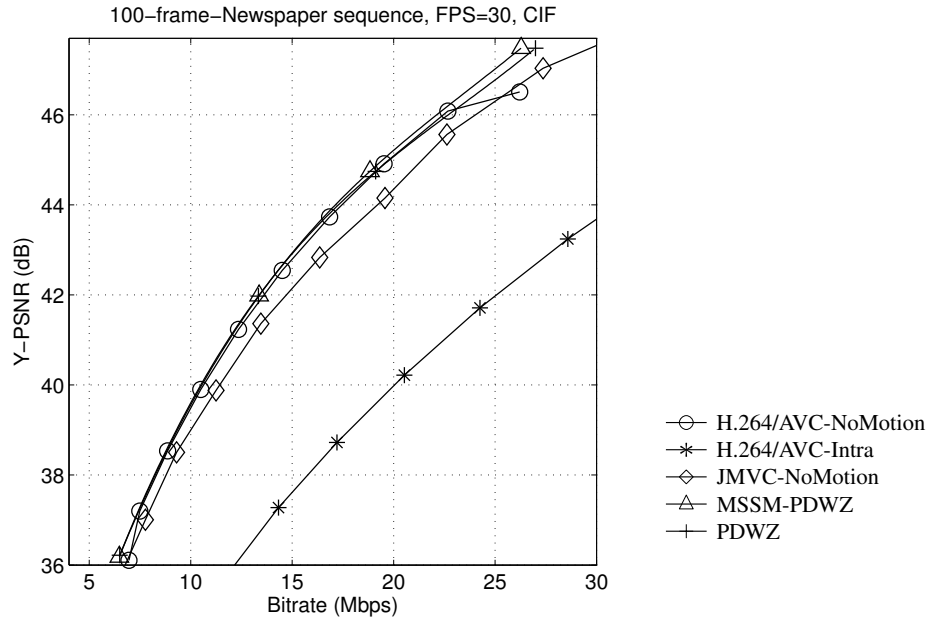


Figure 5.17: Rate-distortion performance comparison of the MSSM-PDWZ codec and the benchmarks. The schematic of Figure 5.10 configured with the parameters in Table 5.3 was used. Newspaper sequence, GOP=2

Sequence	GOP=2		GOP=4	
	ΔR (%)	$\Delta PSNR$ (dB)	ΔR (%)	$\Delta PSNR$ (dB)
Newspaper	0.3	0.02	1.8	0.17
Leavinglaptop	6.7	0.42	15.6	0.92
Outdoor	12.6	0.97	30.7	2.27
Ballroom	13.4	0.80	27.2	1.58

Table 5.5: Bjøntegaard [198] comparison of the MSSM-PDWZ versus the PDWZ schemes for the considered multi-view sequences organized in GOPs of 2 and 4. “R” indicates “Rate”. The schematic of Figure 5.10 configured with the parameters in Table 5.3 was used.

the BER upon the turbo codec during the ISCD process. Observe from Figure 5.16 that BER of the WZ frame using the MSSM-turbo decoder becomes vanishingly low at about 850 Kbps, while that of the turbo decoder vanishes at 1920 Kbps. The MSSM-turbo decoder requires 1070 less Kbps for achieving a BER of 10^{-4} . This observation is due to the fact that MSSM is capable of further reducing the BER by exploiting the residual redundancy within the WZ frame by iteratively exchanging extrinsic information with the turbo decoder.

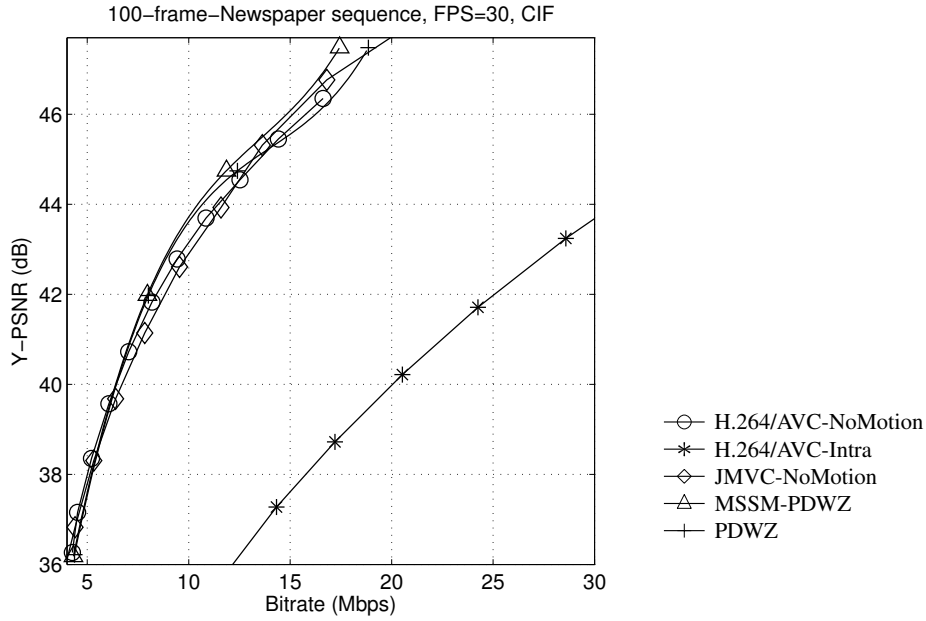


Figure 5.18: Rate-distortion performance comparison of the MSSM-PDWZ codec and the benchmarks. The schematic of Figure 5.10 configured with the parameters in Table 5.3 was used. Newspaper sequence, GOP=4

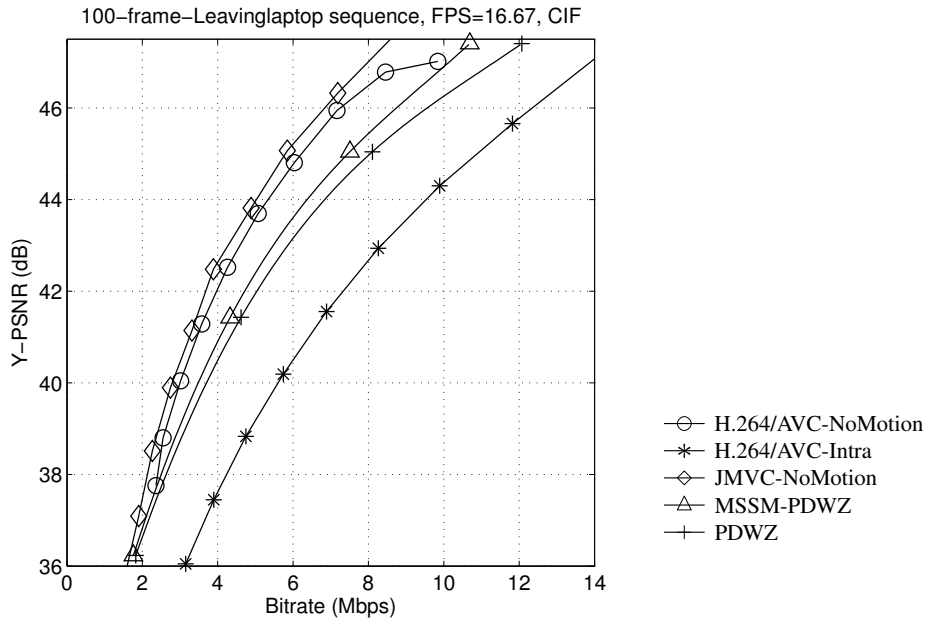


Figure 5.19: Rate-distortion performance comparison of the MSSM-PDWZ codec and the benchmarks. The schematic of Figure 5.10 configured with the parameters in Table 5.3 was used. Leavinglaptop sequence, GOP=2

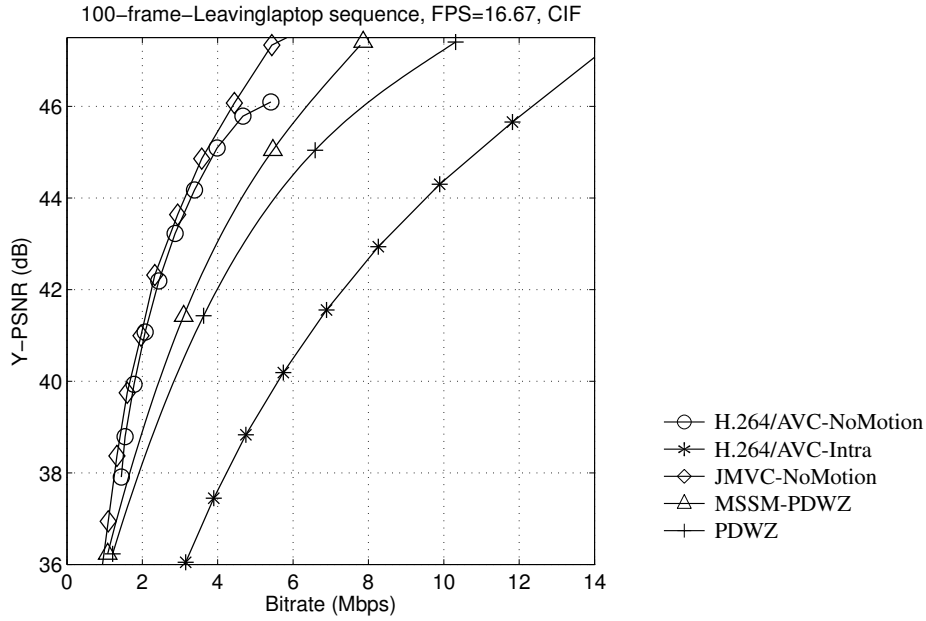


Figure 5.20: Rate-distortion performance comparison of the MSSM-PDWZ codec and the benchmarks. The schematic of Figure 5.10 configured with the parameters in Table 5.3 was used. Leavinglaptop sequence, GOP=4

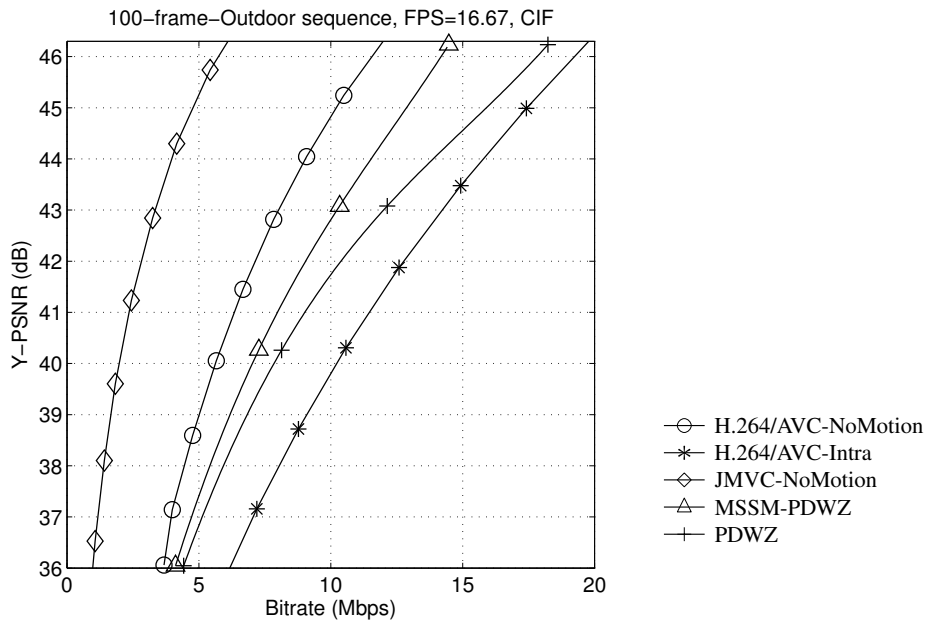


Figure 5.21: Rate-distortion performance comparison of the MSSM-PDWZ codec and the benchmarks. The schematic of Figure 5.10 configured with the parameters in Table 5.3 was used. Outdoor sequence, GOP=2

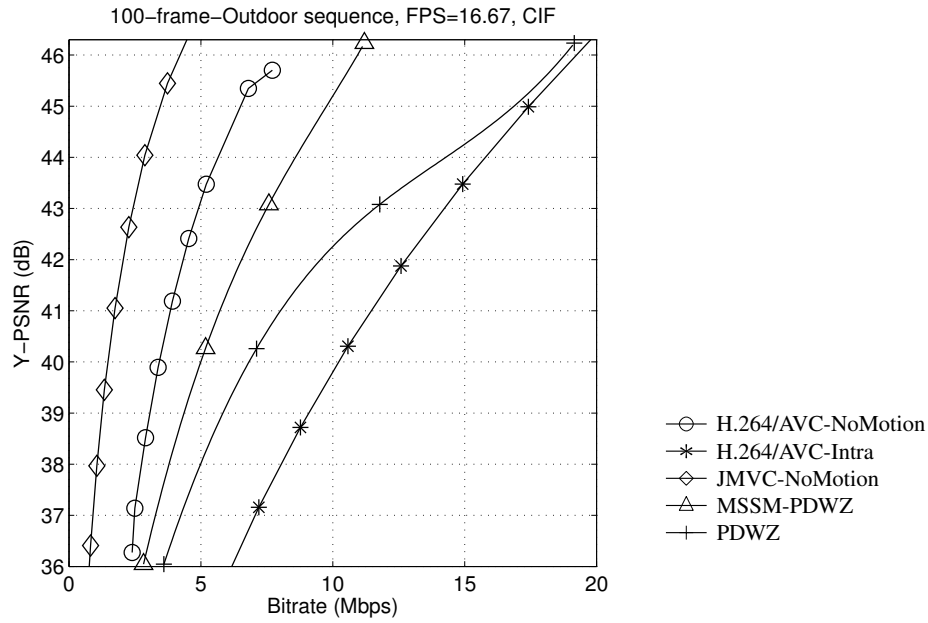


Figure 5.22: Rate-distortion performance comparison of the MSSM-PDWZ codec and the benchmarks. The schematic of Figure 5.10 configured with the parameters in Table 5.3 was used. Outdoor sequence, GOP=4

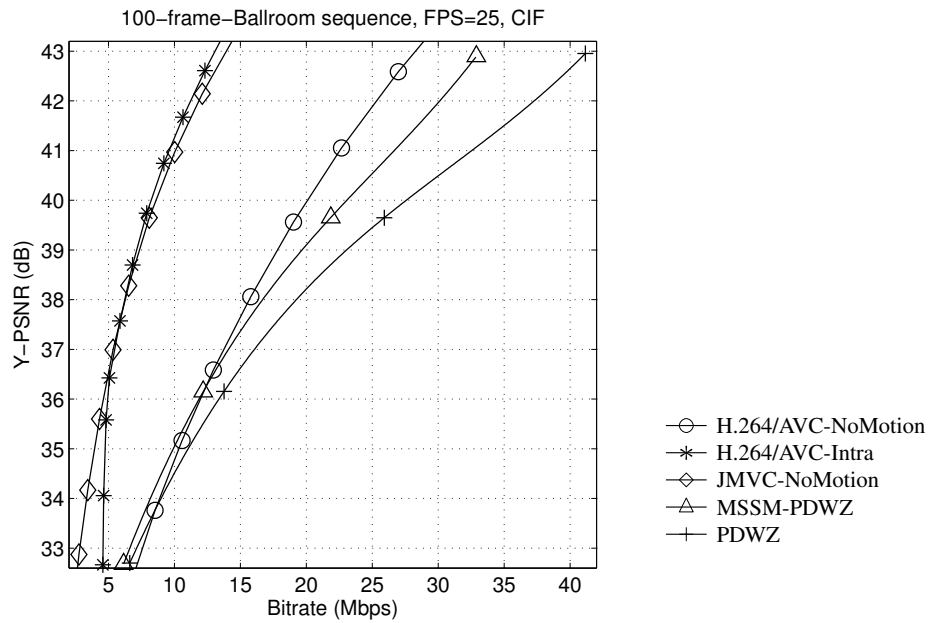


Figure 5.23: Rate-distortion performance comparison of the MSSM-PDWZ codec and the benchmarks. The schematic of Figure 5.10 configured with the parameters in Table 5.3 was used. Ballroom sequence, GOP=2

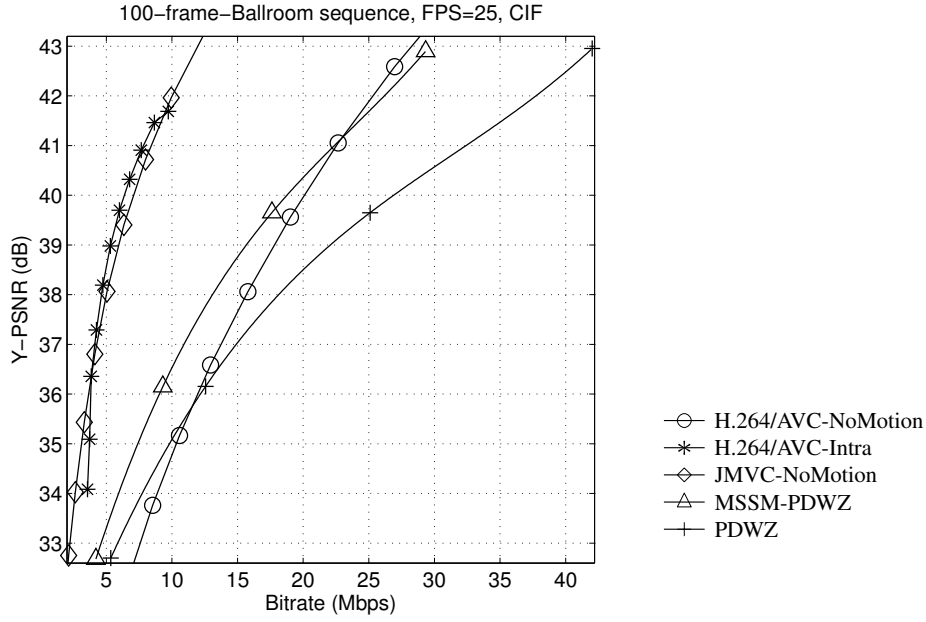


Figure 5.24: Rate-distortion performance comparison of the MSSM-PDWZ codec and the benchmarks. The schematic of Figure 5.10 configured with the parameters in Table 5.3 was used. Ballroom sequence, GOP=4

5.4.3.2.2 Rate-distortion Characteristics The simulation results recorded for the four sequences are displayed in Figures 5.17-5.24, while the Bjøntegaard [198] comparison of the MSSM-PDWZ versus the PDWZ schemes for the four multi-view sequences considered and organized in GOPs of 2 and 4 is provided in Table 5.5. For the MSSM-PDWZ, Figures 5.17-5.24 show that the GOP=4 scenarios outperforms the GOP=2 regime for the Newspaper, Leavinglaptop and Outdoor sequences, while the opposite trends were observed for the Ballroom sequence. This is due to the fact that less bits are required for reconstructing the WZ frames than that of the key frames, when the SI for the WZ frames can be accurately estimated. Table 5.5 shows that the bitrate reduction ratio increases in the sequence order of Newspaper, Leavinglaptop, Outdoor and Ballroom. Similar trends are observed in terms of the PSNR reduction attained. The reason for this trend is that the receiver is more unlikely to be able to estimate the SI frame accurately from the received key frames, while our MSSM-PDWZ scheme is capable of effectively concealing the errors, which is an explicit benefit of our MSSM-Turbo decoder.

