



**UNIVERSIDADE TÉCNICA DE LISBOA  
INSTITUTO SUPERIOR TÉCNICO**

# Advances on Distributed Video Coding

**Catarina Isabel Carvalheiro Brites**

(Licenciada)

Dissertação para obtenção do Grau de Mestre em Engenharia Electrotécnica e de Computadores

**Orientador:** Doutor Fernando Manuel Bernardo Pereira

**Presidente:** Doutor José Manuel Nunes Leitão

**Vogais:** Doutor Armando José Formoso de Pinho

Doutor Fernando Manuel Bernardo Pereira

**Dezembro 2005**







To my family.



# Acknowledgments

There are many people who have contributed one way or the other to make this dissertation possible. I would like to thank all those people for their assistance.

Special thanks to Professor Fernando Pereira, supervisor of this dissertation, for allowing me to choose such an interesting subject like Distributed Video Coding. His guidance, enthusiasm, constructive criticism, outstanding support and patience were very important factors during the last two years. I can not forget the stimulating and fruitful discussions which so much have contributed to make this work possible. Thanks also for enabling me to attend the ICVIP'05 conference and to participate in the European Network of Excellence VISNET (Networked Audiovisual Media Technologies).

Thanks to João Ascenso, for his unconditional friendship and words of incentive during this dissertation. Thanks also for helping me in the frame interpolation framework development, for the valuable suggestions and technical support.

I wish to thanks my colleagues of the Image Group, for their suggestions and availability to help me whenever it was necessary and also for the pleasant working environment.

Thanks to Professor Fernando Nunes and Professor Rui Dinis for sharing their time with me to exchange some ideas about turbo codes.

Warm words of gratitude are addressed to my family, specially my parents, Armando and Isabel, my sister Sandrina and my grandmother Maria de Jesus whose love, understanding, patience and encouragement has enabled me to find energy in the most difficult moments.

Thanks to all my friends, for their friendship and support.





# Abstract

Distributed Video Coding (DVC) is a new coding paradigm based on two major Information Theory results: the Slepian-Wolf (1973) and Wyner-Ziv Theorems (1976). This new video coding paradigm allows exploiting the source statistic, partially or totally, at the decoder only. A particular case of distributed video coding is the Wyner-Ziv video coding. In this scenario, two correlated sources are independently encoded using separated encoders and the bitstreams associated to each source are jointly decoded exploiting the correlation between them.

Although the distributed coding study dates back to the 1970's, efforts towards developing practical solutions of Wyner-Ziv video coding are more recent. Emerging applications (such as wireless, low-power surveillance systems and mobile camera phones among others) with encoding requirements quite different from those targeted by the MPEG-x and H.26x video coding standards have stimulated such efforts. In the MPEG-x and H.26x standards, the correlation between temporally adjacent frames is exploited through a complex motion estimation task which leads to a high complexity encoder. Since the correlation between temporally adjacent frames in Wyner-Ziv video coding is performed only at the decoder, the encoder can typically present a low complexity. Improved error resilience is another major functionality of this new video coding paradigm since the usual encoder prediction loop and the associated error propagation do not exist anymore.

The main objectives of the current Thesis are: 1) Perform a revision of the state-of-the-art on distributed video coding. 2) Develop and implement distributed video coding solutions for the pixel and the transform domains, notably frame interpolation techniques with motion compensation to improve the coding efficiency of the distributed video solutions developed.

## Keywords:

Distributed video coding, Wyner-Ziv, Slepian-Wolf, frame interpolation, channel codes, low encoding complexity, error resilience.



# Resumo

A codificação distribuída de vídeo é um novo paradigma de codificação de vídeo baseado em dois importantes resultados da Teoria da Informação: os Teoremas de Slepian-Wolf (1973) e de Wyner-Ziv (1976). Este novo paradigma da codificação de vídeo permite explorar a estatística da fonte, parcial ou totalmente, apenas no decodificador. Um caso particular da codificação distribuída de vídeo é a codificação de vídeo Wyner-Ziv. Neste cenário, duas fontes correlacionadas são independentemente codificadas usando codificadores separados e os fluxos binários associados a cada fonte são conjuntamente decodificados, explorando a correlação entre eles.

Apesar do estudo da codificação distribuída remontar aos anos 70, só recentemente se verificaram esforços mais intensos no sentido de desenvolver soluções práticas de codificação de vídeo Wyner-Ziv. O aparecimento recente de aplicações (tais como redes de vigilância de baixa potência e vídeo em telefones móveis entre outras) com requisitos de codificação bastante diferentes daqueles contemplados pelas normas de codificação de vídeo MPEG-x e H.26x estimularam tais esforços. Nas normas MPEG-x e H.26x, a correlação entre duas imagens adjacentes é explorada através da complexa operação de estimação de movimento o que conduz a um codificador de elevada complexidade. Uma vez que a exploração da correlação entre duas imagens temporalmente adjacentes na codificação de vídeo Wyner-Ziv é realizada apenas no decodificador, o codificador pode tipicamente apresentar baixa complexidade. A resiliência a erros é outra importante funcionalidade deste novo paradigma de codificação de vídeo uma vez que a tradicional malha de predição no codificador e a propagação de erros associada a essa mesma malha não existe ao não se explorar a correlação do sinal no codificador.

Os principais objectivos da presente Tese são: 1) Realizar uma revisão bibliográfica do estado da arte na área da codificação distribuída de vídeo. 2) Desenvolver e implementar soluções eficientes de codificação distribuída de vídeo para os domínios do pixel e da transformada, nomeadamente técnicas de interpolação de imagem com compensação de movimento, com o intuito de melhorar a eficiência das soluções de codificação distribuída de vídeo desenvolvidas.

## Palavras-chave:

Codificação distribuída de vídeo, Wyner-Ziv, Slepian-Wolf, interpolação de imagem, códigos de canal, codificação de baixa complexidade, resiliência a erros.



# Contents

<b>Chapter 1</b>	<b>Introduction</b>	<b>1</b>
1.1	Information Theory Background .....	2
1.1.1	Slepian-Wolf Theorem .....	3
1.1.2	Wyner-Ziv Theorem .....	6
1.2	Target Applications .....	8
1.3	Main Objectives of the Thesis .....	11
1.4	Outline of the Thesis .....	12
1.5	Publications .....	13
<b>Chapter 2</b>	<b>Wyner-Ziv Video Coding: a Review</b>	<b>15</b>
2.1	Overview on Distributed Coding .....	16
2.1.1	Basic Wyner-Ziv Coding Architecture .....	19
2.1.2	Transforming in the Basic Wyner-Ziv Codec .....	20
2.1.3	Quantizing in the Basic Wyner-Ziv Codec .....	21
2.1.4	Slepian-Wolf Encoding in the Basic Wyner-Ziv Codec .....	22
2.1.5	Slepian-Wolf Decoding in the Basic Wyner-Ziv Codec .....	23
2.1.6	Reconstructing in the Basic Wyner-Ziv Codec .....	23
2.1.7	Inverse Transforming in the Basic Wyner-Ziv Codec .....	24
2.2	Most Relevant Wyner-Ziv Video Coding Solutions .....	24
2.2.1	Stanford Wyner-Ziv Low-Complexity Video Coding Solution .....	25
2.2.1.1	Encoding Procedure .....	27
2.2.1.2	Decoding Procedure .....	29
2.2.1.3	Some Experimental Results .....	30
2.2.2	Stanford Wyner-Ziv Robust Video Coding Solution .....	32
2.2.2.1	FEP Encoding Procedure .....	34
2.2.2.2	FEP Decoding Procedure .....	35
2.2.2.3	Some Experimental Results .....	36

2.2.3	Berkeley Wyner-Ziv Robust Video Coding Solution .....	38
2.2.3.1	PRISM Encoding Procedure .....	40
2.2.3.2	PRISM Decoding Procedure .....	42
2.2.3.3	Some Experimental Results .....	43
2.3	Final Remarks .....	45
<b>Chapter 3</b>	<b>IST-Pixel Domain Wyner-Ziv Codec</b>	<b>47</b>
3.1	IST-Pixel Domain Wyner-Ziv Codec Architecture .....	48
3.1.1	Quantizer in the IST-PDWZ Codec .....	50
3.1.2	Slepian-Wolf Encoder in the IST-PDWZ Codec .....	50
3.1.3	Frame Interpolation in the IST-PDWZ Codec .....	58
3.1.3.1	Forward Motion Estimation .....	59
3.1.3.2	Bidirectional Motion Estimation .....	60
3.1.3.3	Spatial Motion Smoothing Based Estimation .....	61
3.1.3.4	Bidirectional Motion Compensation .....	63
3.1.4	Slepian-Wolf Decoder in the IST-PDWZ Codec .....	64
3.1.4.1	SISO Decoding Algorithm .....	66
3.1.4.2	Decision Operation .....	73
3.1.5	Reconstruction in the IST-PDWZ Codec .....	73
3.2	IST-PDWZ Experimental Results .....	75
3.2.1	<i>Foreman</i> Test Sequence Evaluation .....	76
3.2.2	<i>Mother and Daughter</i> Test Sequence Evaluation .....	77
3.2.3	<i>Coastguard</i> Test Sequence Evaluation .....	78
3.2.4	<i>Stefan</i> Test Sequence Evaluation .....	79
3.2.5	IST-PDWZ versus H.263+ Intraframe Coding Gains .....	80
3.3	Final Remarks .....	80
<b>Chapter 4</b>	<b>IST-Transform Domain Wyner-Ziv Codec</b>	<b>83</b>
4.1	IST-Transform Domain Wyner-Ziv Codec Architecture .....	85
4.1.1	Discrete Cosine Transform in the IST-TDWZ Codec .....	87
4.1.2	Quantizer in the IST-TDWZ Codec .....	90
4.1.3	Reconstruction in the IST-TDWZ Codec .....	97
4.2	IST-TDWZ Experimental Results .....	99
4.2.1	<i>Foreman</i> Test Sequence Evaluation .....	101
4.2.2	<i>Mother and Daughter</i> Test Sequence Evaluation .....	102
4.2.3	<i>Coastguard</i> Test Sequence Evaluation .....	103

4.2.4	<i>Stefan</i> Test Sequence Evaluation .....	104
4.2.5	IST-TDWZ versus H.263+ Intraframe Coding Gains .....	105
4.3	Final Remarks .....	105
<b>Chapter 5</b>	<b>Conclusions and Future Work</b>	<b>107</b>
5.1	Future Work .....	109
<b>Annex A</b>	<b>Video Test Sequences</b>	<b>113</b>
A.1	<i>Coastguard</i> Sequence .....	114
A.2	<i>Foreman</i> Sequence .....	115
A.3	<i>Mother and Daughter</i> Sequence .....	116
A.4	<i>Stefan</i> Sequence .....	117
<b>References</b>		<b>119</b>