The comparison of the original frames to the corresponding estimated SI frames is displayed in Figure 5.25 for the *Newspaper*, *Leavinglaptop*, *Outdoor* and *Ballroom* sequences, respectively. Specifically, Figures 5.25a and 5.25b portray both the original and the corresponding estimated SI frames, respectively, where for each sequence a specific block is marked by a rectangular box and enlarged for the sake of clearer

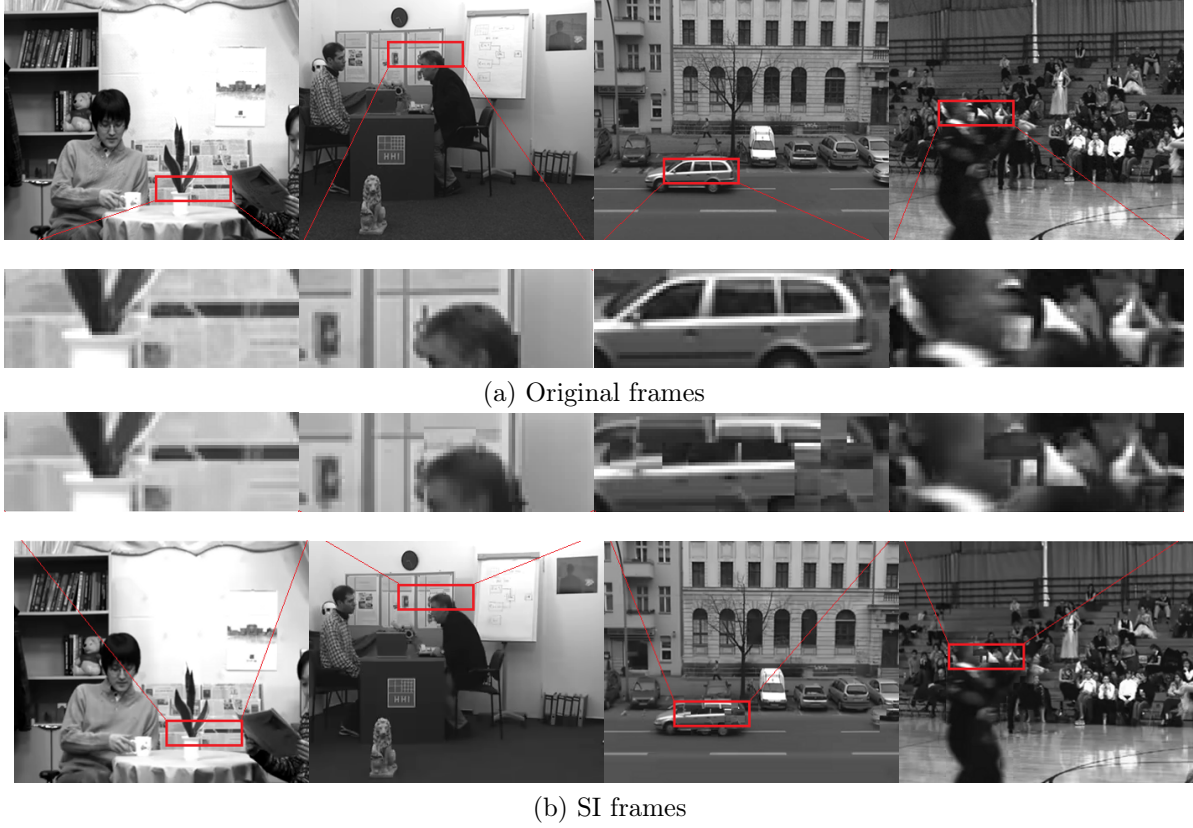


Figure 5.25: Comparison of original WZ frames and the related estimated SI frames at the receiver. From left to right are the *Newspaper*, *Leaving-laptop*, *Outdoor* and *Ballroom* sequences, respectively. Figures 5.25a and 5.25b list the original and the estimated SI frames, respectively. The schematic of Figure 5.10 configured with the parameters in Table 5.3 was used.

comparison. As shown in Figures 5.25a and 5.25b, for the *Newspaper* and *Leaving-laptop* sequences the estimated SI frames closely match the original frames, although minor distortions may be observed in the enlarged rectangular blocks. In contrast to the *Newspaper* and *Leaving-laptop* sequences, severe distortions are observed in the marked blocks for the *Outdoor* and *Ballroom* sequences. All these severely distorted blocks tend to appear at the position, where high-motion objects exist, such as a passing car in the *Outdoor* sequence. We note that these distorted blocks may substantially increase the bitrate required for reconstructing the multiview WZ frames. Overall, we observe from Figure 5.25 that the receiver fails to estimate the SI frames of the higher-motion sequences, namely of the *Outdoor* and the *Ballroom* sequences, as precisely as for the lower-motion sequences, namely for the *Newspaper* and the *Leaving-laptop* sequences. Based on Table 5.5 and Figure 5.25, we may conclude that our proposed MSSM technique is capable of reducing the bitrate more substantially for the higher-motion sequences, where the SI of key frames cannot be accurately estimated at the receiver.

	@34.2 dB	@37.4 dB	@40.5 dB	@43.3 dB
H.264/AVC-NoMotion (%)	45	44	26.9	17.2
IHVSM-PDWZ (%)	10	22	21.1	15.3
JPEG (%)	-80	-79	-59.5	-35.4
PDWZ (%)	10	13	7.7	4.7

(a) Hall

	@32.9 dB	@36 dB	@40 dB	@43.3 dB
H.264/AVC-NoMotion (%)	32.1	34.5	27.8	21
IHVSM-PDWZ (%)	4.3	10	17.5	19.2
JPEG (%)	-59.7	-61.6	-54.3	na
PDWZ (%)	0	0	3.6	-9.9

(b) Foreman

	@31 dB	@34 dB	@38 dB	@43 dB
H.264/AVC-NoMotion (%)	36.5	31	26.4	31.9
IHVSM-PDWZ (%)	36.5	25.5	23.1	34.1
JPEG (%)	-36.5	-29.2	-26.4	na
PDWZ (%)	18.3	12.8	8.8	7.7

(c) Coastguard

Table 5.6: Bitrate difference of different benchmarks compared to the H.264/AVC-Intra scheme at variable PSNRs. The schematic of Figure 5.7 configured with the parameters in Table 5.1 was used. Hall/Foreman/Coastguard sequence, GOP=2

5.5 Summary and Conclusions

In this chapter, we proposed a symbol-based model of iterative source decoding for the Wyner-Ziv video coding scheme of Figure 5.4, where the IHVSM scheme of Figure 4.1 was combined with a turbo decoder for the sake of reducing the required bitrate. Specifically, the IHVSM model of the intra-frame correlation within the WZ frames using two-dimensional first-order Markov processes was exploited, by performing iterative source and channel decoding with the aid of a turbo codec. Furthermore, the reduced-complexity first-order Markov model based decoding rules derived in

	@36 dB	@40 dB	@43 dB	@46 dB
H.264/AVC NoMotion (%)	42.2	46.7	44.5	41.2
JMVC-NoMotion (%)	42.2	43.3	39.5	37
MSSM-PDWZ (%)	48.3	45.3	44.5	41.5
PDWZ (%)	48.3	45.3	44.5	39.9

(a) Newspaper

	@36 dB	@40 dB	@43 dB	@46 dB
H.264/AVC NoMotion (%)	na	46	45.6	34.1
JMVC-NoMotion (%)	47.3	49.3	49	44.5
MSSM-PDWZ (%)	47.3	35.2	32.3	28.9
PDWZ (%)	47.3	31.2	26.7	21.5

(b) Leavinglaptop

	@36 dB	@40 dB	@43 dB	@46 dB
H.264/AVC NoMotion (%)	40.8	44.4	43.6	39.7
JMVC-NoMotion (%)	84.2	80.4	76.3	70
MSSM-PDWZ (%)	33.7	30.4	28.2	26.3
PDWZ (%)	29.5	22.8	15.4	7.4

(c) Outdoor

	@33 dB	@36 dB	@39 dB	@43 dB
H.264/AVC NoMotion (%)	39.8	58.3	60.7	53.9
JMVC-NoMotion (%)	63.8	61.5	59.1	51.1
MSSM-PDWZ (%)	11.3	0	-11.8	-17.2
PDWZ (%)	3.8	-11.9	-33.5	46.5

(d) Ballroom

Table 5.7: Bitrate difference of different benchmarks compared to the H.264/AVC-Intra scheme at variable PSNRs. The schematic of Figure 5.10 configured with the parameters in Table 5.3 was used. Newspaper/Leavinglaptop/Outdoor/Ballroom sequence, GOP=2

Section 4.2.3 were employed. The IHVSM scheme of Figure 4.1 was then successfully amalgamated with a PDWZ system of Figure 5.1 for the sake of reducing the bitrate. In our simulations the monoscopic Hall, Foreman and Coastguard sequences of Table 5.1 were invoked for evaluating our system's performance.

Our simulation results showed that we were able to reduce the bitrate by up to 25.1% compared to the PDWZ benchmarker of Figure 5.1. The rate-distortion results of Figures 5.6, 5.7 and 5.8 showed that the proposed IHVSM-PDWZ system substantially improves the achievable system performance and may facilitate a bitrate reduction in excess of 36.5% compared to the H.264/AVC-Intra scheme. More specifically, the achievable bitrate reductions are listed in Table 5.6, which were attained by the benchmarkers of Section 5.3 compared to the H.264/AVC-Intra coded scheme. Although our proposed scheme is capable of outperforming the PDWZ system of Figure 5.1, a substantial gap exists in the lower PSNR range compared to the H.264/AVC-NoMotion scheme. The IHVSM scheme of Figure 4.1 may be applied to the TDWZ [101] in the future for the sake of further reducing the required bitrate.

Furthermore, we proposed the Wyner-Ziv coded multi-view video system of Figure 5.10. Then we conceived the novel mesh-structured pixel correlation model of Figure 5.14 from the inter-view MVs and derived its decoding rules. Specifically, the proposed MSSM scheme links the correlated inter-view pixels of the multi-view WZ frames by creating the mesh-structured trellis of Figure 5.14. Iterative source and channel decoding was performed by exchanging extrinsic information between the MSSM decoder and the turbo codec for the sake of reducing the required bitrate of the multi-view WZ frames. Furthermore, by incorporating the MSSM scheme into the WZ video coding of multi-view video, we were able to substantially reduce the bitrate compared to that of the PDWZ benchmarker systems.

The rate-distortion results of Figures 5.17-5.24 showed that the proposed MSSM-PDWZ system substantially improves the achievable system performance and may facilitate a bitrate reduction in excess of 30% compared to the PDWZ scheme. More specifically, the resultant bitrate reductions are listed in Table 5.7, which are obtained by the benchmarking schemes of Section 5.4.3 compared to the H.264/AVC-Intra scheme. Similar to the IHVSM-PDWZ system of Figure 5.4, the MSSM scheme of Figure 5.15 may be applied to a TDWZ system in the future for the sake of further reducing the required bitrate.

Conclusions and Future Work

In this concluding chapter, a summary of the thesis and the main findings of our investigations will be presented. This will be followed by a range of ideas on future research.

6.1 Summary and Conclusions

This thesis investigated a number of techniques that may be utilized for video compression and communication. Specifically, the T-MDC, IL-FEC, IHVSM, M3DISC and MSSM schemes of Sections 2.2, 3.2, 4.2.1, 4.3.1 and 5.4 were proposed.

- Chapter 1 portrayed the background of our research in this thesis. Specifically, a brief history of video compression techniques was provided in Section 1.1. Then the techniques of UEP, ISCD and DVC invoked for video communications were reviewed in Section 1.2. Section 1.3 briefly introduced the principle of EXIT charts, which may be utilized for characterizing the family of turbo-like iterative decoders. The new research proposals of this thesis were motivated in Section 1.4, followed by the outline of this treatise. Finally, the novel contributions of this thesis were summarized in Section 1.6.
- Chapter 2 proposed the T-MDC scheme, which offers a flexible technique of creating multiple descriptions that may be beneficially combined with arbitrary video codecs. The creation of multiple descriptions was detailed in Section 2.2, followed by the video reconstruction techniques of Section 2.3. More specifically, the technique advocated is capable of representing the original video signal with the aid of an appropriately chosen number of correlated descriptions in the time-domain, while retaining the correlation among the video frames within each description. Each description may be encoded using arbitrary video compression tools into a bitstream. Furthermore, our proposed scheme is

also capable of splitting the video stream into multiple descriptions of unequal importance. T-MDC scheme advocated was also employed in the scenario of MDC for multiview video in Section 2.4. In Figure 2.10, the proposed HOCT-MDC scheme was shown to increasingly outperform the simulcast-SDC-H.264 arrangement upon increasing the bitrate. Figure 2.14 demonstrated that the BT-MDC scheme of Section 2.2.2 significantly outperforms both the HOCT-MDC and the simulcast-SDC-H.264 scheme.

- Chapter 3 proposed a novel inter-layer FEC coded video scheme. Specifically, Section 3.2 detailed the IL-FEC aided layered video coding systems advocated, where the information of the base layer was also incorporated into the systematic bits of the enhancement layers with the aid of an XOR operation. When the base layer can be successfully decoded in its own right, the systematic bits of the enhancement layers can be extracted by simply flipping the sign of the XOR-based parity check information received without introducing any degradation. This parity-check the check information is generated by performing inter-layer *XOR* operation on the base layer and the enhancement layers. However, when the base layer cannot be correctly decoded without the assistance of the enhancement layers, the proposed inter-layer FEC decoding philosophy relying on exchanging information between the base layer and the enhancement layers will be activated to assist in decoding the base layer. Then, in Section 3.3 we analyzed the system detailed in Section 3.2 using the powerful tools of EXIT charts, for explicitly demonstrating the benefits of our proposed IL-FEC arrangement using both RSC and turbo codecs. Then, in Section 3.4 we conceived the Opt-IL-FEC system for finding the optimized coding rates for coded bitstreams “on-the-fly” at the transmitter, which optimizes the IL-FEC coded system’s performance. Sections 3.5 and 3.6 presented our simulation results characterizing the IL-FEC systems using RSC, turbo and SECCC codes in a range of layered video communications scenarios. The performance of the Opt-IL-FEC codes was characterized in Section 3.7, where the transmission of SVC coded video over Rayleigh fading channels was considered. Moreover, our simulation results showed that our proposed Opt-IL-FEC techniques were capable of substantially improving the system performance at a modestly increased complexity.
- In Chapter 4, we investigated first-order Markov process modelling based source decoder aided video transmission. Firstly, the novel concept of our IHVSM decoder consisting of a horizontal and a vertical Markov modelling aided decoder was proposed in Section 4.2, which can be combined with chosen FEC codecs.

Iterative decoding may be performed between these two source decoders by exchanging extrinsic information. More specifically, reduced-complexity decoding rules were derived for the first-order Markov process. The proposed system was benchmarked in Section 4.2.4, where we considered the transmission of RSC-encoded uncompressed video through a Rayleigh channel. Section 4.3 proposed the first-order Markov process aided three-dimensional iterative source-channel decoding concept relying on an RSC codec for uncompressed video transmissions. Both the horizontal and vertical intra-frame correlations as well as the inter-frame correlations were exploited by relying on their first-order Markov modelling processes. To elaborate a little further, the proposed technique is capable of exploiting both the intra-frame and inter-frame correlations for iterative source-channel decoding. In these investigations, a single RSC codec was combined with three independent source decoders for forming three decoder-pairs for three-stage decoding, where the RSC codec was utilized for improving the source decoder's convergence behavior. The proposed M3DISC-RSC system was benchmarked in Section 4.3, where we considered the transmission of RSC encoded uncompressed video through a Rayleigh channel.

- In Chapter 5, we investigated ISCD aided distributed video coding techniques. Specifically, Section 5.3 proposed a symbol-based model of iterative source decoding technique conceived for Wyner-Ziv video coding, where the IHVSM was combined with a turbo decoder for the sake of reducing the required bitrate. In Section 5.3, the simulation results showed that the IHVSM aided DVC arrangement advocated was capable of reducing the bitrate by up to 21.5% compared to the PDWZ benchmarker system employed. Then, in Section 5.4, we proposed a Wyner-Ziv coded multi-view video system followed by the conception of a novel mesh-structured pixel correlation model of the inter-view MVs. We also derived the associated decoding rules. Furthermore, by incorporating the MSSM scheme of Section 5.4 into WZ video coding of multi-view video, we were able to substantially reduce the bitrate compared to that of the PDWZ benchmarker systems.

6.2 Design Guidelines

Based on the above-mentioned investigations, below we summarize our general design guidelines conceived for video communication schemes by carefully examining our proposals detailed in Chapters 1-5.

1. Multiple Description Coding Design

The MDC schemes are typically benchmarked against the classic characteristics

routinely interpreted as the PSNR versus bitrate metric. However, the correlation among the different MDC descriptions substantially affects the bitrate at a specific PSNR. This is due to the fact that MDC techniques artificially increase the redundancy inherent in the video signals, since they introduce multiple descriptions. *Hence the basic design guideline for MDC schemes is to impose as low redundancy as possible, when creating multiple descriptions.*

2. Layered Video Transmission Design

- (a) In layered video coding, we typically assume that each GOP contains F frames, where each frame is encoded into L layers. The F frames of each GOP tend to have unequal importance. Furthermore, where the L layers of each frame usually also have different importance. Hence, each GOP contains $(F \times L)$ different-significance layers. It is intuitive to aim for allocating more transmit/coding power to the more important layers. *Hence, the everliving design guideline for layered video transmission is to optimize the power allocation among the $(F \times L)$ layers for the sake of improving the attainable video quality in the face of transmission errors.*
- (b) In layered video streams, each layer may depend on other layers for decoding. Hence this characteristic may be interpreted as a type of redundancy within the bitstreams. Therefore it is beneficial to exploit this characteristic for the sake of achieving an improved video quality. *Therefore the corresponding design lesson for layered video transmission is to jointly encode multiple layers for the sake of exploiting their dependency at the receiver in order to improve the robustness against transmission errors.*

3. Sensitive Video Transmission Design

- (a) In low-complexity video communications scenarios, uncompressed video signals may be preferred for transmission, since the compressed video signals are typically sensitive to bit errors. Owing to the associated error propagation, the redundancy inherent within the uncompressed video signals should be modeled and exploited by a source decoder for the sake of achieving an improved video quality. *Hence a dominant design guideline for sensitive video transmission is to exploit the source redundancy without the aid of modeling the source signals using a source decoder.*
- (b) As mentioned above, the redundancy residing in the video signals should be efficiently exploited by a source decoder. ISCD has been widely investigated in video communications, since it is capable of improving the

reconstructed video quality. *Hence ISCD should be invoked where possible for jointly exploiting the redundancy within the source signals and the artificially imposed redundancy within the channel encoded signals.*

4. Distributed Video Coding Design

- (a) In distributed video coding, the key issue is that of exploiting the redundancy residing in the WZ frames for the sake of achieving a near-capacity rate-distortion performance. This redundancy may be exploited either at the transmitter or at the receiver. Completely removing the redundancy of the WZ frames at the transmitter may result in difficulty of estimating the SI of the WZ frames at the receiver. Hence it is necessary to exploit the redundancy at the receiver with the aid of modeling the redundancy using a source decoder. *The logical design guideline for distributed video coding is therefore to exploit the source redundancy by modeling the source signals using a source decoder.*
- (b) In distributed multiview video coding, the correlation among different views cannot be exploited at the transmitter. Fortunately, by exploiting the Slepian-Wolf theorem, we are able to exploit this correlation at the receiver for the sake of reducing the required bitrate, which is achieved with the aid of modeling the inter-view correlation using a source decoder. *The logical design guideline for distributed multiview video coding is hence to model and exploit the inter-view correlation using a source decoder.*
- (c) As mentioned above, the redundancy exhibiting itself in terms of correlation should be modeled by source decoders. Moreover, a FEC codec is invoked in the cutting-edge DVC systems. Hence ISCD may be employed for exploiting the joint action of source decoders and the FEC codec. *Therefore, intelligent ISCD should be employed for jointly exploiting the inherent redundancy within the multi-view signals and the artificial redundancy within the channel encoded signals.*

6.3 Suggestions for Future Work

A number of techniques were proposed for video compression and communication. However, further research may be conducted based upon these schemes to further improve the attainable performance.

6.3.1 Optimization of T-MDC

The family of rateless codecs, such as LT and Raptor codes, facilitate the transmission of information at adaptive rates under time-variant channel conditions. Hence

they should be further investigated in the context of near-instantaneously adaptive schemes. As an alternative techniques of combating packet loss events, by the T-MDC arrangement of Chapter 2, a number of parameters may be adaptively configured, which is expected to improve the attainable performance of the scheme quite substantially. Hence, further parameter optimization should be conducted in order to find the best parameter configuration set.

6.3.2 Joint Encoding of the BL and ELs

In Chapter 3, an IL-FEC scheme was proposed for the protection of layered video bitstreams, which achieved an improved video quality. However, in our current systems only the systematic bits of the enhancement layers were utilized for protection of the systematic bits of the base layer. However, all the data of the dependent layers, including the systematic bits and parity bits, became useless when the base layer was corrupted. Hence it is intuitive to make use of both the systematic bits and the parity bits of the dependent layers in our IL-FEC schemes. Moreover, the XOR operation was employed in our IL-FEC systems for jointly encoding the systematic bits of the BL and ELs. However, more advanced FEC codes may be employed for further improving the achievable performance. Similar solutions may also be used for both multiview and stereoscopic video coding.

6.3.3 ISCD Aided DVC

As detailed in Chapters 4 and 5, the ISCD principle was utilized in DVC and uncompressed video communication by exploiting the intra-frame and inter-frame correlation. However, the attainable performance was still far from the theoretical bound. Hence optimum algorithms should be conceived for exploiting the three-dimensional video signal to further improve the overall system performance. Specifically, in the DVC application considered, our system was parameterized using a GOP size of 2. In traditional high complexity video encoder scenarios, the bitrate can be reduced by increasing the GOP size, especially in the lower range of the GOP values. However, it is quite a challenge to improve the attainable performance especially when increasing the GOP size. This is due to the fact even the cutting-edge SI estimation techniques predict the WZ frames less accurately in these large GOP scenarios. Hence an increased bitrate is required for the WZ frames, when GOP size is increased. Based on our preliminary experimental results we anticipate that our proposed JSCC scheme may be employed for error-concealment at the receiver, while our ISCD technique may be employed in this scenario for the sake of reducing the bitrate required for the

WZ frames. Moreover, since the accuracy of the Markov transition table substantially affects the performance of our ISCD modeling algorithm, an optimum solution should be found for the sake of reducing the bitrate.

6.3.4 Joint Intra- and Inter-View ISCD

The MSSM scheme of Chapter 5 was invoked for the sake of reducing the bitrate of our distributed multiview video coding scheme. The intra-view correlation modeling of Section 5.3 should also be activated to assist in reducing the bitrate. The joint intra- and inter-view correlation modeling may further reduce the required bitrate. However, the joint activation of all the correlation modeling techniques conceived is expected to increase both the conceptional-and implementational- complexity of the system. Hence efficient algorithm should be conceived for activating them simultaneously, whilst improving a modest system complexity, despite activating both the intra- and inter-view ISCD techniques of Chapter 5.

6.3.5 Reduced-Complexity ISCD

In this thesis, we employed first-order Markov processes for modeling the redundancy within the video signals. However, the complexity of the Markov processes increases exponentially upon increasing their memory, which may hence prohibit their practical application. Hence reduced-complexity source modeling techniques should be conceived for practical ISCD applications.

Decoding of First-Order Markov Process

Let us initially follow the procedure of the classic Bahl-Cocke-Jelinek-Raviv (BCJR) [183] algorithm based MAP decoder for deriving the APP $p(x_i|y_0^t)$. The APP $p(x_i|y_0^t)$ of the m -bit pixel $x_i, x_i \in X_m$ conditioned on the soft Markov process y_0, \dots, y_t may be formulated as

$$p(x_i|y_0^t) = \frac{p(x_i \wedge y_0^t)}{p(y_0^t)}, \quad (\text{A.1})$$

where $p(x_i \wedge y_0^t)$ indicates the probability that the soft Markov process based sequence of y_0^t is received and the pixel value x_i is transmitted at time instant i . Furthermore, the joint probability of $p(x_i \wedge y_0^t)$ may be further formulated as follows

$$\begin{aligned} & p(x_i \wedge y_0^t) \\ &= p(x_i \wedge y_i \wedge y_0^{i-1} \wedge y_{i+1}^t) \\ &= p(y_{i+1}^t | x_i \wedge y_i \wedge y_0^{i-1}) \cdot p(y_i | y_0^{i-1} \wedge x_i) \cdot p(y_0^{i-1} \wedge x_i) \\ &= p(y_{i+1}^t | x_i) \cdot p(y_i | x_i) \cdot p(y_0^{i-1} \wedge x_i) \\ &= \beta_i(x_i) \cdot \chi_i(x_i) \cdot \alpha_i(x_i). \end{aligned} \quad (\text{A.2})$$

In Eq. (A.2), the probability functions α, β, χ are defined as follows

$$\begin{aligned} \alpha_i(x_i) &= p(y_0^{i-1} \wedge x_i) \\ \beta_i(x_i) &= p(y_{i+1}^t | x_i) \\ \chi_i(x_i) &= p(y_i | x_i). \end{aligned} \quad (\text{A.3})$$

The probability function α , β , χ in Eq. (A.3) are illustrated graphically in the trellis of Figure 4.4. Specifically, the function α indicates the probability that the soft pixels y_0^{i-1} are received and the pixel x_i is transmitted at time instant i , while the function β indicates the probability that the soft pixels y_{i+1}^t are received conditioned on the pixel x_i being transmitted at time instant i . In Eq. (A.3), the symbol-based channel information $\chi_i(x_i) = p(y_i|x_i)$ gives the a-priori probability of the pixel x_i , which may be calculated from the bit-based channel information $p[y_i(k)|x_i(k)]$ as follows

$$\begin{aligned}\chi_i(x_i) &= p(y_i|x_i) \\ &= \prod_{k=0}^{m-1} p[y_i(k)|x_i(k)] \\ &= C_{\chi_i} \cdot \exp \sum_{k=0}^{m-1} \frac{x_i(k)}{2} \cdot L[y_i(k)|x_i(k)],\end{aligned}\tag{A.4}$$

where C_{χ_i} is the normalization factor, which solely depends on the soft information y_i . Furthermore, similar to the forward recursion calculation of the BCJR algorithm, the function $\alpha_i(x_i)$ in Eq. (A.3) may be formulated as

$$\begin{aligned}\alpha_i(x_i) &= p(y_0^{i-1} \wedge x_i) \\ &= p(y_{i-1} \wedge y_0^{i-2} \wedge x_i) \\ &= \sum_{x_{i-1} \in X_m} p(y_{i-1} \wedge y_0^{i-2} \wedge x_i \wedge x_{i-1}) \\ &= \sum_{x_{i-1} \in X_m} p(y_{i-1} \wedge x_i | y_0^{i-2} \wedge x_{i-1}) \cdot p(y_0^{i-2} \wedge x_{i-1}) \\ &= \sum_{x_{i-1} \in X_m} p(y_{i-1} \wedge x_i | x_{i-1}) \cdot p(y_0^{i-2} \wedge x_{i-1}) \\ &= \sum_{x_{i-1} \in X_m} p(y_{i-1} | x_{i-1}) \cdot p(x_i | x_{i-1}) \cdot p(y_0^{i-2} \wedge x_{i-1}) \\ &= \sum_{x_{i-1} \in X_m} \chi_{i-1}(x_{i-1}) \cdot p(x_i | x_{i-1}) \cdot \alpha_{i-1}(x_{i-1}).\end{aligned}\tag{A.5}$$

Similarly, the backward recursion calculation of the function $\beta_i(x_i)$ in Eq. (A.3) may be formulated as follows

$$\begin{aligned}
\beta_i(x_i) &= p(y_{i+1}^t | x_i) \\
&= p(y_{i+2}^t \wedge y_{i+1} | x_i) \\
&= \sum_{x_{i+1} \in X_m} p(y_{i+2}^t \wedge y_{i+1} \wedge x_{i+1} | x_i) \\
&= \sum_{x_{i+1} \in X_m} p(y_{i+2}^t | y_{i+1} \wedge x_{i+1} \wedge x_i) \cdot p(y_{i+1} \wedge x_{i+1} | x_i) \\
&= \sum_{x_{i+1} \in X_m} p(y_{i+2}^t | x_{i+1}) \cdot p(y_{i+1} | x_{i+1}) \cdot p(x_{i+1} | x_i) \\
&= \sum_{x_{i+1} \in X_m} \beta_{i+1}(x_{i+1}) \cdot \chi_{i+1}(x_{i+1}) \cdot p(x_{i+1} | x_i).
\end{aligned} \tag{A.6}$$

Based on the above rules, the functions α , β , χ of Eq. (A.2) may be calculated. On the other hand, the bit-based APP $p[x_i(k)|y_0^t]$ may be determined from the symbol-based APP $p(x_i|y_0^t)$ as [87]

$$p[x_i(k) = u | y_0^t] = \sum_{\substack{x_i(k)=u \\ x_i \in X_m}} p(x_i | y_0^t), \tag{A.7}$$

where u represents the legitimate binary values 0 or 1 of $x_i(k)$. Then, the bit-based APP of the bit $x_i(k)$ conditioned on the soft Markov process y_0^t may be formulated as

$$\begin{aligned}
L[x_i(k) | y_0^t] &= \ln \frac{p[x_i(k) = 0 | y_0^t]}{p[x_i(k) = 1 | y_0^t]} \\
&= \ln \frac{\sum_{\substack{x_i(k)=0 \\ x_i \in X_m}} p(x_i | y_0^t)}{\sum_{\substack{x_i(k)=1 \\ x_i \in X_m}} p(x_i | y_0^t)} \\
&= \ln \frac{\sum_{\substack{x_i(k)=0 \\ x_i \in X_m}} p(x_i \wedge y_0^t)}{\sum_{\substack{x_i(k)=1 \\ x_i \in X_m}} p(x_i \wedge y_0^t)} \\
&= \ln \frac{\sum_{\substack{x_i(k)=0 \\ x_i \in X_m}} \beta_i(x_i) \cdot \chi_i(x_i) \cdot \alpha_i(x_i)}{\sum_{\substack{x_i(k)=1 \\ x_i \in X_m}} \beta_i(x_i) \cdot \chi_i(x_i) \cdot \alpha_i(x_i)}.
\end{aligned} \tag{A.8}$$

In contrast to all other bit patterns, the symbol-based *a-posteriori* probability of the first and last m -bit patterns, namely those at time instant 0 and t respectively, may be calculated as follows

$$\begin{aligned}
 p(x_0 \wedge y_0^t) &= p(x_0 \wedge y_0 \wedge y_1^t) \\
 &= p(y_1^t | x_0 \wedge y_0) \cdot p(y_0 | x_0) \cdot p(x_0) \\
 &= p(y_1^t | x_0) \cdot p(y_0 | x_0) \cdot p(x_0) \\
 &= \beta_0(x_0) \cdot \chi_0(x_0) \cdot p(x_0); \\
 p(x_t \wedge y_0^t) &= p(x_t \wedge y_t \wedge y_0^{t-1}) \\
 &= p(y_t | y_0^{t-1} \wedge x_t) \cdot p(y_0^{t-1} \wedge x_t) \\
 &= p(y_t | x_t) \cdot p(y_0^{t-1} \wedge x_t) \\
 &= \chi_t(x_t) \cdot \alpha_t(x_t).
 \end{aligned} \tag{A.9}$$

The forward and backward calculation of α and β , respectively, may be initialized as follows

$$\begin{aligned}
 \alpha_0(x_0) &= p(x_0) \\
 \beta_t(x_t) &= 1.
 \end{aligned} \tag{A.10}$$