# List of Figures

<i>Figure 1.1 – Ideal coding configuration for some emerging video applications</i> .....	2
<i>Figure 1.2 – Illustration of distributed source coding with multiple dependent video sequences</i> .....	3
<i>Figure 1.3 – Traditional coding paradigm</i> .....	3
<i>Figure 1.4 – Distributed compression of two statistically dependent discrete random sequences, X and Y, independently and identically distributed (i.i.d.)</i> .....	4
<i>Figure 1.5 – Achievable rate region following the Slepian-Wolf theorem [4]</i> .....	5
<i>Figure 1.6 – Relationship between channel coding and Slepian-Wolf coding</i> .....	5
<i>Figure 1.7 – Lossy compression with decoder side information</i> .....	6
<i>Figure 1.8 – Video surveillance scenario</i> .....	9
<i>Figure 1.9 – Wireless mobile video scenario</i> .....	9
<i>Figure 1.10 – Large camera array scenario</i> .....	10
<i>Figure 2.1 – Block diagram of the basic Wyner-Ziv codec</i> .....	19
<i>Figure 2.2 – Three-node network considered in [25]</i> .....	22
<i>Figure 2.3 – Stanford Wyner-Ziv video codec architecture [19]</i> .....	26
<i>Figure 2.4 – Average PSNR for the Salesman sequence [19]</i> .....	31
<i>Figure 2.5 – Average PSNR for the Hall Monitor sequence [19]</i> .....	31
<i>Figure 2.6 – Systematic lossy Forward Error Protection (FEP) system architecture [20]</i> .....	33
<i>Figure 2.7 – Reed-Solomon code applied across slices of an entire frame [20]</i> .....	35
<i>Figure 2.8 – Performance comparison between FEP and FEC systems for the same parity information rate [20]</i> .....	37
<i>Figure 2.9 – Performance comparison in terms of visual quality between FEP and FEC systems for a symbol error rate of <math>10^{-3}</math> [20]</i> .....	37
<i>Figure 2.10 – PRISM encoder architecture [40]</i> .....	38
<i>Figure 2.11 – PRISM decoder architecture [40]</i> .....	39
<i>Figure 2.12 – Selective encoding for the various transform coefficients within a block [40]</i> ...	41
<i>Figure 2.13 – Bitstream syntax at the block level [40]</i> .....	42

<i>Figure 2.14 – Comparison of the rate-distortion performance for PRISM and H.263+ (both intra and inter coding modes) when no frames are lost [40]</i> .....	44
<i>Figure 2.15 - Impact of a frame loss in PRISM (left) and H.263+ (right): each column, from top to bottom, represents the first, third and fourteenth decoded frames for the Football sequence [40]</i> .....	45
<i>Figure 3.1 – IST-PDWZ codec architecture</i> .....	48
<i>Figure 3.2 – Turbo encoder structure using a parallel concatenation of two identical constituent Recursive Systematic Convolutional encoders (<math>RSC_1</math> and <math>RSC_2</math>)</i> .....	51
<i>Figure 3.3 – Interleaving and deinterleaving of a 10-bit sequence</i> .....	51
<i>Figure 3.4 – Rate <math>\frac{1}{2}</math> (one input, two output) constituent recursive systematic convolutional encoder with memory 4 (16 states) and generator matrix given by equation (3.6)</i> .....	53
<i>Figure 3.5 – Trellis diagram of a RSC encoder with a generator matrix given by (3.6)</i> .....	55
<i>Figure 3.6 – Parity sequence of 16 bit length at the output of a RSC encoder</i> .....	56
<i>Figure 3.7 – Possible composition of the first block of <math>(16/4) = 4</math> bits obtained by division of the sequence illustrated in Figure 3.6 into <math>P = 4</math> blocks</i> .....	57
<i>Figure 3.8 – Frame interpolation framework</i> .....	59
<i>Figure 3.9 – Uncover pixels in the interpolated frame due to the rigid block-based motion estimation algorithm</i> .....	60
<i>Figure 3.10 – Selection of the motion vector</i> .....	60
<i>Figure 3.11 – Bidirectional motion estimation</i> .....	61
<i>Figure 3.12 – Frame #7 of the Foreman QCIF sequence: (a) with and (b) without spatial motion smoothing</i> .....	62
<i>Figure 3.13 – Neighboring blocks of block B for weighted median vector filter</i> .....	63
<i>Figure 3.14 – Turbo decoder implementation using two identical soft-input, soft-output (SISO) decoders</i> .....	64
<i>Figure 3.15 – Residual distribution for the Foreman QCIF video sequence</i> .....	71
<i>Figure 3.16 – Reconstruction function for a 4-level uniform scalar quantizer</i> .....	74
<i>Figure 3.17 – IST-PDWZ rate-distortion performance for the Foreman test sequence</i> .....	77
<i>Figure 3.18 – IST-PDWZ rate-distortion performance for the Mother and Daughter test sequence</i> .....	78
<i>Figure 3.19 – IST-PDWZ rate-distortion performance for the Coastguard test sequence</i> .....	79
<i>Figure 3.20 – IST-PDWZ rate-distortion performance for the Stefan test sequence</i> .....	79
<i>Figure 3.21 – PSNR gains over H.263+ intraframe coding for the Foreman, Mother and Daughter, Coastguard and Stefan QCIF video sequences</i> .....	80
<i>Figure 4.1 – IST-TDWZ codec architecture</i> .....	85
<i>Figure 4.2 – Position ordering inside a <math>4 \times 4</math> DCT coefficients block</i> .....	89
<i>Figure 4.3 – Uniform scalar quantizer with quantization interval width <math>W</math> for the DC coefficient</i> .....	91

<i>Figure 4.4 – DCT coefficients distribution for the lowest spatial frequency AC band (<math>b_2</math>) of the Foreman QCIF sequence</i> .....	91
<i>Figure 4.5 – DCT coefficients distribution for the highest spatial frequency AC band (<math>b_{16}</math>) of the Foreman QCIF sequence</i> .....	92
<i>Figure 4.6 – Uniform scalar quantizer without symmetric quantization interval around the zero amplitude</i> .....	92
<i>Figure 4.7 – Uniform scalar quantization scenario</i> .....	93
<i>Figure 4.8 – Uniform scalar quantizer with a symmetric quantization interval around the zero amplitude</i> .....	93
<i>Figure 4.9 – Eight quantization matrices associated to different IST-TDWZ codec rate-distortion performances</i> .....	96
<i>Figure 4.10 – Reconstruction procedure of each <math>b_k</math> band DCT coefficient: (a) Case I, (b) Case II, (c) Case III</i> .....	98
<i>Figure 4.11 – IST-TDWZ rate-distortion performance for the Foreman test sequence</i> .....	102
<i>Figure 4.12 – IST-TDWZ rate-distortion performance for the Mother and Daughter test sequence</i> .....	103
<i>Figure 4.13 – IST-TDWZ rate-distortion performance for the Coastguard test sequence</i> .....	104
<i>Figure 4.14 – IST-TDWZ rate-distortion performance for the Stefan test sequence</i> .....	104
<i>Figure 4.15 – PSNR gains over H.263+ intraframe coding for the Foreman, Mother and Daughter, Coastguard and Stefan QCIF video sequences</i> .....	105
<i>Figure A.1 – Coastguard sequence: (a) frame 0; (b) frame 60; (c) frame 120; (d) frame 180; (e) frame 240; (f) frame 299</i> .....	114
<i>Figure A.2 – Foreman Sequence: (a) frame 0; (b) frame 80; (c) frame 160; (d) frame 240; (e) frame 320; (f) frame 399</i> .....	115
<i>Figure A.3 – Mother and Daughter sequence: (a) frame 0; (b) frame 192; (c) frame 384; (d) frame 576; (e) frame 768; (f) frame 960</i> .....	116
<i>Figure A.4 – Stefan sequence: (a) frame 0; (b) frame 60; (c) frame 120; (d) frame 180; (e) frame 240; (f) frame 299</i> .....	117



# List of Tables

<i>Table 2.1 – Main simulation conditions for the Foreman video sequence .....</i>	<i>36</i>
<i>Table 3.1 – Possible RSC encoder state transitions, <math>S_{k-1} \rightarrow S_k</math>, and output bits <math>(u_k^s, u_k^p)</math> given the RSC encoder input bit <math>u_k</math> .....</i>	<i>55</i>
<i>Table 3.2 – Main characteristics of the video test sequences .....</i>	<i>75</i>
<i>Table 4.1 – Main characteristics of the video test sequences .....</i>	<i>99</i>
<i>Table A.1 – Main characteristics of the test sequences .....</i>	<i>113</i>



# List of Acronyms

<b>AC</b>	Alternating Current
<b>AVC</b>	Advanced Video Coding
<b>AWGN</b>	Additive White Gaussian Noise
<b>CRC</b>	Cyclic Redundancy Check
<b>dB</b>	Decibel
<b>DC</b>	Direct Current
<b>DCT</b>	Discrete Cosine Transform
<b>DISCOVER</b>	DIStributed COding for Video sERvices
<b>DSC</b>	Distributed Source Coding
<b>DVC</b>	Distributed Video Coding
<b>FEC</b>	Forward Error Correction
<b>FEP</b>	Forward Error Protection
<b>GOP</b>	Group Of Pictures
<b>HVS</b>	Human Visual System
<b>IDCT</b>	Inverse Discrete Cosine Transform
<b>IEC</b>	International Electrotechnical Commission
<b>IRA</b>	Irregular Repeat-Accumulate
<b>ISO</b>	International Organization for Standardization
<b>IST-PDWZ</b>	Instituto Superior Técnico-Pixel Domain Wyner-Ziv
<b>IST-TDWZ</b>	Instituto Superior Técnico-Transform Domain Wyner-Ziv
<b>ITU-T</b>	International Telecommunications Union – Telecommunications standardization sector
<b>JPEG</b>	Joint Photographic Experts Group
<b>KLT</b>	Karhunen-Loève Transform
<b>LAPP</b>	Logarithm of the <i>A Posteriori</i> Probability
<b>LDPC</b>	Low-Density Parity-Check code
<b>MAP</b>	Maximum <i>A Posteriori</i>
<b>ML</b>	Maximum Likelihood
<b>MPEG</b>	Motion Picture Experts Group
<b>MSE</b>	Mean Square Error
<b>PC</b>	Personal Computer
<b>PRISM</b>	Power-efficient, Robust, hIgh-compression, Syndrome-based Multimedia coding

<b>PSNR</b>	Peak Signal Noise Ratio
<b>QCIF</b>	Quarter Common Intermediate Format for images (144 lines by 176 columns)
<b>RCPT</b>	Rate Compatible Punctured Turbo code
<b>RD</b>	Rate-Distortion
<b>RSC</b>	Recursive Systematic Convolutional code
<b>R-S</b>	Reed-Solomon code
<b>SISO</b>	Soft-Input Soft-Output
<b>SNR</b>	Signal Noise Ratio
<b>SOVA</b>	Soft-Output Viterbi Algorithm
<b>VCEG</b>	Video Coding Experts Group
<b>VISNET</b>	Networked Audiovisual Media Technologies



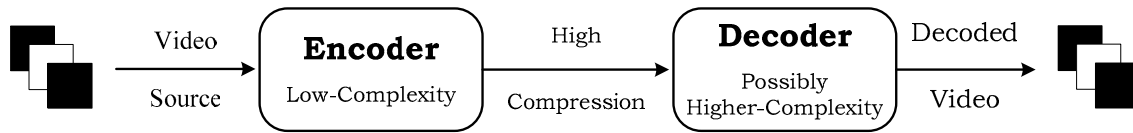
# Chapter 1

## Introduction

Today's digital video coding paradigm, represented by the standardization efforts of ITU-T VCEG and ISO/IEC MPEG, lies on hybrid Discrete Cosine Transform (DCT) and interframe predictive coding. In this coding framework, the encoder architecture is based on the popular hybrid motion compensation and DCT transform solution used to exploit the spatial and temporal redundancy existing in a video sequence. In order to explore those spatial and temporal correlations, the encoder requires a higher computational complexity, than the decoder (typically 5 to 10 times more complex [1]), mainly due to the motion estimation task; it is after all the encoder that has to take all coding decisions, and work in order to achieve the best performance, while the decoder remains a pure executer of the encoder "orders". This kind of architecture is well-suited for applications where the video is encoded once and decoded many times, i.e. one-to-many topologies, such as broadcasting or video-on-demand, where the cost of the decoder is more critical than the cost of the encoder.

In recent years, with emerging applications such as wireless low-power surveillance and multimedia sensor networks, wireless PC cameras and mobile camera phones, the traditional video coding architecture is being challenged. These applications have very different requirements than those of traditional video delivery systems. For some applications, it is essential a low power consumption both at the encoder and decoder sides, e.g. in mobile camera phones. In other types of applications, notably when there is a high number of encoders and only one decoder, e.g. surveillance, low cost encoder devices are necessary. In order to fulfil these requirements, it is essential to have a low-power and low-complexity encoder device, possibly at the expense of a higher complexity decoder. Figure 1.1 illustrates such scenario.

In this scenario, another important goal is to achieve a coding efficiency similar to that of traditional video coding schemes (nowadays represented by the ITU-T H.264/MPEG-4 AVC standard [2]), i.e. the shift of complexity from the encoder to the decoder should ideally not compromise the coding efficiency.



*Figure 1.1 – Ideal coding configuration for some emerging video applications.*

In conclusion, a challenging problem emerges with this new type of visual communication systems: How to achieve efficient and low-complexity encoder video compression, notably when traditional video coding does not provide an acceptable solution ?

Several results from Information Theory suggest that this problem can be solved by exploiting source statistics, partially or totally, at the decoder. These results can be used in the design of a new type of coding algorithms, the so-called Distributed Video Coding (DVC) solutions, presented in the following chapters.

## **1.1 Information Theory Background**

Distributed Source Coding (DSC) is a new compression paradigm relying on the coding of two or more dependent random sequences in an independent way, i.e. associating a separated, independent encoder to each of them; in this context, the term “distributed” refers to the encoding operation mode and not to its location. An independent bitstream is sent from each encoder to a single decoder which performs a joint decoding of all the received bitstreams exploiting the statistical dependencies between them. Based on the DSC independent encoding-joint decoding configuration, a new video coding paradigm, called distributed video coding (DVC), emerged. Figure 1.2 illustrates a wireless surveillance scenario where multiple sensor nodes (cameras) are sensing the same scene from different positions. While the cameras do not share information with each other, their associated video sequences are typically correlated since neighbouring cameras sense partially overlapping areas. Hence, each camera sends an independent bitstream to a centralised decoder, which performs joint decoding of all the received bitstreams exploiting the correlation between them. Therefore, it is possible to reduce the complexity of the encoding process by exploiting the correlation between the multiple encoded sequences just at the decoder. Since the power consumption is a key issue in a wireless sensor network [3], like the wireless surveillance scenario, the traditional video coding scheme is not well-suited for such video networks; with a traditional video coding scheme, the motion estimation task performed at the encoder to exploit temporal correlation corresponds to a high computational burden.

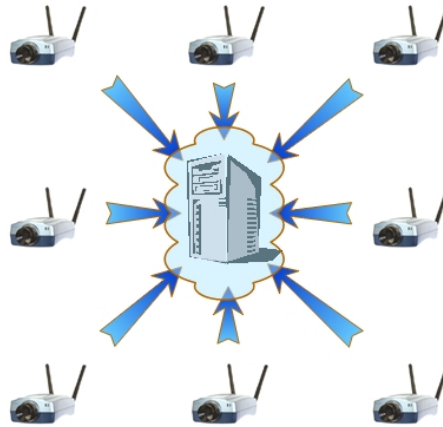


Figure 1.2 – Illustration of distributed source coding with multiple dependent video sequences.

Before presenting some theoretical Information Theory results relevant for distributed coding, a brief description of the traditional video coding paradigm is made from the Information Theory point of view.

With the traditional video coding schemes, the goal is to give response to questions such as: What is the minimum encoding rate,  $R$ , required such that two statistically dependent sequences  $X$  and  $Y$ , for instance the consecutive frames of a video sequence, can be perfectly recovered, i.e. without errors, by a joint decoder? Figure 1.3 depicts such question. As expected, the answer to this question derived from Information Theory is the joint entropy  $R = H(X, Y)$ .

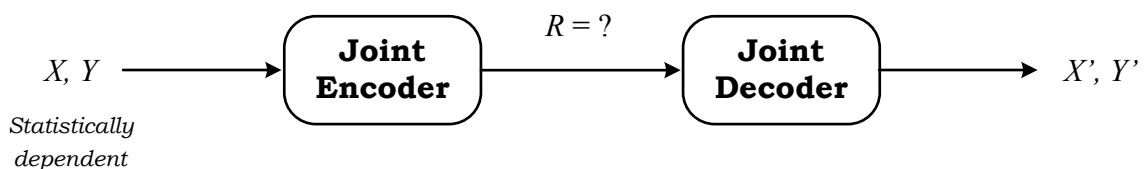


Figure 1.3 – Traditional coding paradigm.

When the statistically dependent sequences,  $X$  and  $Y$ , are independently encoded and decoded, the outcome for the corresponding query is also well-known:  $R_X \geq H(X)$  and  $R_Y \geq H(Y)$ . In this situation, the minimum number of bits per source symbol necessary to encode  $X$  and  $Y$  is given by the entropy of each source,  $H(X)$  and  $H(Y)$ , respectively. The total transmission rate  $R$  associated to the independent encoding and decoding of  $X$  and  $Y$  is given by  $R = R_X + R_Y \geq H(X, Y)$ ; an error-free reconstruction, in terms of coding errors, of the sequences  $X$  and  $Y$  is therefore guaranteed. Notice that it is always assumed that the transmission channel is error-free; when an error-prone channel will be considered, this will be explicitly mentioned.

### 1.1.1 Slepian-Wolf Theorem

As was seen in Section 1.1, the separate encoding and decoding of two sequences,  $X$  and  $Y$  with rates  $R_X \geq H(X)$  and  $R_Y \geq H(Y)$ , respectively, enables an error-free reconstruction at the decoder. Again notice that this error-free reconstruction refers to coding errors so this refers to

the so-called lossless coding. Consider now the situation described in Figure 1.4 where  $X$  and  $Y$  are two statistically dependent sequences separately encoded but the decoding process is performed jointly for the two sequences (distributed source coding).

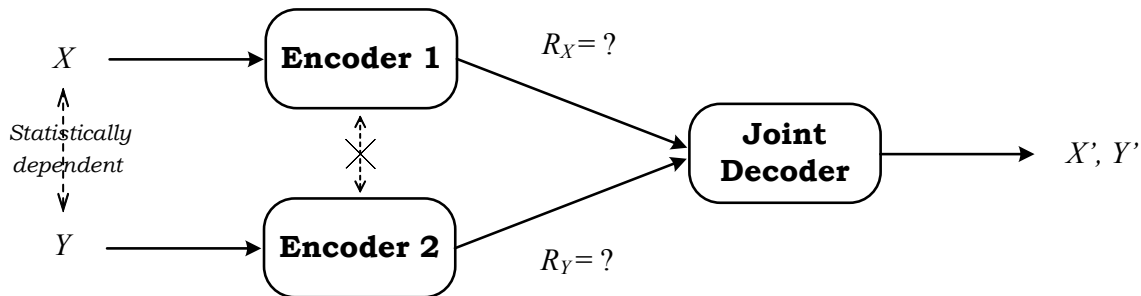


Figure 1.4 – Distributed compression of two statistically dependent discrete random sequences,  $X$  and  $Y$ , independently and identically distributed (i.i.d.).

It seems natural that using the same total bitrate than that of separate encoding and decoding situation,  $R \geq H(X) + H(Y) > H(X, Y)$ , the  $X$  and the  $Y$  sequences can be accurately reconstructed at the decoder since they are correlated and the total bitrate  $R$  is higher than the joint entropy  $H(X, Y)$ . But the real question is if it would be possible to fully recover these dependent sequences, with an arbitrarily small reconstruction error probability, using lower bitrates than individual entropies to encode them.

In the 1970s, Slepian and Wolf studied this problem [4], providing the first study about distributed source coding. Considering again the situation in Figure 1.4, let  $X$  and  $Y$  be two statistically dependent discrete random sequences, independently and identically distributed (i.i.d.). These sequences are separately encoded with rates  $R_X$  and  $R_Y$ , respectively, but are jointly decoded, exploiting the correlation between them. The possible rate combinations of  $R_X$  and  $R_Y$  for a reconstruction of  $X$  and  $Y$  with an arbitrarily small error probability, were determined by Slepian-Wolf [4]. These possible rate combinations are expressed by (1.1), (1.2) and (1.3).

$$R_X \geq H(X|Y) \tag{1.1}$$

$$R_Y \geq H(Y|X) \tag{1.2}$$

$$R_X + R_Y \geq H(X, Y) \tag{1.3}$$

where  $H(X|Y)$  is the conditional entropy of  $X$  given  $Y$  and  $H(Y|X)$  is the conditional entropy of  $Y$  given  $X$ . Equation (1.3) shows that even when the encoding of correlated sources is performed independently, a total bitrate,  $R = R_X + R_Y$ , equal to the joint entropy is enough. So, separate encoding in distributed video coding schemes does not (theoretically) need to have any compression efficiency loss when compared to the joint encoding used in the traditional video coding paradigm. This is exactly what the Slepian-Wolf theorem states [4].