Furthermore, based on the Jacobian logarithm [78] of

$$\begin{aligned}
 \ln(e^{x_1} + e^{x_2}) &= \max(x_1, x_2) + \ln(1 + e^{-|x_1 - x_2|}) = g(x_1, x_2) \\
 \ln \sum_{i=1}^I e^{x_i} &= g[x_I, g(x_{I-1}, \dots, g(x_2, x_1) \dots)],
 \end{aligned} \tag{A.11}$$

we arrive at the log-domain representation of χ , α , β , which may be formulated as follows

$$\begin{aligned}
\Gamma_i(x_i) &= \ln [\chi_i(x_i)] \\
&= \ln C_{\chi_i} + \sum_{k=0}^{m-1} \frac{x_i(k)}{2} \cdot L[y_i(k)|x_i(k)] \\
A_i(x_i) &= \ln [\alpha_i(x_i)] \\
&= \ln \sum_{x_{i-1} \in X_m} \exp [\Gamma_{i-1}(x_{i-1}) + \ln p(x_i|x_{i-1}) + A_{i-1}(x_{i-1})] \\
B_i(x_i) &= \ln [\beta_i(x_i)] \\
&= \ln \sum_{x_{i+1} \in X_m} \exp [B_{i+1}(x_{i+1}) + \Gamma_{i+1}(x_{i+1}) + \ln p(x_i|x_{i+1})].
\end{aligned} \tag{A.12}$$

Then the *a-posteriori* LLR $L[x_i(k)|y_0^t]$ may be calculated as

$$\begin{aligned}
L[x_i(k)|y_0^t] &= \ln \frac{p[x_i(k)=0|y_0^t]}{p[x_i(k)=1|y_0^t]} \\
&= \ln \frac{\sum_{\substack{x_i \in X_m \\ x_i(k)=0}} \exp [B_i(x_i) + \Gamma_i(x_i) + A_i(x_i)]}{\sum_{\substack{x_i \in X_m \\ x_i(k)=1}} \exp [B_i(x_i) + \Gamma_i(x_i) + A_i(x_i)]}.
\end{aligned} \tag{A.13}$$

Derivation of MSSM Decoder

Let us initially follow the procedure of the classic Bahl-Cocke-Jelinek-Raviv (BCJR) [183] algorithm based determination rule of the MAP decoder for deriving the APP $p(x_{i,j}|Y_{i,j,1}^I)$ of Eq. (5.2). The APP $p(x_{i,j}|Y_{i,j,1}^I)$ of the m -bit pattern $x_{i,j}, x_{i,j} \in X_m$ conditioned on the specific $Y_{i,j,1}^I$ values may be expressed as

$$p(x_{i,j}|Y_{i,j,1}^I) = \frac{p(x_{i,j} \wedge Y_{i,j,1}^I)}{p(Y_{i,j,1}^I)}. \quad (\text{B.1})$$

The joint probability $p(x_{i,j} \wedge Y_{i,j,1}^I)$ of the m -bit pattern $x_{i,j}$ in Eq. (B.1) and that of SI $Y_{i,j,1}^I$ may be further formulated as

$$\begin{aligned} p(x_{i,j} \wedge Y_{i,j,1}^I) &= p(x_{i,j} \wedge Y_{i,j,i} \wedge Y_{i,j,1}^{i-1} \wedge Y_{i,j,i+1}^I) \\ &= p(Y_{i,j,i+1}^I | x_{i,j} \wedge Y_{i,j,i} \wedge Y_{i,j,1}^{i-1}) \cdot p(Y_{i,j,i} | Y_{i,j,1}^{i-1} \wedge x_{i,j}) \cdot p(Y_{i,j,1}^{i-1} \wedge x_{i,j}) \\ &= p(Y_{i,j,i+1}^I | x_{i,j}) \cdot p(Y_{i,j,i} | x_{i,j}) \cdot p(Y_{i,j,1}^{i-1} \wedge x_{i,j}) \\ &= p(Y_{i,j,i+1}^I | x_{i,j}) \cdot p(y_{i,j} | x_{i,j}) \cdot p(Y_{i,j,1}^{i-1} \wedge x_{i,j}) \\ &= \beta(x_{i,j}) \cdot \chi(x_{i,j}) \cdot \alpha(x_{i,j}). \end{aligned} \quad (\text{B.2})$$

In Eq. (B.2), the symbol-based channel information $\chi(x_{i,j}) = p(y_{i,j}|x_{i,j})$ may be calculated from the bit-based channel information as

$$\chi(x_{i,j}) = C_{i,j} \cdot \exp \sum_{k=0}^{m-1} \frac{x_{i,j}(k)}{2} \cdot L[y_{i,j}(k)|x_{i,j}(k)], \quad (\text{B.3})$$

where $C_{i,j}$ is a normalization factor, which solely depends on $y_{i,j}$. Furthermore, similar to the forward recursion calculation in the BCJR algorithm, the component

$\alpha(x_{i,j})$ in Eq. (B.2) may be formulated as

$$\begin{aligned}
\alpha(x_{i,j}) &= p(Y_{i,j,1}^{i-1} \wedge x_{i,j}) \\
&= p(Y_{i,j,i-1} \wedge Y_{i,j,1}^{i-2} \wedge x_{i,j}) \\
&= \sum_{\substack{L_{i,j,i-1} \in \\ X_m^{L_{i,j,i-1}}}} p(Y_{i,j,i-1} \wedge Y_{i,j,1}^{i-2} \wedge x_{i,j} \wedge L_{i,j,i-1}) \\
&= \sum_{\substack{L_{i,j,i-1} \in \\ X_m^{L_{i,j,i-1}}}} p(Y_{i,j,i-1} \wedge x_{i,j} | L_{i,j,i-1}) \cdot p(Y_{i,j,1}^{i-2} \wedge L_{i,j,i-1}) \\
&= \sum_{\substack{L_{i,j,i-1} \in \\ X_m^{L_{i,j,i-1}}}} p(Y_{i,j,i-1} | L_{i,j,i-1}) \cdot p(x_{i,j} | L_{i,j,i-1}) \cdot p(Y_{i,j,1}^{i-2} \wedge L_{i,j,i-1}).
\end{aligned} \tag{B.4}$$

Note that, given the original pixel set $L_{i,j,i-1}$ the soft pixels within $Y_{i,j,i-1}$ are independent to each other, hence the item $p(Y_{i,j,i-1} | L_{i,j,i-1})$ in Eq. (B.4) may be expressed as

$$p(Y_{i,j,i-1} | L_{i,j,i-1}) = \prod_{\substack{x_{i-1,z} \in \\ L_{i,j,i-1}}} \chi(x_{i-1,z}). \tag{B.5}$$

Then the forward recursion calculation may be further formulated as follows

$$\begin{aligned}
\alpha(x_{i,j}) &= \sum_{\substack{L_{i,j,i-1} \in \\ X_m^{L_{i,j,i-1}}}} \left\{ \prod_{x_{i-1,z} \in L_{i,j,i-1}} [p(y_{i-1,z} | x_{i-1,z}) \cdot p(Y_{i-1,j,1}^{i-2} \wedge x_{i-1,z})] \cdot p(x_{i,j} | L_{i,j,i-1}) \right\} \\
&= \sum_{\substack{L_{i,j,i-1} \in \\ X_m^{L_{i,j,i-1}}}} \left\{ \prod_{x_{i-1,z} \in L_{i,j,i-1}} [\chi(x_{i-1,z}) \cdot \alpha(x_{i-1,z})] \cdot p(x_{i,j} | L_{i,j,i-1}) \right\}.
\end{aligned} \tag{B.6}$$

Furthermore, by assuming the pixels are independent to each other the item $p(x_{i,j} | L_{i,j,i-1})$ in Eq. (B.6) may be approximated as follows

$$p(x_{i,j} | L_{i,j,i-1}) = \frac{p(L_{i,j,i-1} | x_{i,j}) \cdot p(x_{i,j})}{p(L_{i,j,i-1})} \approx \frac{\prod_{\substack{x_{i-1,z} \in \\ L_{i,j,i-1}}} p(x_{i-1,z} | x_{i,j}) \cdot p(x_{i,j})}{\prod_{\substack{x_{i-1,z} \in \\ L_{i,j,i-1}}} p(x_{i-1,z})} = \frac{\prod_{\substack{x_{i-1,z} \in \\ L_{i,j,i-1}}} p(x_{i,j} | x_{i-1,z})}{p(x_{i,j})^{|L_{i,j,i-1}|-1}}. \tag{B.7}$$

Then by substituting Eq. (B.7) into the Eq. (B.6), $\alpha(x_{i,j})$ may be approximated as following

$$\alpha(x_{i,j}) \approx \prod_{\substack{x_{i-1,z} \in \\ L_{i,j,i-1}}} \left\{ \sum_{x_{i-1,z} \in X_m} \chi(x_{i-1,z}) \cdot \alpha(x_{i-1,z}) \cdot p(x_{i,j}|x_{i-1,z}) \right\} / p(x_{i,j})^{|L_{i,j,i-1}|-1}. \quad (\text{B.8})$$

Similar to $\alpha(x_{i,j})$, the backward recursion calculation $\beta(x_{i,j})$ in Eq. (B.2) can be formulated as follows

$$\begin{aligned} \beta(x_{i,j}) &= p(Y_{i,j,i+1}^I \wedge x_{i,j}) \\ &= p(Y_{i,j,i+2}^I \wedge Y_{i,j,i+1} | x_{i,j}) \\ &= \sum_{\substack{L_{i,j,i+1} \in \\ X_m^{|L_{i,j,i+1}|}}} p(Y_{i,j,i+2}^I \wedge Y_{i,j,i+1} \wedge L_{i,j,i+1} | x_{i,j}) \\ &= \sum_{\substack{L_{i,j,i+1} \in \\ X_m^{|L_{i,j,i+1}|}}} p(Y_{i,j,i+2}^I | Y_{i,j,i+1} \wedge L_{i,j,i+1} \wedge x_{i,j}) \cdot p(Y_{i,j,i+1} \wedge L_{i,j,i+1} | x_{i,j}) \\ &= \sum_{\substack{L_{i,j,i+1} \in \\ X_m^{|L_{i,j,i+1}|}}} p(Y_{i,j,i+2}^I | L_{i,j,i+1}) \cdot p(Y_{i,j,i+1} | L_{i,j,i+1}) \cdot p(L_{i,j,i+1} | x_{i,j}) \\ &= \sum_{\substack{L_{i,j,i+1} \in \\ X_m^{|L_{i,j,i+1}|}}} \left\{ \prod_{\substack{x_{i+1,z} \in \\ L_{i,j,i+1}}} [p(Y_{i+1,j,i+2}^I | x_{i+1,z}) \cdot p(y_{i+1,z} | x_{i+1,z}) \cdot p(x_{i+1,z} | x_{i,j})] \right\} \\ &= \sum_{\substack{L_{i,j,i+1} \in \\ X_m^{|L_{i,j,i+1}|}}} \left\{ \prod_{\substack{x_{i+1,z} \in \\ L_{i,j,i+1}}} [\beta(x_{i+1,z}) \cdot \chi(x_{i+1,z}) \cdot p(x_{i+1,z} | x_{i,j})] \right\} \\ &= \prod_{\substack{x_{i+1,z} \in \\ L_{i,j,i+1}}} \left\{ \sum_{\substack{x_{i+1,z} \in \\ \in X_m}} [\beta(x_{i+1,z}) \cdot \chi(x_{i+1,z}) \cdot p(x_{i+1,z} | x_{i,j})] \right\}. \end{aligned} \quad (\text{B.9})$$

Finally, the determination of the bit-based APP LLRs $L[x_{i,j}(k) | Y_{i,j,1}^I]$ may be formulated as

$$\begin{aligned}
L[x_{i,j}(k) | Y_{i,j,1}^I] &= \ln \frac{p[x_{i,j}(k) = 0 | Y_{i,j,1}^I]}{p[x_{i,j}(k) = 1 | Y_{i,j,1}^I]} \\
&= \ln \frac{\sum_{\substack{x_{i,j} \in X_m \\ x_{i,j}(k)=0}} p(x_{i,j} | Y_{i,j,1}^I)}{\sum_{\substack{x_{i,j} \in X_m \\ x_{i,j}(k)=1}} p(x_{i,j} | Y_{i,j,1}^I)} \\
&= \ln \frac{\sum_{\substack{x_{i,j} \in X_m \\ x_{i,j}(k)=0}} \beta(x_{i,j}) \cdot \chi(x_{i,j}) \cdot \alpha(x_{i,j})}{\sum_{\substack{x_i \in X_m \\ x_i(k)=1}} \beta(x_{i,j}) \cdot \chi(x_{i,j}) \cdot \alpha(x_{i,j})},
\end{aligned} \tag{B.10}$$

where the Jacobian logarithm [78] can be readily applied for deriving the log-domain representation of our algorithm.

Glossary

Symbols

2D Two-Dimensional.

3D Three-Dimensional.

A

AA Antennas Array.

ACK Acknowledgement.

APP A-Posteriori Probability.

ARQ Automatic Repeat reQuest.

ASP Advanced Simple Profile.

AVC Advanced Video Coding.

AVS Advanced Video Standard.

B

B-frame Bi-directional predicted frame.

BCJR Bahl-Cocke-Jelinek-Raviv.

BEC Binary Erasure Channel.

BER Bit Error Ratio.

BL Base Layer.

BP Belief Propagation.

BPSK Binary Phase-Shift Keying.

BS Base Station.

BT-MDC Binary-Tree based Multiple Description Coding.

C

CC Convolutional Codes.

CGS Coarse Grain Scalability.

CIF Common Intermediate Format.

CN Channel-induced Noise.

CND Check Node Decoder.

CPB Coded Block Pattern.

CRC Cyclic Redundancy Check.

D

DC Direct Current.

DC-MDVC Drift-Compensation based Multiple Description Video Codec.

DCT Discrete Cosine Transform.

DID the parameter dependency_id in scalable video stream.

DISCOVER DIStributed COding for Video sERvices.

DMVC Distributed Multiview Video Coding.

DP Data Partitioning.

DPCM Differential Pulse Code Modulation.

DPT Difference Probability Table.

DV Digital Video.

DVD Digital Versatile Disc.

E

EEP Equal Error Protection.

EL Enhancement Layer.

ERC Efficient encoder Rate Control.

EXIT EXtrinsic Information Transfer.

EZW Embedded Zerotree Wavelet.

F

FEC Forward Error Correction.

FOMM First-Order Markov Model.

FPS Frames Per Second.

G

GOP Group Of Pictures.

H

HD High Definition.

HEVC High-Efficiency Video Coding.

HOCT Height-One Complete Tree.

HQAM Hierarchical Quadrature Amplitude Modulation.

HSMD Horizontal Scanline Model Decoder.

I

I-frame Intra-coded frame.

IEC International Electrotechnical Commission.

IF-MDVC Independent Flow-based Multiple Description Video Codec.

IFPI Intra-Frame Pixel Indices.

IHVSM Iterative Horizontal-Vertical Scanline Model.

IL-FEC Inter-Layer Forward Error Correction.

ISCD Iterative Source-Channel Decoding.

ISDN Integrated Services Digital Networks.

ISMD Inter-frame Scanline Model Decoder.

ITU International Telecommunication Union.

ITU-T International Telecommunication Union Telecommunication standardization sector.

J

JMVM Joint Multi-view Video Model.

JPEG Joint Photographic Experts Group.

JSCC Joint Source-Channel Coding.

JSCD Joint Source-Channel Decoding.

JSVM Joint Scalable Video Model.

JVT Joint Video Team.

K

kbps KBits Per Second.

L

LA-FEC Layer-Aware Forward Error Correction.

LDPC Low-Density Parity-Check Accumulate.

LDPC Low-Density Parity-Check.

LLR Log-Likelihood Ratio.

LMS Legitimate Markov States.

LSSTC Layered Steered Space-time Codes.

LT Luby Transform.

LUT LookUp Table.

LW-EZEP Layer-Weighted Expected Zone of Error Propagation.

M

M3DISC first-order Markov process aided Three-Dimensional Iterative Source-Channel decoding.

MAC Media Access Control.

MAP Maximum A-Posteriori probability.

MB Macro Block.

MDC Multiple Description Coding.

MDD Multiple Description Decoder.

MDDVC Multiple Description Distributed Video Coder.

MDE Multiple Description Encoder.

MDVC Multiple Description Video Coding.

MGS Medium Grain Scalability.

MI Mutual Information.

MIMO Multiple Input Multiple Output.

MMSE Minimum Mean-Square Error.

MMSTT Markov Model's State Transition Table.

MPEG Moving Picture Experts Group.

MRF Markov Random Field.

MSB Most Significant Bit plane.

MSE Mean Square Error.

MSSM Mesh-Structured Source Model.

MV Motion Vector.

MVC Multiview Video Coding.

MVI Monoscopic View Index.

MVP MultiView Profile.

N

NALU Network Abstraction Layer Unit.

NCR Natural Code Rate.

NTSC National Television System Committee.

P

P-frame Predicted frame.

PAL Phase Alternating Line.

PC Personal Computer.

PER Packet Error Ratio.

PF Particle Filtering.

PHY Physical.

PLR Packet Loss Ratio.

PM Partitioning Mode.

PRE Parity Rate Estimator.

PSNR Peak Signal-to-Noise Ratio.

Q

QAM Quadrature Amplitude Modulation.

QCIF Quarter Common Intermediate Format.

QID the parameter quality_id in scalable video stream.

QoE Quality of Experience.

QoS Quality of Service.

QP Quantization Parameter.

QPSK Quadrature Phase-Shift Keying.

R

RBSP Raw Byte Sequence Payload.