Figure 1.5 illustrates the achievable rate region for which the distributed compression of two statistically dependent i.i.d. sources,  $X$  and  $Y$ , allows recovery with an arbitrarily small error probability according to the Slepian-Wolf theorem. In Figure 1.5, the vertical, horizontal and diagonal red lines, corresponding to (1.1), (1.2) and (1.3), respectively, represent the lower bounds for the achievable rate combinations of  $R_X$  and  $R_Y$ .

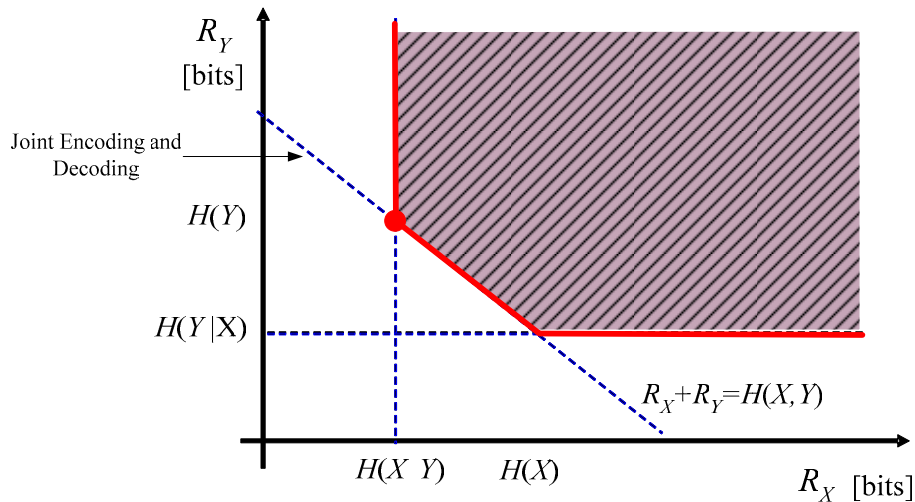


Figure 1.5 – Achievable rate region following the Slepian-Wolf theorem [4].

Slepian-Wolf coding is the term generally used in the literature to characterize coding architectures that follow the scenario described in Figure 1.4. Slepian-Wolf coding is also referred in the literature as lossless distributed source coding since it considers that the two statistically dependent sequences independently encoded are reconstructed with an arbitrarily small error probability at a joint decoder (approaching the lossless case). Notice that in this context, lossless is different from mathematically lossless since a controlled amount of error is allowed.

One interesting feature of Slepian-Wolf coding is the relationship that it has with channel coding. This relationship, already studied in the 1970s by Wyner [5], will be established here through the interpretation of Figure 1.6 taking two different coding points of view.

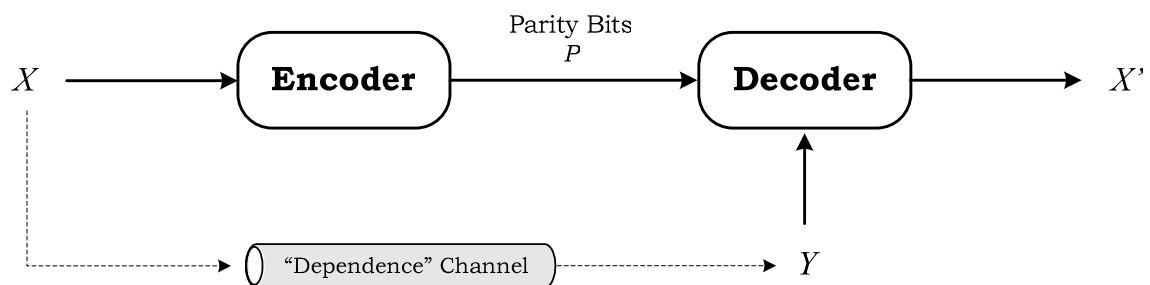


Figure 1.6 – Relationship between channel coding and Slepian-Wolf coding.

- 1) Consider a binary sequence,  $X$ , to be encoded and a channel-noisy version of it,  $Y$ , present at the decoder. To correct the errors between these two binary sequences, a channel code may be applied to the sequence  $X$ . Hence, jointly with  $Y$  the decoder uses the parity bits produced by the encoder to make error correction, achieving a perfect decoding of the sequence  $X$ . This is, in a few words, what the concept of channel coding provides.
- 2) Consider now Figure 1.6 from the Slepian-Wolf coding viewpoint. Since  $X$  and  $Y$  are two statistically dependent sequences, a virtual “dependence channel” can be considered between the  $X$  sequence (the virtual channel input) and the  $Y$  sequence (the virtual channel output). The  $Y$  sequence is therefore a “noisy” or an “errored” version of the  $X$  sequence where the “noise” introduced by the “dependence channel” is associated to the correlation between the sequences. The “dependence channel” concept gives an incentive to apply channel coding techniques since the  $Y$  sequence is a virtual channel-noisy version of the  $X$  sequence, as in 1). Thus, the “errors” between the  $X$  and  $Y$  sequences can be corrected applying a channel code to the  $X$  sequence. The additional bits sent by the encoder together with  $Y$  should provide a perfect reconstruction of the  $X$  sequence at the decoder.

### 1.1.2 Wyner-Ziv Theorem

In 1976, A. Wyner and J. Ziv [6] have studied a particular case of Slepian-Wolf coding corresponding to the rate point  $(H(X|Y), H(Y))$  identified in Figure 1.5 by the red dot. This particular case deals with the source coding of the  $X$  sequence considering the  $Y$  sequence, known as side information, is available at the decoder. Figure 1.7 illustrates such scenario; in the literature, this case is known as lossy compression with decoder side information. The designation of lossy compression is due to Wyner and Ziv having considered an average, acceptable distortion  $d$ , between the sequence to be encoded,  $X$ , and its decoded version,  $X'$ .

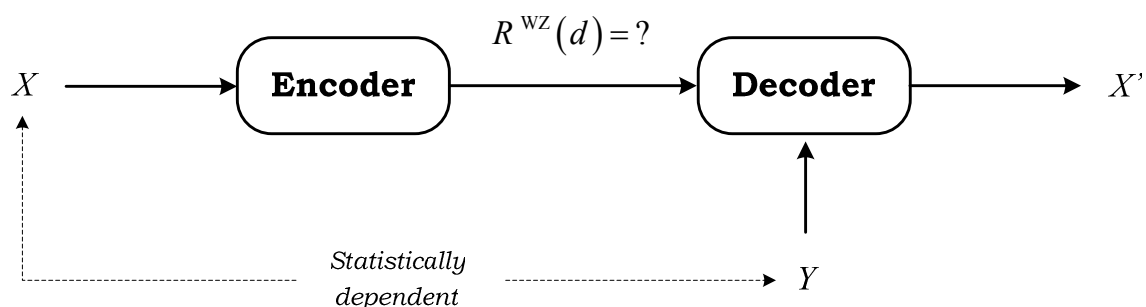


Figure 1.7 – Lossy compression with decoder side information.

Wyner-Ziv coding is another designation used in the literature to characterize the situation depicted in Figure 1.7 this means lossy compression with decoder side information. From now on, this will be the distributed coding situation that will be studied since several realistic scenarios, e.g. multi-camera systems (surveillance scenario) and video coding, may be rather

well characterized by the diagram in Figure 1.7. Typically, in a multi-camera system, a scene is simultaneously observed by multiple cameras at different angles. Each one of the multiple cameras transmits what it observes from the scene (corresponding to the sequence  $X$  in Figure 1.7), independently of the other cameras, to a single decoder at the central station. At each temporal instant, the camera at the central station may have available a rough or a “noisy” observation of the overall scene (corresponding to sequence  $Y$  in Figure 1.7) which also constitutes an input to the same decoder; the overall scene rough observation can be, for instance, generated from the previous temporal instant decoded scene. Since the cameras are not very far away, their observations will be correlated with the “noisy” observation available at the central station. Thus, each camera-central station connection is characterized by the situation depicted in Figure 1.7.

The Wyner-Ziv coding concept is also well-suited to the video coding scenario. In this case, some of the video sequence frames are Wyner-Ziv encoded (corresponding to the  $X$  sequence in Figure 1.7) while the remaining frames may be encoded using traditional video coding standards (like the MPEG-x or the H.26x standards), typically in intra coding mode. The decoder, making use of the traditionally coded frames, generates an estimate of the current frame to be Wyner-Ziv encoded (the side information) through frame interpolation or extrapolation techniques. This side information is then used in the decoding process of the current Wyner-Ziv encoded frame. In the following chapters, particular attention will be given to the Wyner-Ziv coding in the video coding scenario since this is the central subject of this Thesis.

Considering again the situation in Figure 1.7, let  $X$  and  $Y$  be two statistically dependent i.i.d. random sequences where  $X$  is the sequence to be encoded, the so-called main information, and  $Y$ , the so-called side information, is considered available at the decoder (for the moment, it is not that relevant how this information is made available to the decoder). Independently of the way  $Y$  is made available at the decoder, there is no exploitation of the statistical dependency between  $X$  and  $Y$  at the encoder. The Wyner and Ziv work establishes the minimum rate  $R^{\text{WZ}}(d)$  necessary to encode  $X$  guaranteeing its reconstruction with an average distortion below  $d$ , assuming that the decoder has the side information  $Y$  available. The results obtained by Wyner and Ziv indicate that when the statistical dependency between  $X$  and  $Y$  is exploited only at the decoder, the transmission rate increases comparing to the case where the correlation is exploited both at the encoder and the decoder, for the same average distortion,  $d$ . This is precisely what the Wyner-Ziv theorem states [6]. Mathematically, the Wyner and Ziv theorem can be described by

$$R^{\text{WZ}}(d) \geq R_{X|Y}(d), \quad d \geq 0 \quad (1.4)$$

where  $R^{\text{WZ}}(d)$  represents the Wyner-Ziv minimum encoding rate (for  $X$ ) and  $R_{X|Y}(d)$  represents the minimum rate necessary to encode  $X$  when  $Y$  is simultaneously available at the encoder and decoder (always for the same average distortion  $d$ ). In the literature,  $R^{\text{WZ}}(d)$  and

$R_{X|Y}(d)$  are called Rate-Distortion (RD) functions. Notice, however, that when  $d = 0$ , i.e. when no distortion exists, (1.4) falls back to the Slepian-Wolf result, i.e.  $R^{WZ}(0) = R_{X|Y}(0)$ . This means that it is possible (theoretically) to reconstruct the sequence  $X$  with an arbitrarily small error probability even when the correlation between  $X$  and the side information is only exploited at the decoder.

Further, Zamir [7] has demonstrated that the Wyner-Ziv coding rate corresponds to an increase smaller than 0.5 bit/source symbol regarding the rate in a joint encoding and decoding situation (encode of the sequence  $X$  exploiting the correlation with  $Y$  both at the encoder and decoder). This result, mathematically expressed by

$$R_{X|Y}(d) + 0.5 \text{ bit} \geq R^{WZ}(d), \quad (1.5)$$

was obtained for general statistics using a Mean Square Error (MSE) to measure the reconstruction error at the decoder.

Wyner and Ziv showed, however, that there is no rate increase, for all  $d > 0$ , when  $X$  and  $Y$  are jointly Gaussian sequences and a MSE distortion measure is considered [6]. Hence, the equality case in (1.4) holds for  $X$  and  $Y$  jointly Gaussian; this means that there is theoretically no reduction in the transmission rate when the exploitation of the statistical dependency between  $X$  and  $Y$  is performed both at the encoder and the decoder comparing to the situation described by Figure 1.7.

In conclusion, the Slepian-Wolf and the Wyner-Ziv theorems for long well-known in Information Theory suggest that it is possible to compress two statistically dependent signals in a distributed way (separate encoding, jointly decoding) using a rate similar to that used in a system where the signals are encoded and decoded together, i.e. like in traditional video coding schemes; for the Wyner-Ziv theorem, this conclusion is only valid when  $X$  and  $Y$  are jointly Gaussian and a MSE distortion measure is considered. In traditional video coding schemes those signals can be, for instance, the odd and even frames of a video sequence.

## **1.2 Target Applications**

As was seen in previous sections, distributed video coding is a new coding paradigm characterized by a configuration where the encoder has low-complexity at the expense of a higher decoder complexity. This configuration makes DVC a promising coding solution for some emerging applications, where the encoder complexity or the power consumption are scarce resources. Since the DVC paradigm enables to exploit statistical dependencies (for instance, between two temporally adjacent video frames) at the decoder only, there is no need for a prediction loop at the encoder side; thus interframe error propagation (typical in the traditional video coding scenario) may be avoided in the DVC paradigm. Some applications to which DVC is a promising coding solution are described in the following:



### Wireless Low-Power Surveillance Networks

Nowadays, event sensing is almost present everywhere. One example of event sensing is the video surveillance scenario, illustrated in Figure 1.8, where multiple cameras are sensing the same event from different locations.

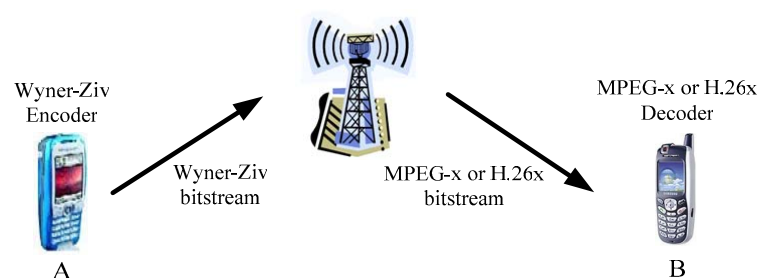


*Figure 1.8 – Video surveillance scenario.*

Neighbouring cameras typically sense partially overlapping areas and therefore their associated video sequences are correlated. Since the number of encoders is usually much higher than the number of existing decoders (typically one), it is possible to reduce the cost of the system if low-complexity encoders are used and if only one or a few more complex decoders are necessary. Wyner-Ziv coding is well-suited for this scenario, since it enables to explore the correlation between the multiple encoded sequences just at the decoder, providing a low encoding complexity. Using Wyner-Ziv coding, the interframe error propagation is also avoided in a video surveillance scenario (where severe errors can occur due to the unpredictable behaviour of the channel) since the correlation is exploited at the decoder and therefore no prediction loop exists at the encoder.

### Wireless Mobile Video

Another application that can benefit from the DVC paradigm is wireless video, e.g. wireless video communication between a pair of camera phones as illustrated in Figure 1.9.



*Figure 1.9 – Wireless mobile video scenario.*

The major requirement in this application is to have a low-complexity encoder and decoder in each terminal, since power consumption and battery life is closely related to the complexity of the encoder/decoder pair. However, to take advantage of Wyner-Ziv coding in a wireless mobile video scenario, it is necessary to have a high-complexity decoder device, as was previously seen. For this application, the high-complexity decoder is located at a base station together with a transcoder. This base station will be responsible for receiving the low-complexity encoded bitstream (known as Wyner-Ziv bitstream), transcode it to a MPEG-x or a H.26x bitstream and transmit it to another terminal with a low complexity decoder (terminal B in Figure 1.9). This will enable to have a low-complexity encoder and decoder at each terminal. In a wireless mobile video scenario, the source statistics are exploited at the base station and therefore the Wyner-Ziv encoder does not need of a prediction loop (typical in traditional video coding architectures); since no prediction loop exists in the Wyner-Ziv encoder, the interframe error propagation is avoided in the uplink situation (Wyner-Ziv encoder – base station link). In the downlink scenario, the interframe error propagation is reduced by exploiting the error resilience tools available in the MPEG-x or H.26x standards.

### **Multi-View Acquisition**

In several applications, a scene or an object is acquired by many cameras located at fixed spatial positions, e.g. for special movie effects, image-based rendering (3D reconstruction with texture mapping). One of the research challenges is to find the best algorithms and sensors to acquire the video sequences especially because a large number of cameras are necessary to fulfil the requirements, e.g. to have photo-realistic scenes or immersive 3D models. Figure 1.10 illustrates a large camera array scenario to acquire these scenes.



*Figure 1.10 – Large camera array scenario.*

In such scenario, neighbouring cameras of a large camera array capture overlapped views and therefore views that are correlated. With Wyner-Ziv coding, independent encoding of each view can be performed in each camera while a central station has to perform joint decoding, in order to exploit the correlation between views. This will enable to have low-complexity encoders, and thus to use low-cost cameras, minimizing the total cost of the camera array.

## **Video-Based Sensor Networks**

Sensor networks are becoming a new field of research driven by advances in microelectronics and communications networks. The main goal is the development of technologies that would enable to employ thousand of sensors in a chosen environment to accomplish a certain task. If these sensors have video acquisition capabilities, several applications are possible such as tracking of persons throughout an environment, monitoring of activities, tracking events and creating alarms, if necessary. Also by having a large number of sensors, multiple camera angles are available, making some computer vision tasks (e.g. gesture recognition) much easier than using a single view. Wyner-Ziv coding can help the construction of such video-based sensor networks, since it allows the construction of low-complexity, low-cost and low power consumption encoder devices. In this type of networks, the decoder is a central processing unit with high computational capabilities responsible to collect and process all the information received (namely to explore the redundancy between all the received video signals).

### **1.3 Main Objectives of the Thesis**

The main objective of this Thesis is to study, develop and evaluate new, more efficient algorithms for distributed video coding, thus reducing the gap in performance when compared to the traditional video coding systems.

Practical efforts towards distributed video coding solutions are, nowadays, just starting and the technology is not yet sufficiently mature. The available state-of-the-art results, in terms of rate-distortion performance, are promising; however it is essential to improve and to create tools for the DVC scenario with the purpose of achieving better rate-distortion performances than the ones available today in the literature. In this context, the major goals are:

1. Review and analyze the most relevant distributed video coding solutions available in the literature.
2. Implement and develop state-of-the-art distributed video coding schemes for the pixel and transform domains. While the pixel domain codec is simpler (in terms of complexity), the transform domain one provides better performance (at the cost of higher encoding complexity). Both codecs will be used as a starting point (or as a testbed) to implement new techniques that improve the overall rate-distortion performance, notably:
  - **Improved side information generation:** In low-complexity Wyner-Ziv video coding, motion estimation and compensation are now performed only at the decoder (the encoder is relieved of this task) by using frame interpolation schemes. The goal is to achieve better coding efficiency in terms of rate-distortion, by generating at the decoder a better estimate of the side information from temporally adjacent frames using efficient frame interpolation tools.

- **Improved channel coding:** In DVC schemes, channel codes are used to perform source coding, together with other coding tools. It is therefore essential to optimize those channel codes, in terms of coding efficiency, to work well for source coding with decoder side information. Another goal is to model the virtual channel statistics in order to accurately estimate the error distribution between the side information and the original frame.

Taking into account the goals stated above, this Thesis includes several contributions to the distributed video coding field with a particular emphasis on the frame interpolation tools employed at the decoder. It should be noticed that the pixel and transform domain codecs were fully developed by the author of this Thesis.

## 1.4 Outline of the Thesis

This Thesis describes in detail the development of two Wyner-Ziv video coding solutions bringing better performance than current state-of-the-art: one for the time domain (pixel based) and other for the transform domain (DCT coefficient based).

- ◆ The context and motivation for this work is presented in Chapter 1 together with the definition of the main objectives and the outline of the Thesis.
- ◆ In Chapter 2, a review of the most relevant distributed video coding schemes is presented. The work of the two research groups who have been responsible for the development of the most relevant distributed source video coding systems nowadays available: Bernd Girod's group at Stanford (University of Stanford) and Kannan Ramchandran's group at Berkeley (University of California) is described.
- ◆ In Chapter 3, the Instituto Superior Técnico-Pixel Domain Wyner-Ziv video codec, referred as IST-PDWZ, is presented highlighting two major important modules: the turbo codec and the frame interpolation tools. As is well-known, turbo codes are a powerful channel code [8]. In a digital communication system, the channel codec plays an important role: to make the communication system less vulnerable to transmission errors by adding redundant information; in this context, the turbo code mathematical formalism is typically described considering Additive White Gaussian Noise (AWGN). In the IST-PDWZ solution, the turbo codec is used in a source coding context where the decoder has available an estimate (a "noisy" version) of the original frame (the side information). In this solution context, the noise distribution is twofold:
  - The noise distribution associated to the transmission channel.
  - The noise (error) distribution between the side information and the original frame; this noise is often called virtual channel noise since it is associated with the virtual dependence channel (see Section 1.1.1).

In this Chapter, the author of this Thesis, adjust the turbo code mathematical formalism (considered in a channel coding situation) to the source coding scenario mentioned above. Due to the lack of details in the state-of-the-art solution about the turbo codec implementation, the modelling of the noise distributions is the major contribution in the IST-PDWZ solution regarding the turbo codec architectural module. Regarding the frame interpolation tools, a block-based framework including motion estimation, bidirectional motion estimation and a spatial smoothing algorithm is proposed in this Chapter. This frame interpolation framework aims at improving the RD performance of the IST-PDWZ codec regarding the state-of-the-art solution available in the literature and constitutes a major achievement of this Thesis.