RCPC Rate-Compatible Convolutional Code.

RDO Rate Distortion Optimization.

RSC Recursive Systematic Convolutional.

S

SBC Short Block Code.

SBSD SoftBit Source Decoding.

SD Standard Definitions.

SDC Single Description Coding.

SECCC Self-Concatenated Convolutional Code.

SI Side Information.

SLDPCA Sum Low-Density Parity-Check Accumulate.

SMPTE Society of Motion Picture and Television Engineers.

SNR Signal-to-Noise Ratio.

SP Simple Profile.

SPIHT Set Partitioning relying on Hierarchical Trees.

SVC Scalable Video Coding.

SVEF Scalable Video-streaming Evaluation Framework.

SW Slepian-Wolf.

T

T-MDC Tree-structured Multiple Description Coding.

TCP-IP Transmission Control Protocol-Internet Protocol.

TDWZ Transform Domain Wyner-Ziv video coding.

TID the parameter `temporal_id` in scalable video stream.

TV TeleVision.

U

UDE Unequal error protection Density Evolution.

UEP Unequal Error Protection.

UHD Ultra High Definition.

V

VCD Video Compact Disc.

VCEG Video Coding Experts Group.

VLC Variable Length Coding.

VND Variable Node Decoder.

VSMD Vertical Scanline Model Decoder.

W

WMV9 Windows Media Video version 9.

WPAN Wireless Personal Area Networks.

WVSN Wireless Video Sensor Networks.

WZ Wyner-Ziv.

X

XOR eXclusive OR operation.

Bibliography

- [1] L. Hanzo, P. Cherriman, and J. Streit, *Video Compression and Communications: From Basics to H.261, H.263, H.264, MPEG2, MPEG4 for DVB and HSDPA-Style Adaptive Turbo-Transceivers*. New York: John Wiley, 2007.
- [2] P. Cherriman and L. Hanzo, "Programmable H.263-based wireless video transceivers for interference-limited environments," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 8, pp. 275–286, June 1998.
- [3] P. Cherriman, T. Keller, and L. Hanzo, "Orthogonal frequency division multiplex transmission of H.263 encoded video over highly frequency-selective wireless networks," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, pp. 701–712, August 1999.
- [4] COST 211 and CCITT, *ITU-T Rec. H.120: Codecs for videoconferencing using primary digital group transmission*, November 1988.
- [5] ITU-T, *H.261: Video codec for audiovisual services at p x 384 kbit/s*, November 1988.
- [6] G. K. Wallace, "The JPEG still picture compression standard," *Communications of the ACM*, pp. 30–44, 1991.
- [7] ISO/IEC 11172, *Coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbit/s*, 1993.
- [8] ITU-R, *BT.470 : Conventional analogue television systems*, February 2005.
- [9] ITU-T, *H.262 : Information technology - Generic coding of moving pictures and associated audio information: Video*, February 2012.
- [10] IEC 61834-2, *Recording - Helical-scan digital video cassette recording system using 6,35 mm magnetic tape for consumer use*, August 1998.
- [11] ITU-T, *H.263 : Video coding for low bit rate communication*, January 2005.
- [12] Internet Video Magazine, *The History and Future of Real Networks*, May 2011.
- [13] "SMPTE VC-1 Receiving Industry-wide Support." Available: <http://www.microsoft.com/en-us/news/press/2006/apr06/04-24VC1PR.aspx>.

- [14] Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, *ITU-T Rec. H.264/ISO/IEC 14496-10 AVC: Advanced Video Coding for Generic Audiovisual Services*, March 2010.
- [15] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the H.264/AVC standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, pp. 1103–1120, September 2007.
- [16] Y. Chen, Y.-K. Wang, K. Ugur, M. M. Hannuksela, J. Lainema, and M. Gabbouj, "The emerging MVC standard for 3D video services," *EURASIP Journal on Advances in Signal Processing*, vol. 2009, pp. 1–13, January 2009.
- [17] AVS Workgroup of China, *Audio and Video Coding Standard*, January 2005.
- [18] ITU-T, *H.265 : High efficiency video coding*, April 2013.
- [19] W. Jiang and A. Ortega, "Multiple description coding via polyphase transform and selective quantization," *SPIE Visual Communications and Image Processing*, vol. 3653, pp. 998–1008, February 1999.
- [20] B. J. Kim, Z. Xiong, and W. A. Pearlman, "Low bit-rate scalable video coding with 3-D set partitioning in hierarchical trees (3-D SPIHT)," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 10, pp. 1374–1387, December 2000.
- [21] J. M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Transactions on Signal Processing*, vol. 41, pp. 3445–3462, December 1993.
- [22] N. Franchi, M. Fumagalli, R. Lancini, and S. Tubaro, "Multiple description video coding for scalable and robust transmission over IP," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, pp. 321–334, March 2005.
- [23] M. Biswas, M. R. Frater, and J. F. Arnold, "Multiple description wavelet video coding employing a new tree structure," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, pp. 1361–1368, October 2008.
- [24] H. Karim, A. Sali, S. Worrall, A. Sadka, and A. Kondoz, "Multiple description video coding for stereoscopic 3D," *IEEE Transactions on Consumer Electronics*, vol. 55, pp. 2048–2056, November 2009.
- [25] A. K. Ramasubramonian and J. W. Woods, "Multiple description coding and practical network coding for video multicast," *IEEE Signal Processing Letters*, vol. 17, no. 3, pp. 265–268, 2010.
- [26] S. Milani and G. Calvagno, "Multiple description distributed video coding using redundant slices and lossy syndromes," *IEEE Signal Processing Letters*, vol. 17, no. 1, pp. 51–54, 2010.
- [27] C. Lin, T. Tillo, Y. Zhao, and B. Jeon, "Multiple description coding for h.264/avc with redundancy allocation at macro block level," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21, no. 5, pp. 589–600, 2011.

- [28] N. Kamnoonwatana, D. Agrafiotis, and C. N. Canagarajah, "Flexible adaptive multiple description coding for video transmission," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 1, pp. 1–11, 2012.
- [29] W.-J. Tsai and H.-Y. You, "Multiple description video coding based on hierarchical B pictures using unequal redundancy," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 2, pp. 309–320, 2012.
- [30] V. K. Goyal, "Multiple description coding: Compression meets the network," *IEEE Signal Processing Magazine*, vol. 18, pp. 74–93, September 2001.
- [31] Y. Wang, A. R. Reibman, and S. N. Lin, "Multiple description coding for video delivery," *Proceedings of the IEEE*, vol. 93, no. 1, pp. 57–69, 2005.
- [32] P. Cherriman and L. Hanzo, "Error-rate based power-controlled multimode H.263-assisted video telephony," *IEEE Transactions on Vehicular Technology*, vol. 48, pp. 1726–1738, September 1999.
- [33] J. N. Laneman, D. N. C. Tse, and G. W. Wornell, "Cooperative diversity in wireless networks: Efficient protocols and outage behavior," *IEEE Transactions on Information Theory*, vol. 50, pp. 3062–3080, December 2004.
- [34] M. Lukacs, "Predictive coding of multi-viewpoint image sets," in *IEEE Acoustics, Speech, and Signal Processing*, vol. 11, (Tokyo, Japan), pp. 521–524, April 1986.
- [35] A. Vetro, T. Thomas, and G. J. Sullivan, "Overview of the stereo and multiview video coding extensions of the H.264/MPEG-4 AVC standard," *Proceedings of the IEEE*, vol. 99, pp. 626–642, March 2011.
- [36] H. Imaizumi and A. Luthra, *Three-Dimensional Television, Video and Display Technologies*, ch. MPEG-2 Multiview Profile, pp. 169–181. Berlin, Heidelberg, and New York: Springer Verlag, 2002.
- [37] T. Zhang and Y. Xu, "Unequal packet loss protection for layered video transmission," *IEEE Transactions on Broadcasting*, vol. 45, pp. 243–252, June 1999.
- [38] Y. Huo, M. El-Hajjar, and L. Hanzo, "Wireless video: An inter-layer error protection aided multi-layer approach," *IEEE Vehicular Technology Magazine*, submitted for publication.
- [39] A. Vetro, T. Wiegand, and G. J. Sullivan, "Overview of the stereo and multiview video coding extensions of the H.264/MPEG-4 AVC standard," *Proceedings of the IEEE*, vol. 99, pp. 626–642, April 2011.
- [40] Nasruminallah and L. Hanzo, "EXIT-chart optimized short block codes for iterative joint source and channel decoding in H.264 video telephony," *IEEE Transactions on Vehicular Technology*, vol. 58, pp. 4306–4315, October 2009.

- [41] Nasruminallah and L. Hanzo, "Near-capacity H.264 multimedia communications using iterative joint source-channel decoding," *IEEE Communications Surveys and Tutorials*, vol. 14, pp. 538–564, Second Quarter 2012.
- [42] S. Wenger, "H.264/AVC over IP," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, pp. 645–656, July 2003.
- [43] A. Detti, G. Bianchi, C. Pisa, F. S. Proto, P. Loreti, W. Kellerer, S. Thakolsri, and J. Widmer, "SVEF: an open-source experimental evaluation framework for H.264 scalable video streaming," in *IEEE Symposium on Computers and Communications, ISCC*, (Sousse, Tunisia), pp. 36–41, July 2009.
- [44] I. Boyarinov and G. Katsman, "Linear unequal error protection codes," *IEEE Transactions on Information Theory*, vol. 27, pp. 168–175, March 1981.
- [45] N. Görtz, "Joint source channel decoding using bit-reliability information and source statistics," in *International Symposium on Information Theory*, (Cambridge, MA, USA), p. 9, August 1998.
- [46] B. Masnick and J. Wolf, "On linear unequal error protection codes," *IEEE Transactions on Information Theory*, vol. 13, pp. 600–607, October 1967.
- [47] F. Marx and J. Farah, "A novel approach to achieve unequal error protection for video transmission over 3G wireless networks," *Signal Processing: Image Communication*, vol. 19, no. 4, pp. 313–323, 2004.
- [48] T. Brüggén and P. Vary, "Unequal error protection by modulation with unequal power allocation," *IEEE Communications Letters*, vol. 9, pp. 484–486, June 2005.
- [49] H. Wang, F. Zhai, Y. Eisenberg, and A. K. Katsaggelos, "Cost-distortion optimized unequal error protection for object-based video communications," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, pp. 1505–1516, December 2005.
- [50] V. Pavlushkov, R. Johannesson, and V. V. Zyablov, "Unequal error protection for convolutional codes," *IEEE Transactions on Information Theory*, vol. 52, pp. 700–708, February 2006.
- [51] J. Hagenauer, "Rate-compatible puncture convolutional codes (RCPC) and their application," *IEEE Transactions on Communications*, vol. 36, pp. 389–400, April 1988.
- [52] V. Kumar and O. Milenkovic, "On unequal error protection LDPC codes based on Plotkin-type constructions," *IEEE Transactions on Communications*, vol. 54, pp. 994–1005, June 2006.
- [53] R. Gallager, *Low-Density Parity-Check Codes*. Cambridge, MA, USA: MIT Press, 1963.

- [54] N. Rahnavard, H. Pishro-Nik, and F. Fekri, "Unequal error protection using partially regular LDPC codes," *IEEE Transactions on Communications*, vol. 55, pp. 387–391, March 2007.
- [55] M. Aydinlik and M. Salehi, "Turbo coded modulation for unequal error protection," *IEEE Transactions on Communications*, vol. 56, pp. 555–564, April 2008.
- [56] H. Ha and C. Yim, "Layer-weighted unequal error protection for scalable video coding extension of H.264/AVC," *IEEE Transactions on Consumer Electronics*, vol. 54, pp. 736–744, May 2008.
- [57] Y. C. Chang, S. W. Lee, and R. Komiya, "A fast forward error correction allocation algorithm for unequal error protection of video transmission over wireless channels," *IEEE Transactions on Consumer Electronics*, vol. 54, pp. 1066–1073, August 2008.
- [58] Y. C. Chang, S. W. Lee, and R. Komiya, "A low complexity hierarchical QAM symbol bits allocation algorithm for unequal error protection of wireless video transmission," *IEEE Transactions on Consumer Electronics*, vol. 55, pp. 1089–1097, August 2009.
- [59] E. Maani and A. K. Katsaggelos, "Unequal error protection for robust streaming of scalable video over packet lossy networks," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 20, pp. 407–416, March 2010.
- [60] S. Ahmad, R. Hamzaoui, and M. M. Al-Akaidi, "Unequal error protection using fountain codes with applications to video communication," *IEEE Transactions on Multimedia*, vol. 13, pp. 92–101, February 2011.
- [61] M. Luby, "LT codes," (Vancouver, BC, Canada), pp. 271–280, November 2002.
- [62] C. Hellge, D. Gomez-Barquero, T. Schierl, and T. Wiegand, "Layer-aware forward error correction for mobile broadcast of layered media," *IEEE Transactions on Multimedia*, vol. 13, pp. 551–562, June 2011.
- [63] K. M. Alajel, W. Xiang, and Y. Wang, "Unequal error protection scheme based hierarchical 16-QAM for 3-D video transmission," *IEEE Transactions on Consumer Electronics*, vol. 58, no. 3, pp. 731–738, 2012.
- [64] A. A. Khalek, C. Caramanis, and R. W. Heath, "A cross-layer design for perceptual optimization of H.264/SVC with unequal error protection," *IEEE Journal on Selected Areas in Communications*, vol. 30, no. 7, pp. 1157–1171, 2012.
- [65] J. Micallef, R. Farrugia, and C. Debono, "Correlation noise based unequal error protected rate-adaptive codes for distributed video coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. PP, no. 99, pp. 1–1, 2013.
- [66] B. Masnick and J. Wolf, "On linear unequal error protection codes," *IEEE Transactions on Information Theory*, vol. 13, pp. 600–607, October 1967.
- [67] N. Rahnavard and F. Fekri, "New results on unequal error protection using LDPC codes," *IEEE Communications Letters*, vol. 10, pp. 43–45, January 2006.

- [68] C. Gong, G. Yue, and X. Wang, "Message-wise unequal error protection based on low-density parity-check codes," *IEEE Transactions on Communications*, vol. 59, pp. 1019–1030, April 2011.
- [69] F. Yang, Q. Zhang, W. Zhu, and Y. Zhang, "End-to-end TCP-friendly streaming protocol and bit allocation for scalable video over wireless Internet," *IEEE Journal on Selected Areas in Communications*, vol. 22, pp. 777–790, May 2004.
- [70] S. X. Ng, J. Y. Chung, and L. Hanzo, "Turbo-detected unequal protection MPEG-4 wireless video telephony using multi-level coding, trellis coded modulation and space-time trellis coding," *IEE Proceedings Communications*, vol. 152, pp. 1116–1124, December 2005.
- [71] D. Sejdinović, D. Vukobratović, A. Doufexi, V. Šenk, and R. J. Piechocki, "Expanding window fountain codes for unequal error protection," *IEEE Transactions on Communications*, vol. 57, no. 9, pp. 2510–2516, 2009.
- [72] D. Vukobratović, V. Stanković, D. Sejdinović, L. Stanković, and Z. Xiong, "Scalable video multicast using expanding window fountain codes," *IEEE Transactions on Multimedia*, vol. 11, no. 6, pp. 1094–1104, 2009.
- [73] K. Nguyen, T. Nguyen, and S.-C. Cheung, "Video streaming with network coding," *Journal of Signal Processing Systems*, vol. 59, pp. 319–333, June 2010.
- [74] M. Halloush and H. Radha, "Network coding with multi-generation mixing: A generalized framework for practical network coding," *IEEE Transactions on Wireless Communications*, vol. 10, no. 2, pp. 466–473, 2011.
- [75] Y. Huo, M. El-Hajjar, and L. Hanzo, "Inter-layer FEC aided unequal error protection for multilayer video transmission in mobile TV," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 23, no. 9, pp. 1622–1634, 2013.
- [76] D. Wu, Y. T. Hou, and Y.-Q. Zhang, "Transporting real-time video over the Internet: challenges and approaches," *Proceedings of the IEEE*, vol. 88, no. 12, pp. 1855–1877, 2000.
- [77] T. Stockhammer, M. M. Hannuksela, and T. Wiegand, "H.264/AVC in wireless environments," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 657–673, 2003.
- [78] L. Hanzo, T. H. Liew, B. Yeap, and R. Y. S. Tee, *Turbo Coding, Turbo Equalisation and Space-Time Coding*. New York: John Wiley, 2011.
- [79] S. X. Ng, J. Y. Chung, P. Cherriman, and L. Hanzo, "Burst-by-Burst Adaptive Decision Feedback Equalised TCM, TTCM and BICM for H.263-Assisted Wireless Video Telephony," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, pp. 363–374, March 2006.

- [80] T. Stockhammer, A. Shokrollahi, M. Watson, M. Luby, and T. Gasiba, "Application Layer Forward Error Correction for Mobile Multimedia Broadcasting," in *Handbook of Mobile Broadcasting: DVB-H, DMB, ISDB-T and Media FLO* (B. Furhet and S. Ahson, eds.), pp. 239–280, Boca Raton, FL: CRC Press, 2008.
- [81] C. Hellge, T. Schierl, and T. Wiegand, "Multidimensional layered forward error correction using rateless codes," in *IEEE International Conference on Communications*, (Beijing, China), pp. 480–484, May 2008.
- [82] C. E. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, vol. 27, pp. 379–423 and 623–656, June and October 1948.
- [83] N. Görtz, "On the iterative approximation of optimal joint source-channel decoding," *IEEE Journal on Selected Areas in Communications*, vol. 19, pp. 1662–1670, September 2001.
- [84] T. Fingscheidt and P. Vary, "Softbit speech decoding: A new approach to error concealment," *IEEE Transaction on Speech and Audio Processing*, vol. 9, pp. 240–251, March 2001.
- [85] J. Kliewer, N. Görtz, and A. Mertins, "On iterative source-channel image decoding with Markov random field source models," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 4, (Montreal, Canada), pp. iv–661–iv–664, May 2004.
- [86] J. Kliewer, N. Görtz, and A. Mertins, "Iterative source-channel decoding with Markov random field source models," *IEEE Transactions on Signal Processing*, vol. 54, pp. 3688–3701, October 2006.
- [87] M. Adrat and P. Vary, "Iterative source-channel decoding: Improved system design using EXIT charts," *EURASIP Journal on Applied Signal Processing*, vol. 2005, pp. 928–941, May 2005.
- [88] J. Kliewer and R. Thobaben, "Iterative joint source-channel decoding of variable-length codes using residual source redundancy," *IEEE Transactions on Wireless Communications*, vol. 4, pp. 919–929, May 2005.
- [89] L. Hanzo, R. G. Maunder, J. Wang, and L.-L. Yang, *Near-Capacity Variable-Length Coding: Regular and Exit-Chart Aided Irregular Designs*. John Wiley & Sons Ltd, 2010.
- [90] Y. Zhang, C. Zhu, and K.-H. Yap, "A joint source-channel video coding scheme based on distributed source coding," *IEEE Transactions on Multimedia*, vol. 10, pp. 1648–1656, December 2008.
- [91] D. Slepian and J. Wolf, "Noiseless coding of correlated information sources," *IEEE Transactions on Information Theory*, vol. 19, pp. 471–480, July 1973.

- [92] Nasruminallah and L. Hanzo, "Short block codes for guaranteed convergence in soft-bit assisted iterative joint source and channel decoding," *Electronics Letters*, vol. 44, pp. 1315–1316, October 2008.
- [93] M. Fresia, F. Pérez-Cruz, H. V. Poor, and S. Verdú, "Joint source and channel coding," *IEEE Signal Processing Magazine*, vol. 27, pp. 104–113, November 2010.
- [94] W. Ji, Z. Li, and Y. Chen, "Joint source-channel coding and optimization for layered video broadcasting to heterogeneous devices," *IEEE Transactions on Multimedia*, vol. 14, no. 2, pp. 443–455, 2012.
- [95] K. Sayood and J. Borkenhagen, "Use of residual redundancy in the design of joint source/channel coders," *IEEE Transactions on Communications*, vol. 39, pp. 838–846, June 1991.
- [96] C. S. Lee, T. Keller, and L. Hanzo, "OFDM-based turbo-coded hierarchical and non-hierarchical terrestrial mobile digital video broadcasting," *IEEE Transactions on Broadcasting*, vol. 46, pp. 1–22, March 2000.
- [97] C. S. Lee, S. Vlahoyiannatos, and L. Hanzo, "Satellite based turbo-coded, blind-equalised 4-QAM and 16-QAM digital video broadcasting," *IEEE Transactions on Broadcasting*, vol. 46, pp. 23–34, March 2000.
- [98] M. Adrat, R. Vary, and J. Spittka, "Iterative source-channel decoder using extrinsic information from softbit-source decoding," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 4, (Salt Lake City, Utah, USA), pp. 2653–2656, May 2001.
- [99] A. Wyner and J. Ziv, "The rate-distortion function for source coding with side information at the decoder," *IEEE Transactions on Information Theory*, vol. 22, pp. 1–10, January 1976.
- [100] C. Brites and F. Pereira, "Correlation noise modeling for efficient pixel and transform domain Wyner-Ziv video coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, pp. 1177–1190, September 2008.
- [101] X. Artigas, J. Ascenso, M. Dalai, S. Klomp, D. Kubasov, and M. Ouaret, "The DISCOVER codec: Architecture, techniques and evaluation," in *Picture Coding Symposium*, vol. 6, (Lisbon, Portugal), pp. 1–4, November 2007.
- [102] J. Streit and L. Hanzo, "Dual-mode vector-quantised low-rate cordless videophone systems for indoors and outdoors applications," *IEEE Transactions on Vehicular Technology*, vol. 46, pp. 340–357, May 1997.
- [103] B. Girod, A. Aaron, S. Rane, and D. Rebollo-Monedero, "Distributed video coding," *Proceedings of the IEEE*, vol. 93, pp. 71–83, January 2005.
- [104] D. Varodayan, A. Aaron, and B. Girod, "Rate-adaptive codes for distributed source coding," *EURASIP Signal Processing Journal, Special Section on Distributed Source Coding*, vol. 86, pp. 3123–3130, November 2006.

- [105] A. Aaron, R. Zhang, and B. Girod, "Wyner-Ziv coding of motion video," in *Asilomar Conference on Signals, Systems and Computers*, vol. 1, (Pacific Grove, CA, USA), pp. 240–244, November 2002.
- [106] A. Aaron, E. Setton, and B. Girod, "Towards practical Wyner-Ziv coding of video," in *IEEE International Conference on Image Processing, ICIP*, vol. 3, (Barcelona, Catalonia, Spain), pp. 869–872, September 2003.
- [107] A. Aaron, S. Rane, R. Zhang, and B. Girod, "Wyner-Ziv coding for video: applications to compression and error resilience," in *IEEE Data Compression Conference, DCC*, (Snowbird, Utah, USA), pp. 93–102, March 2003.
- [108] N. Gehrig and P. L. Dragotti, "DIFFERENT - distributed and fully flexible image encoders for camera sensor networks," in *IEEE International Conference on Image Processing, ICIP*, vol. 2, (Genoa, Italy), pp. II – 690–3, September 2005.
- [109] W. Weerakkody, W. A. C. Fernando, and A. B. B. Adikari, "Unidirectional distributed video coding for low cost video encoding," *IEEE Transactions on Consumer Electronics*, vol. 53, pp. 788–795, May 2007.
- [110] D. Varodayan, D. Chen, M. Flierl, and B. Girod, "Wyner-Ziv coding of video with unsupervised motion vector learning," *Signal Processing: Image Communication*, vol. 23, pp. 369–378, June 2008.
- [111] X. Guo, Y. Lu, F. Wu, D. Zhao, and W. Gao, "Wyner-Ziv-based multiview video coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, pp. 713–724, June 2008.
- [112] C. Yeo and K. Ramchandran, "Robust distributed multiview video compression for wireless camera networks," *IEEE Transactions on Image Processing*, vol. 19, pp. 995–1008, April 2010.
- [113] R. Puri, A. Majumdar, and K. Ramchandran, "PRISM: A video coding paradigm with motion estimation at the decoder," *IEEE Transactions on Image Processing*, vol. 16, pp. 2436–2448, October 2007.
- [114] C. Brites and F. Pereira, "An efficient encoder rate control solution for transform domain Wyner-Ziv video coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21, pp. 1278–1292, September 2011.
- [115] Y. Zhang, H. Xiong, Z. He, S. Yu, and C. Chen, "Reconstruction for distributed video coding: A context-adaptive Markov random field approach," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21, pp. 1100–1114, August 2011.
- [116] S. Wang, L. Cui, L. Stankovic, V. Stankovic, and S. Cheng, "Adaptive correlation estimation with particle filtering for distributed video coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 5, pp. 649–658, 2012.

- [117] J. Skorupa, J. Slowack, S. Mys, N. Deligiannis, J. De Cock, P. Lambert, C. Grecos, A. Munteanu, and R. Van De Walle, "Efficient low-delay distributed video coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 4, pp. 530–544, 2012.
- [118] A. Abou-Elailah, F. Dufaux, J. Farah, M. Cagnazzo, and B. Pesquet-Popescu, "Fusion of global and local motion estimation for distributed video coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 23, no. 1, pp. 158–172, 2013.
- [119] P. Cherriman, E. L. Kuan, and L. Hanzo, "Burst-by-Burst Adaptive Joint-detection CDMA H.263 Based Video Telephony," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, pp. 342–348, May 2002.
- [120] Z. Xiong, A. D. Liveris, and S. Cheng, "Distributed source coding for sensor networks," *IEEE Signal Processing Magazine*, vol. 21, pp. 80–94, September 2004.
- [121] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo codes," in *IEEE International Conference on Communications*, (Geneva, Switzerland), pp. 1064–1070, May 1993.
- [122] M. Flierl and B. Girod, "Multiview video compression," *IEEE Signal Processing Magazine*, vol. 24, pp. 66–76, November 2007.
- [123] Y. Chen, M. M. Hannuksela, L. Zhu, A. Hallapuro, M. Gabbouj, and H. Li, "Coding techniques in multiview video coding and joint multiview video model," in *Picture Coding Symposium, PCS*, (Chicago, Illinois, USA), pp. 1–4, May 2009.
- [124] J. Zhang, M. M. Hannuksela, and H. Li, "Joint multiview video plus depth coding," in *IEEE International Conference on Image Processing, ICIP*, (Hong Kong, China), pp. 2865–2868, September 2010.
- [125] P. Merkle, A. Smolic, K. Muller, and T. Wiegand, "Comparative study of MVC prediction structures," in *Joint Video Team of ISO/IEC MPEG & ITU-T VCEG*, pp. Doc. JVT–V13, January 2007.
- [126] "Joint multiview video model (JMVM) 8.0," *JVT-AA207*, Geneva, Switzerland, April 2008.
- [127] I. F. Akyildiz, T. Melodia, and K. R. Chowdhury, "A survey on wireless multimedia sensor networks," *Computer Network*, vol. 51, pp. 921–960, March 2007.
- [128] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson, "Wireless sensor networks for habitat monitoring," in *ACM international workshop on Wireless sensor networks and applications*, WSNA, (New York, NY, USA), pp. 88–97, ACM, 2002.
- [129] C. Yeo, J. Wang, and K. Ramchandran, "View synthesis for robust distributed video compression in wireless camera networks," in *IEEE International Conference on Image Processing, ICIP*, vol. 3, (San Antonio, Texas), pp. III–21 – III–24, September 16–October 19 2007.

- [130] A. Said and W. A. Pearlman, "A new, fast, and efficient image codec based on set partitioning in hierarchical trees," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, pp. 243–250, June 1996.
- [131] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *International Joint Conference on Artificial intelligence - Volume 2*, IJCAI'81, (San Francisco, CA, USA), pp. 674–679, Morgan Kaufmann Publishers Inc., 1981.
- [132] S. ten Brink, "Convergence of iterative decoding," *Electronics Letters*, vol. 35, pp. 806–808, May 1999.
- [133] S. ten Brink, "Convergence behavior of iteratively decoded parallel concatenated codes," *IEEE Transactions on Communications*, vol. 49, pp. 1727–1737, October 2001.
- [134] S. ten Brink, G. Kramer, and A. Ashikhmin, "Design of low-density parity-check codes for modulation and detection," *IEEE Transactions on Communications*, vol. 52, pp. 670–678, April 2004.
- [135] S. X. Ng, J. Y. Chung, and L. Hanzo, "Turbo-detected unequal protection MPEG-4 wireless video telephony using multi-level coding, trellis coded modulation and space-time trellis coding," *IEE Proceedings - Communications*, vol. 152, pp. 1116–1124, December 2005.
- [136] Y. Huo and L. Hanzo, "Tree-structured multiple description coding for multiview mobile TV and camera-phone networks," in *IEEE Vehicular Technology Conference (VTC Fall)*, (San Francisco, USA), pp. 1–5, September 2011.
- [137] M. El-Hajjar and L. Hanzo, "Layered steered space-time codes and their capacity," *Electronics Letters*, vol. 43, pp. 680–682, June 2007.
- [138] L. Hanzo, O. Alamri, M. El-Hajjar, and N. Wu, *Near-Capacity Multi-Functional MIMO Systems: Sphere-Packing, Iterative Detection and Cooperation*. John Wiley & Sons, IEEE press, 2009.
- [139] Y. Huo, X. Zuo, R. G. Maunder, and L. Hanzo, "Inter-layer FEC decoded multi-layer video streaming," in *IEEE Global Telecommunications Conference (GLOBECOM)*, (Anaheim, CA, USA), pp. 2113–2118, December 2012.
- [140] Y. Huo, C. Zhu, and L. Hanzo, "Spatio-temporal iterative source-channel decoding aided video transmission," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 4, pp. 1597–1609, 2013.
- [141] Y. Huo, M. El-Hajjar, M. F. U. Butt, and L. Hanzo, "Inter-layer-decoding aided self-concatenated coded scalable video transmission," in *IEEE Wireless Communications and Networking Conference*, (Shanghai, China), pp. 4647–4652, April 2013.
- [142] J. Streit and L. Hanzo, "Quad-tree based parametric wireless videophone systems," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, pp. 225–237, April 1996.

- [143] P. Cherriman, T. Keller, and L. Hanzo, "Subband-adaptive turbo-coded OFDM-based interactive video telephony," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, pp. 829–839, October 2002.
- [144] T. Kunert, *User-Centered Interaction Design Patterns for Interactive Digital Television Applications*. Dordrecht, Heidelberg, London and New York: Springer Verlag, 2009.
- [145] W. Matusik and H. Pfister, "3D TV: A scalable system for real-time acquisition, transmission, and autostereoscopic display of dynamic scenes," *ACM Transactions on Graphics (TOG)*, vol. 23, pp. 811–821, August 2004.
- [146] L. Hanzo, J. Blogh, and S. Ni, *3G, HSPA and FDD versus TDD Networking: Smart Antennas and Adaptive Modulation*. New York: John Wiley, 2008.
- [147] F. Verdicchio, A. Munteanu, A. I. Gavrilescu, J. Cornelis, and P. Schelkens, "Embedded multiple description coding of video," *IEEE Transactions on Image Processing*, vol. 15, pp. 3114–3130, October 2006.
- [148] O. Crave, B. P. Popescu, and C. Guillemot, "Robust video coding based on multiple description scalar quantization with side information," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 20, pp. 769–779, June 2010.
- [149] P. W. Wolniansky, G. J. Foschini, G. D. Golden, and R. A. Valenzuela, "V-BLAST: an architecture for realizing very high data rates over the rich-scattering wireless channel," in *International Symposium on Signals, Systems, and Electronics, ISSSE*, (Pisa, Italy), pp. 295–300, September 1998.
- [150] V. Tarokh, H. Jafarkhani, and A. R. Calderbank, "Space-time block codes from orthogonal designs," *IEEE Transactions on Information Theory*, vol. 45, pp. 1456–1467, July 1999.
- [151] J. S. Blogh and L. Hanzo, *Third-Generation Systems and Intelligent Wireless Networking: Smart Antennas and Adaptive Modulation*. John Wiley and IEEE Press, 2002.
- [152] J. Streit and L. Hanzo, "Comparative study of programmable-rate videophone codecs for existing and future multimode wireless systems," *European Transactions on Telecommunications*, vol. 8, pp. 551–572, June 1997.
- [153] Y. Huo, M. El-Hajjar, R. G. Maunder, and L. Hanzo, "Layered wireless video relying on minimum-distortion inter-layer FEC coding," *IEEE Transactions on Multimedia*, early access, 2014.
- [154] J. Hagenauer, E. Offer, and L. Papke, "Iterative decoding of binary block and convolutional codes," *IEEE Transactions on Information Theory*, vol. 42, pp. 429–445, March 1996.

- [155] J. Chen, A. Dholakia, E. Eleftheriou, M. P. C. Fossorier, and X.-Y. Hu, "Reduced-complexity decoding of LDPC codes," *IEEE Transactions on Communications*, vol. 53, pp. 1288–1299, August 2005.
- [156] R. G. Maunder and L. Hanzo, "Extrinsic information transfer analysis and design of block-based intermediate codes," *IEEE Transactions on Vehicular Technology*, vol. 60, pp. 762–770, March 2011.
- [157] P. Cherriman, C. Wong, and L. Hanzo, "Turbo- and BCH-coded wide-band burst-by-burst adaptive H.263-assisted wireless video telephony," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 10, pp. 1355–1363, December 2000.
- [158] R. Otnes and M. Tüchler, "EXIT chart analysis applied to adaptive turbo equalization," in *Nordic Signal Processing Symposium*, (Hurtigruten, Norway), pp. 1–6, October 2002.
- [159] Y. Huo, M. El-Hajjar, and L. Hanzo, "Inter-layer turbo coded unequal error protection for multi-layer video transmission," in *IEEE Vehicular Technology Conference*, (Las Vegas, USA), pp. 1–5, September 2013.
- [160] R. Khalili and K. Salamatian, "A new analytic approach to evaluation of packet error rate in wireless networks," in *Communication Networks and Services Research Conference*, (Halifax, Nova Scotia, Canada), pp. 333–338, May 2005.
- [161] X. K. Yang, C. Zhu, Z. G. Li, X. Lin, G. N. Feng, S. Wu, and N. Ling, "Unequal loss protection for robust transmission of motion compensated video over the Internet," *Signal Processing: Image Communication*, vol. 18, no. 3, pp. 157–167, 2003.
- [162] M. Clerc and J. Kennedy, "The particle swarm - explosion, stability, and convergence in a multidimensional complex space," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 1, pp. 58–73, 2002.
- [163] D. Schonfeld and N. Bouaynaya, "A new method for multidimensional optimization and its application in image and video processing," *IEEE Signal Processing Letters*, vol. 13, no. 8, pp. 485–488, 2006.
- [164] Z.-H. Zhan, J. Zhang, Y. Li, and H. S.-H. Chung, "Adaptive particle swarm optimization," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 39, no. 6, pp. 1362–1381, 2009.
- [165] M. F. U. Butt, S. X. Ng, and L. Hanzo, "Self-concatenated code design and its application in power-efficient cooperative communications," *IEEE Communications Surveys Tutorials*, vol. 14, pp. 858–883, Third Quarter 2012.
- [166] R. Stedman, H. Gharavi, L. Hanzo, and R. Steele, "Transmission of subband-coded images via mobile channels," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 3, pp. 15–27, February 1993.