- ◆ Chapter 4 describes the Instituto Superior Técnico-Transform Domain Wyner-Ziv video codec, referred as IST-TDWZ codec; this codec is an advanced version of the IST-PDWZ codec since it makes use of an integer  $4 \times 4$  DCT to achieve better rate-distortion performance. The IST-TDWZ solution has some differences regarding the similar, state-of-the-art solution available in the literature, particularly in the turbo codec, DCT, quantizer and frame interpolation modules; these differences are not only motivated by the lack of details regarding the state-of-the-art solution but also related to improvements explicitly introduced. The turbo codec implementation presented in Chapter 3 for the pixel domain is adapted to the transform domain (different magnitudes and signs are associated to the sample values of the pixel and transform domains). A  $4 \times 4$  block-based DCT as defined by the ITU-T H.264/MPEG-4 AVC video coding standard [2] is employed to decorrelate samples blocks. Some transform coefficients are quantized using a uniform scalar quantizer with a symmetric quantization interval around the zero amplitude in order to reduce the block artefacts effect. The IST-TDWZ codec employs the same frame interpolation tools used in the IST-PDWZ solution. In both Chapters 3 and 4, an extensive analysis of the rate-distortion performance of the proposed video codecs is made, showing results for a wide range of sequences and testing conditions.
- ◆ Finally, Chapter 5 summarizes the achievements of this Thesis and points out some directions for future work.

## 1.5 Publications

The work presented in this Thesis or somehow related to it resulted in one national conference publication

- C. Brites, F. Pereira; “Distributed Video Coding: Bringing New Applications to Life”, *5th Conference on Telecommunications - ConfTele*, Tomar, Portugal, April 2005.

five international conference publications

- J. Ascenso, C. Brites, F. Pereira, “Improving Frame Interpolation with Spatial Motion Smoothing for Pixel Domain Distributed Video Coding”, *5th EURASIP Conference on Speech and Image Processing, Multimedia Communications and Services*, Slovak Republic, July 2005.
- J. Ascenso, C. Brites, F. Pereira, “Motion Compensated Refinement for Low Complexity Pixel Based Distributed Video Coding”, *IEEE International Conference on Advanced Video and Signal Based Surveillance*, Como, Italy, September 2005.
- L. Natário, C. Brites, J. Ascenso, F. Pereira, “Extrapolating Side Information for Low-Delay Pixel-Domain Distributed Video Coding”, *International Workshop on Very Low Bitrate Video - VLBV*, Sardinia, Italy, September 2005.
- A. Trapanese, M. Tagliasacchi, S. Tubaro, J. Ascenso, C. Brites, F. Pereira, “Embedding a Block-based Intra Mode in Frame-based Pixel Domain Wyner-Ziv Video Coding”, *International Workshop on Very Low Bitrate Video - VLBV*, Sardinia, Italy, September 2005.
- A. Trapanese, M. Tagliasacchi, S. Tubaro, J. Ascenso, C. Brites, F. Pereira, “Improved Correlation Noise Statistics Modelling in Frame-based Pixel Domain Wyner-Ziv Video Coding”, *International Workshop on Very Low Bitrate Video - VLBV*, Sardinia, Italy, September 2005.

and one journal paper submission

- C. Brites, J. Ascenso, A. Trapanese, M. Tagliasacchi, F. Pereira, S. Tubaro, “Advances on Pixel Domain Wyner-Ziv Video Coding”, *IEEE Transactions on Image Processing* (submitted).

To be more precise, some of the research work included in the papers above is not described in this Thesis although that work is a direct consequence of the efforts developed in this context; in other words, the software developed in this Thesis served as starting point to quickly and collaboratively experiment new ideas and achieve better rate-distortion performance. The last two *VLBV* publications and the journal paper submission are the result of a joint collaboration with *Politecnico di Milano* (Italy) in the framework of the Network of Excellence VISNET (Networked Audiovisual Media Technologies). In this collaboration, the researchers got the software developed by the author of this Thesis and in a very short time it was possible to collaboratively try new ideas.

Besides this collaboration, the software developed in the context of this Thesis was chosen as the basic software framework for the European project DISCOVER (DISTRIBUTED CODING for Video sERVICES) starting in September 2005, and targeting the development of advanced distributed video coding tools in Europe.

## Chapter 2

# Wyner-Ziv Video Coding: a Review

Distributed Video Coding (DVC) is a new video coding paradigm which allows among other things shifting complexity from the encoder to the decoder. DVC theory relies on the coding of two or more dependent random sequences in an independent way, i.e. associating an independent encoder to each sequence. A single decoder is used to perform joint decoding of all encoded sequences, exploiting the statistical dependencies between them.

As was seen in Chapter 1, the Information Theory, through the Slepian-Wolf and the Wyner-Ziv theorems, suggests a lower bound for the transmission rate in a situation of separate encoding and joint decoding of two statistically dependent sequences. The Information Theory states that this rate transmission can be similar to the one used in traditional video coding schemes where a joint encoding and decoding paradigm is used.

From those theoretical results, a video coding challenge comes out: What coding system must be developed in order to approach the rate limits suggested by the Information Theory ? Thus, the goal is to design a video coding system using Distributed Source Coding (DSC) that is able to offer, or at least approaching, the compression performance of traditional video coding in terms of the transmission rate, this means  $H(X, Y)$ .

Moreover, what coding tools should be included in this new video coding system in order to simultaneously fulfil the constraints of low-complexity and high compression efficiency encoding ? In the traditional coding systems, tools like motion compensation, transform coding, quantization and entropy coding are the main building modules; from these modules, entropy coding is the only one that it is always truly (mathematically) lossless. In the new coding paradigm, the statistical dependencies between sequences cannot be exploited at the encoding process (see Chapter 1) if low encoder complexity is targeted; this suggests that the motion

compensation module cannot, in principle, be used at the encoder side. But what happens with the transform coding, quantization and entropy coding blocks ? Can they be useful in distributed video coding to approach the Rate-Distortion (RD) performance of traditional video coding ? The answer to this question is the aim of the following two sections.

It is worthwhile to remind that the target scenario is the lossy source coding of the main information with available receiver side information (Wyner-Ziv coding) as described in Chapter 1. Thus, during this Thesis, the following terminology will be used:

- ◆ **Main information** is the information that is to be Wyner-Ziv encoded (the  $X$  sequence).
- ◆ **Side information** is an estimate of the main information generated at the decoder; the side information helps the decoder in the decoding process of the main information (the  $Y$  sequence).

## 2.1 Overview on Distributed Coding

Theoretical foundations for distributed coding have been established in the 1970s, as mentioned in Chapter 1. However practical efforts with the aim of approaching the Slepian-Wolf and Wyner-Ziv bounds were started more recently with the emergence of applications where low encoding complexity is a major requirement.

In the Wyner-Ziv coding scenario described in Chapter 1, it was assumed that the side information  $Y$  is available at the decoder; in this context, available means that a reliable reconstruction of  $Y$  is accessible to the decoder: encoding  $Y$  with a rate  $R_Y \geq H(Y)$  provides such a reliable reconstruction. For example,  $Y$  may be provided to the decoder using a traditional coding solution such as the MPEG-x or H.26x standards. The dilemma would be then how to encode  $X$  this means what coding solutions are well-suited in order to reach optimal performance in terms of rate, reminding that the target here is  $H(X|Y)$ . The aim is thus to build a system capable of approaching the rate point  $(H(X|Y), H(Y))$ , corresponding to the distributed coding scenario in study.

To achieve the goal of approaching the rate point  $(H(X|Y), H(Y))$  adequate techniques to encode  $X$  have been considered. The first technique considered in the study of distributed source coding was quantization. A source data observation is encoded, in this case quantization-based encoded, and transmitted to a decoder; the decoder, having access to an uncoded source data observation correlated to the one encoded, attempts to obtain the original source data observation.

The relationship between Slepian-Wolf coding and channel coding, previously studied in [5], has then stimulated the usage of encoding techniques based on channel coding to encode the sequence  $X$ . Typically, channel coding is used to “protect” a previously source coded signal against channel transmission errors; in other words, channel coding aims to reduce the decoding error probability. In a nutshell, over a signal (data information) source encoded it may be applied a channel coding technique to generate redundant information relative to that source



encoded data information. This redundant information is added to the source encoded data information and both are transmitted to the decoder. In the case of the received data information being corrupted by channel errors, the decoder has therefore additional information that can explore to detect and correct those errors. Thus, it is attained a lower decoding error probability than that in a coding system where no channel coding is utilized and therefore better quality in the reconstructed signal is achieved. Applying this brief description of channel coding to the distributed coding context, it seems that it is sufficient to transmit redundant information about the sequence  $X$  to the decoder in order to obtain an  $X$  reconstruction with low error decoding probability since in a distributed coding scheme the decoder knows an estimate of the  $X$  sequence; that  $X$  estimate, named side information  $Y$ , corresponds to the corrupted signal received at the decoder using a traditional coding system. Several channel codes were therefore tested in the context of distributed coding such as turbo codes [9], Low-Density Parity-Check (LDPC) codes [10] as well as syndromes [11]. The test results show that channel coding performed after quantization allows approaching the theoretical bounds, i.e. the Slepian-Wolf and the Wyner-Ziv limits.

In a non-distributed source coding scenario, transform coding is another source coding technique used to reduce the transmission rate. Generically, transform coding is applied over  $n \times n$  sample blocks of an image, decorrelating those samples. Typically, most of the image energy is enclosed in a small number of the resulting  $n \times n$  transform coefficients; since this small number of coefficients contains most of the energy, they present larger values compared to the remaining transform coefficient values. Thus, only the larger value transform coefficients need to be encoded, reducing therefore the transmission bitrate; the remaining coefficients may be “deleted” without this operation being perceptible in the reproduced image quality. Relying on this idea, the rate-distortion performance of distributed coding schemes that perform transform coding followed by quantization and channel coding has been determined [12]; the Discrete Cosine Transform (DCT) was one of the transforms evaluated. Generally, the results obtained corroborate the idea that performing transform coding before the quantization and the channel coding stages allows a reduction of the  $X$  sequence transmission rate [12]. Notice that the major part of the approaches that have been developed considered the Wyner-Ziv coding scenario.

Just as examples, some distributed coding approaches for image and video coding available in the literature will be here briefly described. Note that those approaches may comprise all or only some of the tools mentioned above (quantization, transform coding and channel coding) in accordance with the purpose of each approach.

### **Image Coding**

Pradhan and Ramchandran used in 2001 cosets (sets of source codewords resulting from the partitioning of the source codeword space) to improve the quality of a noisy analogue image transmission [13]; each coset has associated an order index called syndrome. To the image transmitted without channel coding over an analogue channel,  $Y$ , an encoded version,  $X$ , of the same image sent over a digital channel is added. The digital bitstream, corresponding to the

syndromes, is decoded using the noisy analogue version of the same image previously converted to a digital version,  $Y'$ , as side information. The result is an improved image reconstruction since portions of  $Y'$  where there are errors introduced by the analogue transmission channel are replaced by the corresponding portions resulting from the syndrome decoding associated to  $X$ .