- [167] L. Hanzo and J. Streit, "Adaptive low-rate wireless videophone systems," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 5, pp. 305–319, August 1995.
- [168] H. Singh, J. Oh, C. Y. Kweon, X. Qin, H. Shao, and C. Ngo, "A 60 GHz wireless network for enabling uncompressed video communication," *IEEE Communications Magazine*, vol. 46, pp. 71–78, December 2008.
- [169] K. H. Liu, X. Ling, X. S. Shen, and J. W. Mark, "Performance analysis of prioritized MAC in UWB WPAN with bursty multimedia traffic," *IEEE Transactions on Vehicular Technology*, vol. 57, pp. 2462–2473, July 2008.
- [170] L. Hanzo, P. Cherriman, and E. L. Kuan, "Interactive cellular and cordless video telephony: State-of-the-art, system design principles and expected performance," *Proceedings of IEEE*, vol. 88, pp. 1388–1416, September 2000.
- [171] J. M. Gilbert, C. H. Doan, S. Emami, and C. B. Shung, "A 4-Gbps uncompressed wireless HD A/V transceiver chipset," *IEEE Micro*, vol. 28, pp. 56–64, March-April 2008.
- [172] A. Hutanu, R. Paruchuri, D. Eiland, M. Liska, P. Holub, S. R. Thorpe, and Y. Xin, "Uncompressed HD video for collaborative teaching - an experiment," in *International Conference on Collaborative Computing: Networking, Applications and Worksharing, CollaborateCom*, (New York, USA), pp. 253–261, November 2007.
- [173] S. T. Wei, C. W. Tien, B. D. Liu, and J. F. Yang, "Adaptive truncation algorithm for Hadamard-transformed H.264/AVC lossless video coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21, pp. 538–549, May 2011.
- [174] R. Fisher, "60 GHz WPAN standardization within IEEE 802.15.3c," in *International Symposium on Signals, Systems and Electronics, ISSSE*, (Montréal, Québec, Canada), pp. 103–105, July 30-August 2 2007.
- [175] C. Park and T. S. Rappaport, "Short-range wireless communications for next-generation networks: UWB, 60 GHz millimeter-wave WPAN, and ZigBee," *IEEE Wireless Communications*, vol. 14, pp. 70–78, August 2007.
- [176] <http://www.wirelessHD.org>, *WirelessHD Specification Overview*, October 2007.
- [177] D. Pepe and D. Zito, "60-GHz transceivers for wireless HD uncompressed video communication in nano-era CMOS technology," in *IEEE Mediterranean Electrotechnical Conference (MELECON)*, (Valletta, Malta), pp. 1237–1240, April 2010.
- [178] H. Singh, X. Qin, H. Shao, C. Ngo, C. Y. Kweon, and S. S. Kim, "Support of uncompressed video streaming over 60GHz wireless networks," in *IEEE Consumer Communications and Networking Conference, CCNC*, (Las Vegas, USA), pp. 243–248, January 2008.

- [179] H. Singh, H. Niu, X. Qin, H. Shao, C. Y. Kweon, G. Fan, S. S. Kim, and C. Ngo, "Supporting uncompressed HD video streaming without retransmissions over 60GHz wireless networks," in *IEEE Wireless Communications and Networking Conference, WCNC*, (Las Vegas, USA), pp. 1939–1944, March 31–April 3 2008.
- [180] H. Shao, C. Ngo, H. Singh, S. Qin, C. Y. Kweon, G. Fan, and S. S. Kim, "Adaptive multi-beam transmission of uncompressed video over 60GHz wireless systems," in *Future Generation Communication and Networking, FGCN*, vol. 1, (Ramada Plaza Jeju, Jeju-Island, Korea), pp. 430–435, December 2007.
- [181] S. E. Hong and W. Y. Lee, "Flexible unequal error protection scheme for uncompressed video transmission over 60GHz multi-Gigabit wireless system," in *International Conference on Computer Communications and Networks (ICCCN)*, (Mavi, Hawaii), pp. 1–6, July 31–August 4 2011.
- [182] P. Cherriman, E. L. Kuan, and L. Hanzo, "The Performance of H.263-based Video Telephony over Turbo-Equalised GSM/GPRS," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, pp. 909–915, October 2002.
- [183] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimising symbol error rate," *IEEE Transactions on Information Theory*, vol. 20, pp. 284–287, March 1974.
- [184] Y. Huo, T. Wang, R. G. Maunder, and L. Hanzo, "Iterative source and channel decoding relying on correlation modelling for wireless video transmission," *IET Communications*, vol. 7, pp. 1465–1475, September 2013.
- [185] C. Zhu, Y. Huo, R. G. Maunder, S. Kawade, and L. Hanzo, "Iterative joint source-channel decoding aided transmission of losslessly compressed video," in *IEEE Wireless Communications and Networking Conference*, (Shanghai, China), pp. 4629–4634, April 2013.
- [186] Y.-J. Wu and H. Ogiwara, "Symbol-interleaver design for turbo trellis-coded modulation," *IEEE Communications Letters*, vol. 8, pp. 632–634, October 2004.
- [187] R. Bauer and J. Hagenauer, "Symbol-by-symbol MAP decoding of variable length codes," in *ITG Conference on Source and Channel Coding*, (Munich, Germany), pp. 111–116, January 2000.
- [188] C. Brites, *Exploiting Correlation Noise Modeling in Wyner-Ziv Video Coding*. PhD thesis, Instituto Superior Técnico, Technical University of Lisbon, 2011.
- [189] R. W. Young and N. G. Kingsbury, "Frequency-domain motion estimation using a complex lapped transform," *IEEE Transactions on Image Processing*, vol. 2, pp. 2–17, January 1993.
- [190] A. Aaron, R. Zhang, and B. Girod, "Wyner-Ziv coding of motion video," in *Asilomar Conference on Signals, Systems and Computers*, vol. 1, (Pacific Grove, CA, USA), pp. 240–244, November 2002.

- [191] Y. Huo, T. Wang, R. G. Maunder, and L. Hanzo, "Two-dimensional iterative source-channel decoding for distributed video coding," *IEEE Communications Letters*, vol. 18, no. 1, pp. 90–93, 2014.
- [192] Y. Huo, T. Wang, R. G. Maunder, and L. Hanzo, "Motion-aware mesh-structured trellis for correlation modelling aided distributed multi-view video coding," *IEEE Transactions on Image Processing*, vol. 23, no. 1, pp. 319–331, 2014.
- [193] S. Cheng and Z. Xiong, "Successive refinement for the Wyner-Ziv problem and layered code design," *IEEE Transactions on Signal Processing*, vol. 53, pp. 3269–3281, August 2005.
- [194] Y. W. Huang, C. Y. Chen, C. H. Tsai, C. F. Shen, and L. G. Chen, "Survey on block matching motion estimation algorithms and architectures with new results," *Journal of Signal Processing Systems*, vol. 42, pp. 297–320, March 2006.
- [195] "FhG-HHI 3DV data." Available: <http://sp.cs.tut.fi/mobile3dtv/video-plus-depth/>.
- [196] "MERL Multiview Video Sequence." Available: <ftp://ftp.merl.com/pub/avetro/mvc-testseq>.
- [197] "H.264/AVC Reference Software." Available: <http://iphome.hhi.de/suehring/tml/>.
- [198] G. Bjøntegaard, "Calculation of average PSNR differences between RD-curves," in *ITU-T SG16 Q.6 Document, VCEG-M33*, (Austin, US), April 2001.

Author Index

- Aaron, A. 28–31, 153, 197–199, 201–205, 208, 212, 219
- Abou-Elailah, A. 29, 31
- Adikari, A. B. B. 28, 29
- Adrat, M. 22, 23, 147–149, 155, 156, 159, 161, 165, 166, 174, 186, 189–192, 194, 217, 240
- Agrafiotis, D. 5, 7
- Ahmad, S. 19, 20, 62, 64
- Akyildiz, I. F. 31, 197
- Al-Akaidi, M. M. 19, 20, 62, 64
- Alajel, K. M. 19, 21
- Alamri, O. 37, 63, 70, 88
- Anderson, J. 31
- Arnold, J. F. 5, 7
- Artigas, X. 28, 30, 208, 230
- Ascenso, J. 28, 30, 208, 230
- Ashikhmin, A. 32, 64, 93, 94
- Aydinlik, M. 18, 20, 21
- Bahl, L. R. 156, 161, 179, 216, 218, 238, 243
- Bauer, R. 185
- Berrou, C. 29, 71, 102, 204
- Bianchi, G. 12, 15, 133
- Biswas, M. 5, 7
- Bjøntegaard, G. 222, 226
- Blogh, J. 44
- Blogh, J. S. 63
- Borkenhagen, J. 22, 147
- Bouaynaya, N. 102
- Boyarinov, I. 18
- Brites, C. 28, 30, 195, 198, 202–209, 212, 219, 220
- Brüggen, T. 18, 19
- Butt, M. F. U. 38, 62, 121
- Cagnazzo, M. 29, 31
- Calderbank, A. R. 63
- Calvagno, G. 5, 7
- Canagarajah, C. N. 5, 7
- Caramanis, C. 19, 21
- Chang, Y. C. 18, 20, 21, 64, 101
- Chen, C. 29, 30, 198, 208
- Chen, C. Y. 213
- Chen, D. 28, 30, 208
- Chen, J. 66
- Chen, L. G. 213
- Chen, Y. 4, 23, 24, 30
- Cheng, S. 29, 31, 205, 219

- Cherriman, P. 1, 4, 6, 7, 9, 21, 29, 42, 44, 83, 147, 148, 164, 167, 175, 197
- Cheung, S.-C. 20, 22
- Chowdhury, K. R. 31, 197
- Chung, H. S.-H. 102
- Chung, J. Y. 20, 21, 35
- Clerc, M. 102
- Cocke, J. 156, 161, 179, 216, 218, 238, 243
- Cornelis, J. 54
- Crave, O. 54
- Cui, L. 29, 31
- Culler, D. 31
- Dalai, M. 28, 30, 208, 230
- De Cock, J. 29, 31
- Debono, C. 19, 21
- Deligiannis, N. 29, 31
- Detti, A. 12, 15, 133
- Dholakia, A. 66
- Doan, C. H. 147, 148
- Doufexi, A. 20, 22
- Dragotti, P. L. 28, 31, 208
- Dufaux, F. 29, 31
- Eiland, D. 147
- Eisenberg, Y. 18, 20, 62
- El-Hajjar, M. 8, 20, 34, 37, 38, 40, 62–65, 70, 84, 88, 94, 98, 100
- Eleftheriou, E. 66
- Emami, S. 147, 148
- Fan, G. 148
- Farah, J. 18, 20, 29, 31, 64
- Farrugia, R. 19, 21
- Fekri, F. 18, 19
- Feng, G. N. 101
- Fernando, W. A. C. 28, 29
- Fingscheidt, T. 22, 23, 147, 149, 150, 153, 155, 165, 166, 174, 176, 179, 186, 189–192
- Fisher, R. 148
- Flierl, M. 28, 30, 208
- Foschini, G. J. 63
- Fossorier, M. P. C. 66
- Franchi, N. 5, 7
- Frater, M. R. 5, 7
- Fresia, M. 23, 24
- Fumagalli, M. 5, 7
- Gabbouj, M. 4, 30
- Gallager, R. 18, 31, 81
- Gao, W. 28, 31, 198, 208
- Gasiba, T. 21
- Gavrilescu, A. I. 54
- Gehrig, N. 28, 31, 208
- Gharavi, H. 147
- Gilbert, J. M. 147, 148
- Girod, B. 28–31, 153, 197–199, 201–205, 208, 212, 219
- Glavieux, A. 29, 71, 102, 204
- Golden, G. D. 63
- Gomez-Barquero, D. 19, 20, 22, 62–64
- Gong, C. 19
- Görtz, N. 18, 22, 23, 147, 149, 174, 185, 186, 189–192
- Goyal, V. K. 6, 40, 42
- Grecos, C. 29, 31
- Guillemot, C. 54
- Guo, X. 28, 31, 198, 208

- Ha, H. 18, 20, 62–64, 90, 91, 101, 105–111, 134–142
- Hagenauer, J. 18, 19, 66, 93, 185
- Hallapuro, A. 30
- Halloush, M. 20, 22
- Hamzaoui, R. 19, 20, 62, 64
- Hannuksela, M. M. 4, 20, 30
- Hanzo, L. 1, 4, 6–10, 20–23, 28, 29, 34–42, 44, 62–65, 70, 83, 84, 88, 93, 94, 98, 100, 121, 147, 148, 155, 157, 160, 164, 165, 167, 174, 175, 177–179, 182, 184–187, 189–192, 197, 198, 204, 214, 217, 241, 246
- He, Z. 29, 30, 198, 208
- Heath, R. W. 19, 21
- Hellge, C. 19, 20, 22, 62–64
- Holub, P. 147
- Hong, S. E. 148
- Hou, Y. T. 20
- Hu, X.-Y. 66
- Huang, Y. W. 213
- Huo, Y. 8, 20, 34, 36, 38–41, 62–64, 84, 94, 98, 100, 174, 175, 177, 179, 184–187, 189–192, 198, 214, 217
- Hutanu, A. 147
- Imaizumi, H. 6, 8, 9
- Jafarkhani, H. 63
- Jelinek, F. 156, 161, 179, 216, 218, 238, 243
- Jeon, B. 5, 7
- Ji, W. 23, 24
- Jiang, W. 5, 6
- Johannesson, R. 18, 19
- Kamnoonwatana, N. 5, 7
- Kanade, T. 31
- Karim, H. 5, 7
- Katsaggelos, A. K. 18, 20, 62–64, 101
- Katsman, G. 18
- Kawade, S. 185
- Keller, T. 1, 22, 42
- Kellerer, W. 12, 15, 133
- Kennedy, J. 102
- Khalek, A. A. 19, 21
- Khalili, R. 97
- Kim, B. J. 5, 7
- Kim, S. S. 148
- Kingsbury, N. G. 197
- Kliwer, J. 22, 23, 147, 185, 186, 189–192
- Klomp, S. 28, 30, 208, 230
- Komiya, R. 18, 20, 21, 64, 101
- Kondo, A. 5, 7
- Kramer, G. 32, 64, 93, 94
- Kuan, E. L. 29, 147, 148
- Kubasov, D. 28, 30, 208, 230
- Kumar, V. 18, 19
- Kunert, T. 42
- Kweon, C. Y. 147, 148, 164
- Lainema, J. 4
- Lambert, P. 29, 31
- Lancini, R. 5, 7
- Laneman, J. N. 6
- Lee, C. S. 22
- Lee, S. W. 18, 20, 21, 64, 101
- Lee, W. Y. 148
- Li, H. 30
- Li, Y. 102
- Li, Z. 23, 24

- Li, Z. G. 101
- Liew, T. H. 20, 21, 157, 160, 217, 241, 246
- Lin, C. 5, 7
- Lin, S. N. 6, 40
- Lin, X. 101
- Ling, N. 101
- Ling, X. 147
- Liska, M. 147
- Liu, B. D. 147
- Liu, K. H. 147
- Liveris, A. D. 29
- Loreti, P. 12, 15, 133
- Lu, Y. 28, 31, 198, 208
- Luby, M. 19, 21, 81
- Lucas, B. D. 31
- Lukacs, M. 6
- Luthra, A. 6, 8, 9
- Maani, E. 18, 20, 62–64, 101
- Mainwaring, A. 31
- Majumdar, A. 28, 30, 31, 208
- Mark, J. W. 147
- Marpe, D. 4, 8–10, 62, 133
- Marx, F. 18, 20, 64
- Masnick, B. 18, 19, 62
- Matusik, W. 42
- Maunder, R. G. 23, 38, 62–64, 83, 93, 174, 175, 177, 179, 182, 184–187, 189–192, 198, 214, 217
- Melodia, T. 31, 197
- Merkle, P. 30
- Mertins, A. 22, 23, 147, 185, 186, 189–192
- Micallef, J. 19, 21
- Milani, S. 5, 7
- Milenkovic, O. 18, 19
- Muller, K. 30
- Munteanu, A. 29, 31, 54
- Mys, S. 29, 31
- Nasruminallah 10, 20, 21, 23, 147, 155, 165, 177, 178, 204
- Ng, S. X. 20, 21, 35, 121
- Ngo, C. 147, 148, 164
- Nguyen, K. 20, 22
- Nguyen, T. 20, 22
- Ni, S. 44
- Niu, H. 148
- Offer, E. 66, 93
- Ogiwara, H. 185
- Oh, J. 147, 148, 164
- Ortega, A. 5, 6
- Otnes, R. 84
- Ouaret, M. 28, 30, 208, 230
- Papke, L. 66, 93
- Park, C. 148
- Paruchuri, R. 147
- Pavlushkov, V. 18, 19
- Pearlman, W. A. 5, 7, 31
- Pepe, D. 148
- Pereira, F. 28, 30, 198, 202–208, 212, 220
- Pérez-Cruz, F. 23, 24
- Pesquet-Popescu, B. 29, 31
- Pfister, H. 42
- Piechocki, R. J. 20, 22
- Pisa, C. 12, 15, 133

- Pishro-Nik, H. 18, 19
- Polastre, J. 31
- Poor, H. V. 23, 24
- Popescu, B. P. 54
- Proto, F. S. 12, 15, 133
- Puri, R. 28, 30, 31, 208
- Qin, S. 148
- Qin, X. 147, 148, 164
- Radha, H. 20, 22
- Rahnavard, N. 18, 19
- Ramasubramonian, A. K. 5, 7
- Ramchandran, K. 28, 30, 31, 198, 208
- Rane, S. 28–31, 153, 197–199, 201–205, 208, 219
- Rappaport, T. S. 148
- Raviv, J. 156, 161, 179, 216, 218, 238, 243
- Rebollo-Monedero, D. 28, 29, 153, 197–199, 201–205, 208, 219
- Reibman, A. R. 6, 40
- Sadka, A. 5, 7
- Said, A. 31
- Salamatian, K. 97
- Salehi, M. 18, 20, 21
- Sali, A. 5, 7
- Sayood, K. 22, 147
- Schelkens, P. 54
- Schierl, T. 19, 20, 22, 62–64
- Schonfeld, D. 102
- Schwarz, H. 4, 8–10, 62, 133
- Sejdinović, D. 20, 22
- Setton, E. 28, 29, 198, 203, 208
- Shannon, C. E. 22, 23, 27, 147, 166
- Shao, H. 147, 148, 164
- Shapiro, J. M. 5, 7
- Shen, C. F. 213
- Shen, X. S. 147
- Shokrollahi, A. 21
- Shung, C. B. 147, 148
- Singh, H. 147, 148, 164
- Skorupa, J. 29, 31
- Slepian, D. 23, 24, 29, 197
- Slowack, J. 29, 31
- Smolic, A. 30
- Spittka, J. 22, 147, 149, 155, 159, 174
- Stankovic, L. 29, 31
- Stankovic, V. 29, 31
- Stedman, R. 147
- Steele, R. 147
- Stockhammer, T. 20, 21
- Streit, J. 1, 4, 7, 9, 28, 29, 41, 44, 64, 147, 164, 167, 175, 197
- Sullivan, G. J. 6, 8, 30
- Szewczyk, R. 31
- Tarokh, V. 63
- Tee, R. Y. S. 20, 21, 157, 160, 217, 241, 246
- ten Brink, S. 32, 63, 64, 83, 92–94, 149, 182
- Thakolsri, S. 12, 15, 133
- Thitimajshima, P. 29, 71, 102, 204
- Thobaben, R. 22, 23
- Thomas, T. 6
- Thorpe, S. R. 147
- Tien, C. W. 147
- Tillo, T. 5, 7
- Tsai, C. H. 213

- Tsai, W.-J. 5, 7
Tse, D. N. C. 6
Tubaro, S. 5, 7
Tüchler, M. 84

Ugur, K. 4

Valenzuela, R. A. 63
Van De Walle, R. 29, 31
Varodayan, D. 28–30, 201, 208
Vary, P. 18, 19, 22, 23, 147–150, 153, 155, 156, 159, 161, 165, 166, 174, 176, 179, 186, 189–192, 194, 217, 240
Vary, R. 22, 147, 149, 155, 159, 174
Verdicchio, F. 54
Verdú, S. 23, 24
Vetro, A. 6, 8, 30
Vlahoyiannatos, S. 22
Šenk, V. 20, 22
Vukobratović, D. 20, 22

Wallace, G. K. 2
Wang, H. 18, 20, 62
Wang, J. 23, 31, 208
Wang, S. 29, 31
Wang, T. 174, 175, 177, 179, 184, 186, 187, 189–192, 198, 214, 217
Wang, X. 19
Wang, Y. 6, 19, 21, 40
Wang, Y.-K. 4
Watson, M. 21
Weerakkody, W. 28, 29
Wei, S. T. 147
Wenger, S. 10
Widmer, J. 12, 15, 133
Wiegand, T. 4, 8–10, 19, 20, 22, 30, 62–64, 133
Wolf, J. 18, 19, 23, 24, 29, 62, 197
Wolniansky, P. W. 63
Wong, C.H. 83
Woods, J. W. 5, 7
Wornell, G. W. 6
Worrall, S. 5, 7
Wu, D. 20
Wu, F. 28, 31, 198, 208
Wu, N. 37, 63, 70, 88
Wu, S. 101
Wu, Y.-J. 185
Wyner, A. 27–29, 197

Xiang, W. 19, 21
Xin, Y. 147
Xiong, H. 29, 30, 198, 208
Xiong, Z. 5, 7, 20, 22, 29, 205, 219
Xu, Y. 8

Yang, F. 20
Yang, J. F. 147
Yang, L.-L. 23
Yang, X. K. 101
Yap, K.-H. 23
Yeap, B. 20, 21, 157, 160, 217, 241, 246
Yeo, C. 28, 31, 198, 208
Yim, C. 18, 20, 62–64, 90, 91, 101, 105–111, 134–142
You, H.-Y. 5, 7
Young, R. W. 197
Yu, S. 29, 30, 198, 208
Yue, G. 19

-
- | | |
|---|------------------------------|
| Zhai, F. 18, 20, 62 | Zhao, D. 28, 31, 198, 208 |
| Zhan, Z.-H. 102 | Zhao, Y. 5, 7 |
| Zhang, J. 30, 102 | Zhu, C. 23, 38, 39, 101, 185 |
| Zhang, Q. 20 | Zhu, L. 30 |
| Zhang, R. 28, 29, 198, 199, 203, 208, 212 | Zhu, W. 20 |
| Zhang, Rui 28–31, 208 | Zito, D. 148 |
| Zhang, T. 8 | Ziv, J. 27–29, 197 |
| Zhang, Y. 20, 23, 29, 30, 198, 208 | Zuo, X. 38, 62–64 |
| Zhang, Y.-Q. 20 | Zyablov, V. V. 18, 19 |

Subject Index

- 2D, 149–151, 164, 175, 195, 204, 212
- 3D, 149, 174, 181–183
- 3D-SPIHT, 7
- 4CIF, 13
- AA, 83, 85
- ACK, 21
- APP, 19, 23, 158, 159, 216, 238, 240, 243, 245
- ARQ, 148
- ASP, 4
- AVC, 4, 9, 30, 54, 102, 199, 206, 207, 209, 213, 221, 228–230
- AVS, 4
- B-frame, 7, 54, 103, 120, 133, 167, 184
- BCJR, 157, 160, 161, 216–218, 238, 239, 243
- BEC, 18, 19, 63, 64
- BER, 37, 83, 88, 105, 111–114, 124–127, 135–137, 139, 166, 168, 169, 173, 176, 186, 187, 189, 212, 221, 222
- BL, 8, 34, 37, 40, 63, 67–69, 71–73, 82–85, 88–90, 92, 95, 97, 98, 132, 137, 141, 144
- BP, 29
- BPSK, 92, 132, 134, 144, 164, 165, 175, 180, 181
- BS, 31
- BT-MDC, 45, 47, 54, 56, 58, 60
- CB, 185
- CC, 18
- CGS, 9, 13
- CIF, 2, 8, 102, 164, 165, 174, 203, 205–208, 219, 220
- CN, 28, 30
- CND, 64, 66, 71, 72, 75, 78, 80, 90, 92–95, 98, 144
- CPBs, 10
- CRC, 72, 103, 121, 133
- DC, 44
- DC-MDVC, 7
- DCT, 30
- DID, 9, 12
- DISCOVER, 28, 30
- DMVC, iv, 28, 35, 198, 208
- DP, 10
- DPCM, 1
- DPT, 195, 196
- DV, 3
- DVC, iv, 28, 29, 31, 153, 197, 198, 235, 236
- EC, 177
- EEP, 84, 85, 88, 103–105, 108–111, 116, 117, 124, 126, 129, 130, 134, 137, 140, 144, 145, 148
- EL, 8, 40, 62, 63, 67–69, 71, 73, 82–85, 88–90, 92, 94, 95, 98, 99, 104, 120, 129, 133, 139, 144
- ELs, 34, 37
- ERC, 28
- EXIT, 37, 63, 84–88, 149, 180, 182, 183
- EZW, 7
- FEC, iv, 6, 18, 34, 62, 64, 67, 69, 71, 72, 74, 82, 90, 92, 103, 110, 111, 116,

- 117, 142, 144, 148, 149, 165, 168,
185, 192, 198, 232, 235, 236
- FEC-, 102
- FOMM, 148, 149, 155, 165–167, 173, 184,
188–190
- FPS, 54, 57, 103, 120, 132, 164, 180, 203,
220
- GOP, 18, 120, 133, 199, 202, 203, 205, 208,
213, 221–226, 234, 236
- H.120, 1
- H.26, 230
- H.261, 2
- H.263, 3, 21, 29
- H.264, 4, 9, 10, 23, 29, 37, 44, 54, 60, 63,
64, 67, 73, 89, 102, 103, 108, 117,
120, 134, 141, 144, 145, 164–168,
173, 184–186, 188–190, 193, 199,
202, 205–207, 209, 213, 221, 228,
229
- H.264/AVC, 10, 89, 103, 120, 132, 133,
149, 204–206, 208, 230
- H.265, 5
- H.26x, 197
- HD, 3, 148
- HEVC, 5
- HOCT, 42
- HOCT-MDC, 43, 45, 49, 52, 54, 57, 58,
60, 232
- HQAM, 19
- HSMD, 151, 153, 164, 176–179, 181, 192,
204
- I-frame, 21, 103, 120, 133, 167, 184, 205
- IEC, 3
- IEEE, 148
- IF-MDVC, 7
- IFPI, 214, 215
- IHVSM, 35, 148–151, 153, 163–167, 173,
177, 184, 188, 189, 192, 202–208,
228, 230, 231, 233
- IL, 68, 70, 72, 73, 81, 99, 102, 109–111,
116–120, 124, 126, 129, 130, 132,
134, 137, 139–142, 144
- IL-, 137
- IL-FEC, 34, 35, 37–39, 62, 64, 68, 70, 73,
74, 82, 84, 89, 90, 95, 116, 141,
142, 144, 231, 232, 236
- IL-RSC, 105, 108, 109
- IL-turbo, 111
- ISCD, iv, 22–24, 35, 148, 149, 164, 174,
198, 222, 235–237
- ISDN, 2
- ISMD, 176, 178, 179, 181
- ISO, 3
- ITU, 1
- ITU-T, 3, 30, 147, 197
- JM, 54, 102, 163, 186, 221
- JMVC, 57, 221
- JMVM, 30
- JPEG, 2, 205–207
- JSCC, 23, 147
- JSCD, 18
- JSVM, 12, 14, 120, 132, 133
- JVT, 6, 9, 30
- LA-FEC, 19
- LDPC, 18, 23, 24, 28, 29, 81
- LDPCA, 28, 29
- LLR, 37, 39, 64, 66, 70–72, 80, 82, 92–95,
98, 134, 144, 150, 151, 153, 156,
159, 160, 178, 179, 216, 217, 242,
245
- LMS, 215
- LSSTC, 63, 64, 68, 70, 85, 88, 102, 105,
108, 109, 111, 113, 116–120, 124,
126, 129, 130, 132, 142–144
- LT, 81, 235
- LUT, 37, 39, 64, 91–93, 95, 98, 99, 133,
134, 144
- LW-EZEP, 18
- M3DISC, 149, 174, 176, 181, 182, 184–
190, 193, 231
- MAC, 19
- MAP, 153, 154, 157, 167–170, 173, 179,
212, 238
- MB, 5, 103, 152, 153, 213, 214, 219

- MCFI, 209, 212, 215
 MDC, 5, 6, 34, 40, 45, 54, 57, 58, 232–234
 MDD, 45, 54
 MDDVC, 7
 MDE, 45, 52
 MGS, 9, 13, 120, 133
 MI, 37, 39, 64, 83, 84, 88, 91–95, 98, 134, 144, 183
 MIMO, 37, 63, 68, 141
 MMSE, 153–155, 165–170, 173, 179, 184, 186–189, 192
 MMSTT, 162, 165, 181, 193, 196, 205, 220
 Motion-JPEG, 2
 MPEG, 2, 9, 30, 147, 197
 MPEG-2, 3, 44
 MPEG-4, 4
 MPEG2, 29
 MRF, 22, 29, 185, 188, 189
 MSB, 199, 205, 207, 208, 210, 212, 219, 220
 MSE, 11, 163, 186
 MSSM, 35, 208–210, 212, 216–218, 220–227, 230, 231, 233, 237
 MSSM-, 212
 MSSTT, 218, 220
 MV, iv, 198, 213, 214, 230
 MVC, 4, 9, 28, 30, 41, 54, 57, 208, 210
 MVI, 213–215
 MVP, 9
 MVs, 30
 NALU, 11, 12, 17, 103, 117, 121, 124, 130, 133
 NALUs, 9
 NCR, 167, 181
 P-frame, 21, 54, 103, 120, 133, 167, 184, 205
 PC, 9
 PDF, 50
 PDWZ, iv, 28, 30, 198, 199, 203, 205–208, 220, 222–226, 230, 233
 PER, 90–93, 95, 97–99
 PF, 29, 31
 PHY, 19
 PLR, 20, 83
 PM, 63, 102
 PRE, 30
 PSNR, 10, 54, 55, 90, 103, 108–110, 113, 115, 116, 118, 121, 128–130, 132, 133, 137, 139, 140, 142–145, 163, 167, 170, 173, 174, 176, 177, 186–188, 192, 205, 208, 219, 222, 226, 228, 229, 234
 QAM, 21
 QCIF, 2, 8, 54, 60, 120, 132, 134, 164–167, 169, 170, 173, 180, 181, 195
 QID, 9, 13
 QoE, 21
 QoS, 21, 23, 24
 QP, 54, 57, 120, 133
 QPs, 7
 QPSK, 33, 68, 70, 83, 85
 RBSP, 11
 RCPC, 18, 19
 RD, 203, 205, 220
 RDO, 54
 RGB, 41
 RSC, 21, 39, 63, 64, 83, 84, 86–88, 90, 92, 93, 95, 97–99, 102, 105, 108–111, 116–119, 121, 122, 124, 132, 134, 137, 139–142, 144, 157, 158, 163–168, 173–182, 184, 185, 187–190, 192, 193, 219, 233
 SBC, 23, 155
 SBSD, 23, 155, 156, 158, 165, 167, 173, 184, 186, 188, 189
 SC, 185
 SD, 54
 SDC, 54, 60
 SECCC, 63, 120–122, 124, 126, 129, 130, 132, 142, 144
 SI, 29, 198, 201, 202, 204, 205, 207, 212, 215, 216, 226, 227
 SLDPCA, 28, 29
 SMPTE, 4

- SNR, 92, 93, 103, 109, 110, 113, 166
- SP, 4
- SPIHT, 7, 31
- SVC, 4, 9, 12, 14, 37, 62, 63, 89, 120, 121,
130, 133, 134, 141, 144, 145
- SVEF, 15
- SW, 23, 27, 29, 197

- T-MDC, 36, 38, 41, 54, 61, 231, 236
- TCP-IP, 6, 40, 53
- TDWZ, 28, 30, 203, 230
- TID, 9, 12
- TS, 13
- Turbo, 204
- turbo, 85, 88, 111, 113, 116–119, 142, 204
- TV, 9, 42, 63, 64, 141

- UDE, 18
- UEP, 18, 19, 21, 34, 62–64, 73, 85, 89,
102–105, 108–111, 113, 116–120,
124, 126, 129, 130, 132, 134, 137,
139, 141, 142, 144, 145, 148
- UHD, 5

- VBLAST, 63
- VC-1, 4
- VCEG, 4
- VLC, 1, 23, 185, 188, 189
- VND, 64, 66, 71, 72, 75, 77, 78, 90, 92, 93,
95, 98, 144
- VSMD, 151, 153, 164, 165, 176, 178, 179,
181, 192, 204

- WMV9, 4
- WPAN, 147, 148
- WVSN, 29
- WZ, 27–29, 35, 153, 197–199, 201–205,
207–210, 212, 213, 215, 216, 219–
222, 226–228, 230, 233, 235–237

- XOR, 37, 38, 67, 68, 74, 94, 142, 232

- Y-PSNR, 55, 173, 187–189
- YUV, 17, 41, 102, 120, 132, 134, 164, 165,
180, 181, 185, 203, 219