In 2002, Liveris, Xiong and Georgiades applied turbo codes to encode images that exhibit nearly Gaussian correlation between co-located pixel values [14]. Each pixel value (for instance, grey level value) is firstly encoded using cosets. The resulting coset symbols (syndromes) are then encoded using turbo codes achieving more compression than using cosets only.

### **Video Coding**

In 2002, Jagmohan, Sehgal and Ahuja [15] made use of coset codes for predictive encoding in order to reduce the consequences of the predictive mismatch without a large increase in terms of bitrate. Predictive mismatch denotes here an erroneous prediction symbol reconstruction at the decoder due to differences between the decoded and encoder prediction symbols.

In the same year, an approach well-known as PRISM (Power-efficient, Robust, hIgh-compression, Syndrome-based Multimedia coding) was proposed by Puri and Ramchandran [16] for multimedia transmissions on wireless networks using syndromes. The major goal of this approach is to join the traditional intraframe coding error robustness with the traditional interframe compression efficiency.

In 2002, making use of turbo codes, Aaron, Zhang and Girod [17] have shown results on video coding using an intraframe encoding-interframe decoding scheme where individual frames are independently encoded but are jointly decoded.

In 2003, Zhu, Aaron and Girod have proposed an approach to Wyner-Ziv based low-complexity coding under the name of “distributed compression for large cameras arrays” [18]. In this solution, multiple correlated views of a scene are independently encoded with a pixel domain Wyner-Ziv coder but are jointly decoded at a central node. Zhu *et al.* performed in [18] a comparison between pixel domain Wyner-Ziv coder and an independent encoding and decoding of each view employing the JPEG-2000 wavelet image coding standard. The results demonstrate that at lower bitrates the solution presented by Zhu *et al.* achieves higher PSNR than JPEG-2000 with a lower encoder complexity. For more details, the reader should consult [18].

In 2004, Aaron, Rane, Setton and Girod [12] proposed an architecture similar to the one in [17]; the key difference regarding [17] is the additional use of transform coding (DCT transform) at the encoder. The results obtained show that the new coding solution leads to a better coding efficiency when compared with the solution in [17] (at the cost of a high encoder complexity associated with the DCT transform).

In the same year, the most recent Wyner-Ziv low-complexity video coding solution by Aaron, Rane and Girod was proposed in [19]. This solution is based on an intraframe encoding-interframe decoding system and beside the bitstream resulting from the current frame encoding process the encoder also transmits supplementary information about the current frame to help the decoder in the motion estimation task.

In 2004, Rane, Aaron and Girod have presented another approach [20] targeting the increase of the video transmission. Specifically, the aim of this approach is to make a traditionally encoded bitstream becoming more error resilient when it is transmitted over an error-prone channel with no protection against channel transmission errors, for example by means of channel coding.

The application of distributed source coding to the video coding area will deserve special attention during the current and the remaining sections of this Chapter since improved distributed video coding is the aim to attain in this Thesis. The most relevant, available schemes in the area of distributed video coding will be presented with detail in Section 2.2.

### 2.1.1 Basic Wyner-Ziv Coding Architecture

In the beginning of this Chapter, the following question was raised: What DVC scheme must be developed to approach the limits suggested by the Information Theory? The first step to answer this question was made in Section 2.1 with the presentation of a brief overview on the evolution of the encoding techniques employed to encode the sequence  $X$ . In this overview, three main tools to encode  $X$  in a distributed way with receiver side information can be identified: transform coding, quantization and channel coding. From the several solutions present in Section 2.1, it seems that the most consensual Wyner-Ziv video coding architecture comprises three modules combined as in the basic Wyner-Ziv architecture illustrated in Figure 2.1. This architecture is the result of the evolution of DSC schemes along the past few years.

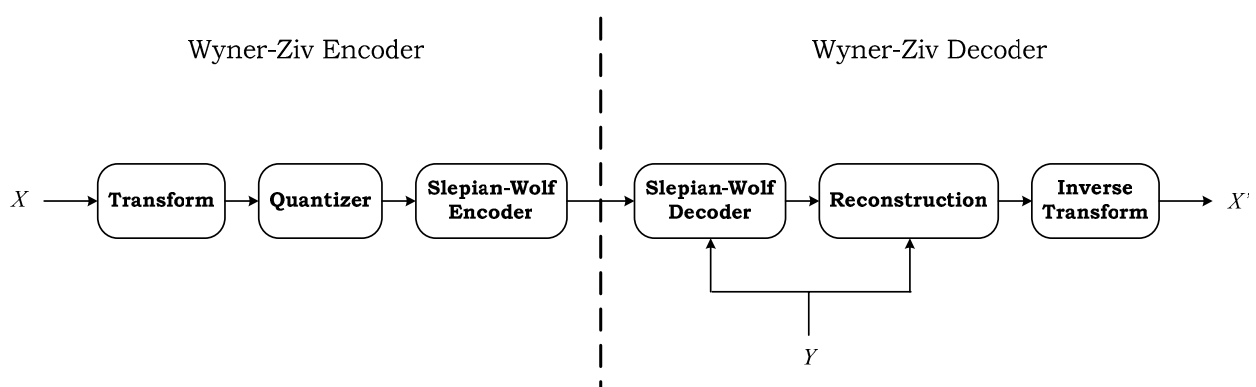


Figure 2.1 – Block diagram of the basic Wyner-Ziv codec.

In a nutshell, the coding procedure illustrated in Figure 2.1 is described as follows:

- The samples of the sequence  $X$  are first transformed from the spatial domain to another domain, for instance the frequency domain, in order to allow a more compact representation of the sequence  $X$  and thus a lower bitrate.

- The corresponding transform coefficients are then quantized, generating the quantized symbol stream, in order to exploit the limitations of the Human Visual System (HVS).
- The basic Wyner-Ziv encoding process ends with a Slepian-Wolf encoding module. As was already mentioned, the Wyner-Ziv decoder has available an estimate or an “errored” version of the  $X$  sequence, represented in Figure 2.1 by  $Y$ ; if redundant information about the sequence  $X$  is transmitted to the Wyner-Ziv decoder, the errors in the  $Y$  sequence may be corrected (as explained in Chapter 1). Thus, the Slepian-Wolf encoder architectural module in Figure 2.1 plays the role of redundant information generator; in other words, making use of channel codes, the Slepian-Wolf encoder produces redundant information (e.g. parity bits) from the quantized symbol stream. The bitstream produced by the Wyner-Ziv encoder results therefore from source coding (transform coding and quantization) followed by Slepian-Wolf coding.
- At the basic Wyner-Ziv decoder, the quantized symbol stream is decoded through joint source-channel decoding with the aid of the sequence  $Y$ , the side information.
- The decoded quantized symbol stream and the side information  $Y$  are then used together in a reconstruction module to estimate the transform coefficients. To reconstruct the  $X$  sequence,  $X'$ , an inverse transform operation, the dual operation of that performed at the encoder, is finally performed.
- The sequence  $Y$  (typically a decoder estimate of the sequence to be Wyner-Ziv encoded,  $X$ ) is considered to be available at the decoder.

Each module of the basic Wyner-Ziv architecture will be presented with more detail in following sections.

### **2.1.2 Transforming in the Basic Wyner-Ziv Codec**

The first stage to encode the main information in the proposed basic architecture is the transform coding. Typically, transform coding is applied over  $n \times n$  sample blocks of a frame to decorrelate those frame samples. The result of that decorrelation process is the block energy concentration in a few large valued transform coefficients; these coefficients are called low-frequency transform coefficients because they typically represent lower frequencies (closer to DC). Since the low-frequency coefficients contain most of the block information and have the largest values, it is possible to reduce the number of bits required to encode  $X$  by transmitting only those coefficients to the decoder (without this shortcut being perceptible in the reproduced frame quality due to the HVS limitations). A brief overview on the usage of transform coding in the distributed coding context will be now presented.

In 2001, Pradhan and Ramchandran in [13] have used wavelets-based image coding in a Wyner-Ziv coding scenario; Wyner-Ziv coding is a particular case of distributed coding, where only one source is considered. The quality of a noisy analogue image transmission is improved by using the analogue image to help in the decoding procedure of a digital version of the same image.

In 2003, Rebollo-Monedero *et al.* [21] have determined for the coding scenario illustrated in Figure 2.1 that the discrete cosine transform (DCT) is an asymptotically optimal choice for the transform module, in terms of rate-distortion performance. In [21] it is also evaluated the use of the DCT for Wyner-Ziv video coding. The results obtained confirm the main idea that motivated the study of transform coding in distributed coding: exploiting the spatial correlation within a frame at the encoder leads to a rate-distortion improvement. This improvement is measured over a non-transforming scheme where just quantization and Slepian-Wolf coding is used to encode the sequence. A detailed description of the scheme suggested in [21] can be found in [12].

### 2.1.3 Quantizing in the Basic Wyner-Ziv Codec

The quantization is a source coding technique used in traditional coding to compress a range of values into a single value. That is, the domain of a signal is divided into intervals also called in the literature bins. To each interval or bin is then associated a value often called quantized symbol or codeword. Since the number of bins is lower than the total number of the values that a signal can assume, the total bitrate is reduced. With the purpose of reducing the bitrate, the quantization stage was also applied to distributed coding. But how similar will the quantizer for distributed coding be to the one used in traditional coding? A brief overview about the study of quantization in the distributed coding context will be presented in the following.

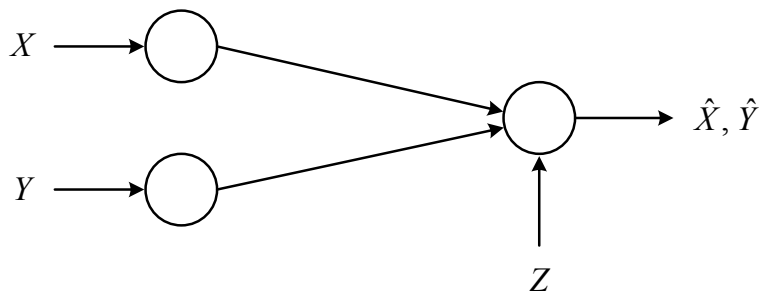
In 1987, Flynn and Gray [22] considered the quantization problem in a distributed sensing system constituted by two separated sensors observing a common target. One of the sensors, called remote sensor, encodes its observation with an encoding scheme purely based on quantization and transmits the output of the encoding process to the other sensor, called local sensor. The local sensor combines its observation with the encoded version of the remote sensor in a joint decoding process producing the best possible estimate of the remote sensor observation. In this context, different quantization algorithms were tested in order to achieve a good overall performance.

In 1998, Shamai, Verdú and Zamir presented quantizers designed for Bernoulli sources and jointly Gaussian sources in a lossy compression with receiver side information scenario [23]. They considered that the encoder comprises a codebook implemented with an entropy-coded randomized (dithered) quantizer and a Slepian-Wolf encoder to encode the source information. For a given codebook, the decoder uses the side information to reduce the rate required to encode the source information providing simultaneously a source reconstruction at the distortion associated to the given codebook. The reconstruction error at the decoder was measured using the Hamming metric and the Mean Square Error (MSE) to the Bernoulli and the Gaussian sources, respectively.

In 2000, Servetto [24] has contributed to the construction of a family of quantizers in order to attain the target rate  $H(X|Y)$ . Servetto's study was performed for jointly Gaussian sources using the MSE to measure the reconstruction error at the decoder. In [24], the analytical and the

empirical rate-distortion performance at high rates are compared, showing that the empirical results obtained are coherent with the analytical one (the Wyner-Ziv rate-distortion function).

In 2001, Fleming and Effros [25] considered vector quantizers in the study of rate-distortion optimized vector quantizers for network source coding with fixed rate. They used the generalized Lloyd algorithm to optimize each encoder and decoder of a three-node network by finding the codewords that minimize the distortion relatively to a node sample value; the three-node network considered in [25] for rate-distortion performance evaluation is depicted in Figure 2.2. The results obtained show significant coding gains over an independent source coding scenario when the sources are correlated.



*Figure 2.2 – Three-node network considered in [25].*

In 2002, Muresan and Effros [26] have demonstrated the relationship between scalar quantization and histogram segmentation in order to develop an algorithm for optimal quantizer design. Muresan and Effros considered in their algorithm discrete alphabet sources and alphabet symbols quantization with contiguous bins. The same authors mention in [27] that, for some sources, a contiguous interval quantization may lead to worse rate-distortion performance comparatively to that was obtained with disjoint intervals quantization.

In 2003, Rebollo-Monedero, Zhang and Girod [28] studied optimal quantizers design considering the Wyner-Ziv coding scenario. The results were obtained for Gaussian sources and disjoint intervals quantization. The authors show that the rate-distortion performance obtained in this situation is nearly similar to the one attained with quantizers without repetitions in the quantization indices. This similarity in rate-distortion performance exists when the quantizer has a large number of bins.

#### **2.1.4 Slepian-Wolf Encoding in the Basic Wyner-Ziv Codec**

The Slepian-Wolf encoder is the module of the basic Wyner-Ziv video encoder with the goal to reduce the transmission rate of the main information to approach Slepian-Wolf and Wyner-Ziv limits. Typically, the Slepian-Wolf encoding process is performed by channel encoding techniques for the reasons exposed in Chapter 1. Major developments in designing codecs that approach Slepian-Wolf and Wyner-Ziv limits have been recently made as described in the following.

In 1999, Pradhan and Ramchandran [11] have made the first steps towards a practical scheme for distributed source coding using cosets. Their solution is known as DISCUS from DIstributed Source Coding Using Syndromes. The results show that the distributed coding performance benefits of the channel coding techniques usage.

Later, Wang and Orchard [29] reached better results in terms of SNR than the previous one (about 1 dB improvement at an error probability of  $P_e = 10^{-6}$ ) by using embedded trellis codes.

Turbo codes were afterwards tested by García-Frías and Zhao [30], [31], and Bajcsy and Mitran [9], [32] using binary random sequences. Aaron and Girod [33] have also proposed a system based on turbo codes and have tested it using binary random and Gaussian sequences. The results obtained by Aaron and Girod using Gaussian sequences outperformed the results attained in [11] and [29], notably reaching an improvement of about 3 dB in SNR over the last one for an error probability of  $P_e = 10^{-4}$ .

Based on the relationship between Slepian-Wolf coding and channel coding [5], Liveris, Xiong and Georgiades applied low-density parity-check (LDPC) codes [10], [34], to distributed source coding showing also a good performance in terms of the total bitrate ( $R_X + R_Y$ ). Irregular Repeat-Accumulate codes (IRA) were also studied by Liveris *et al.* [35]. The results obtained also show a good performance of the IRA codes in the context of distributed coding.

More recently, new Slepian-Wolf codecs based on turbo codes were designed by Liveris *et al.* [36] and Stankovic, Liveris, Xiong and Georgiades [37] as well as codecs based on LDPC codes by Stankovic *et al.* [37], Schonberg, Pradhan and Ramchandran [38], and Coleman, Lee, Medard and Effros [39]. In a general way, the results obtained show performances closer to the Slepian-Wolf and Wyner-Ziv bounds.

### 2.1.5 Slepian-Wolf Decoding in the Basic Wyner-Ziv Codec

The basic Wyner-Ziv decoder (see Figure 2.1) performs essentially similar operations to those performed at the corresponding encoder, although in an inverted order. Hence, the first operation performed by the basic Wyner-Ziv decoder is the Slepian-Wolf decoding process. At this stage, the decoder uses the side information to perform joint source-channel decoding of the received main information bitstream recovering (estimating) the quantized symbols (output of the quantization module). The “historical” evolution of the Slepian-Wolf decoder is similar to that of the Slepian-Wolf encoder already presented in Section 2.1.4; for that reason, it will not be repeated here.

### 2.1.6 Reconstructing in the Basic Wyner-Ziv Codec

Once the quantized symbols are decoded, the second stage performed at the basic Wyner-Ziv decoder architecture is reconstruction. In this decoding stage, an estimate of the main information transform coefficients is obtained using the decoded quantized symbols together with the side information  $Y$ .

In 2002, Aaron, Zhang and Girod [17] considered the reconstruction function given by the conditional expectation in (2.1) to reconstruct the main information  $X$  pixels values

$$E(X | q', Y) \tag{2.1}$$

assuming a Laplacian distribution to model the residual between corresponding elements of  $X$  and  $Y$  (side information); in (2.1),  $q'$  represents the decoder quantized symbol stream. A similar reconstruction module is considered to reconstruct transform coefficients in more recent solutions proposed by Aaron *et al.* (e.g. [19]).

In 2003, Puri and Ramchandran [40] considered a linear estimate algorithm to reconstruct the transform coefficients which were syndrome encoded; the solution presented in [40] will be described with more detail in Section 2.2.3.

### **2.1.7 Inverse Transforming in the Basic Wyner-Ziv Codec**

To reconstruct the main information, inverse transform must be performed over the transform coefficients estimated in the reconstruction module since the dual operation (transform) is performed at the encoder. This operation has an “historical” evolution similar to that of the transform made in Section 2.1.2; for that reason, it will not be repeated here.

In conclusion, the rate-distortion performance associated to the basic Wyner-Ziv architecture proposed in Figure 2.1 depends on the techniques used to implement each module of the architecture. This Thesis expects not only to study and evaluate the current state-of-the-art but also to make some technical proposals which should allow increasing the performance of DVC solutions following the type of basic architecture here adopted.

## **2.2 Most Relevant Wyner-Ziv Video Coding Solutions**

In many emerging applications, the low-complexity encoding requirement inhibits the traditional video coding paradigm to provide acceptable solutions. For these applications, distributed video coding seems to be able to offer efficient and low-complexity encoding video compression although there is still an efficiency gap between theory and practise. Fundamentally, there are two major areas where distributed video coding finds application, notably low-complexity video coding (as in [12] and [19]) and robust video transmission (as in [20] and [40]). While in the former case the aim is to compress video using a low complexity encoder, in the latter case an additional bitstream is produced in order to correct transmission errors in a traditionally coded video signal.

The first Wyner-Ziv practical schemes in both areas used sample by sample or pixel by pixel encoding and decoding, also called pixel domain Wyner-Ziv coding (e.g. [17], [41], [42] and [43]). After transform coding was studied in the context of distributed source coding (Section



2.1.2), some Wyner-Ziv codecs decided to include a transform module (e.g. [12], [20] and [44]) as shown in Figure 2.1.

In the literature, there are essentially two research groups who have been responsible for the development of the most relevant distributed source video coding systems nowadays available: Bernd Girod's group at Stanford (University of Stanford) and Kannan Ramchandran's group at Berkeley (University of California). In this Section, the most relevant examples of Wyner-Ziv video coding solutions will be presented.

### 2.2.1 Stanford Wyner-Ziv Low-Complexity Video Coding Solution

As it is well-known, a video sequence is composed by images or frames. Typically, these frames are jointly encoded and decoded to exploit the similarities between them; however, due to encoder complexity constraints in emerging application scenarios, such configuration may not be acceptable. With the aim to satisfy the low-complexity encoding and compression efficiency requirements, Stanford's group has presented several coding solutions, e.g. [12], [17] and [42], based on the Slepian-Wolf and Wyner-Ziv theorems:

- In [17], a video sequence is divided into Wyner-Ziv frames (the even frames of the video sequence) and key frames (the odd frames of the video sequence); each Wyner-Ziv frame is pixel by pixel encoded, independently of the key frames and other even frames. To decode a Wyner-Ziv frame, the side information (an estimate of the Wyner-Ziv frame) is generated through frame interpolation techniques using the key frames (which are assumed to be losslessly available at the decoder).
- A more flexible approach was presented in [42] where the number of Wyner-Ziv frames between key frames may vary; the key frames are traditionally intraframe encoded with a H.263+ standard and the Wyner-Ziv frames are encoded as in [17]. Using previously reconstructed frames (both Wyner-Ziv and key frames), frame interpolation or extrapolation techniques are employed to generate the side information.
- In [12], an architecture similar to the one in [42] is proposed. The major difference is that in [12] transform coding is considered in Wyner-Ziv frame coding; again, the Wyner-Ziv frames are even frames of the video sequence and the remaining frames are the key frames.

In a nutshell, the Stanford's approaches briefly described above are based on an intraframe encoder-interframe decoder system. In other words, the solutions described rely on a structure where each Wyner-Ziv frame is encoded independent of the other Wyner-Ziv frames and key frames, i.e. similarities with other video frames are not exploited at the encoder, but the decoding process is performed jointly.

In traditional video coding (interframe encoding and decoding), a predictive framework is used both at the encoder and decoder to explore the similarities between frames and thus to achieve high compression efficiency. In an intraframe encoder-interframe decoder system, high compression efficiency may be achieved in the joint decoding process since similarities

between frames are explored at the decoder through frame interpolation or extrapolation techniques.

The most recent Wyner-Ziv low-complexity video coding solution originating from the Stanford's group was proposed in [19] by Aaron, Rane and Girod. In this solution, beside the bitstream resulting from the current Wyner-Ziv frame encoding process, the encoder also generates and transmits supplementary information about the current Wyner-Ziv frame to help the decoder in the motion extrapolation task (to generate the side information); this supplementary information is kept in a small memory at the encoder. Since minimal computation and memory is involved in the generation and storage of the supplementary information compared to traditional interframe predictive coding, the solution in [19] is considered by the authors as a near intraframe encoding-interframe decoding solution. Figure 2.3 illustrates the architecture proposed in [19].

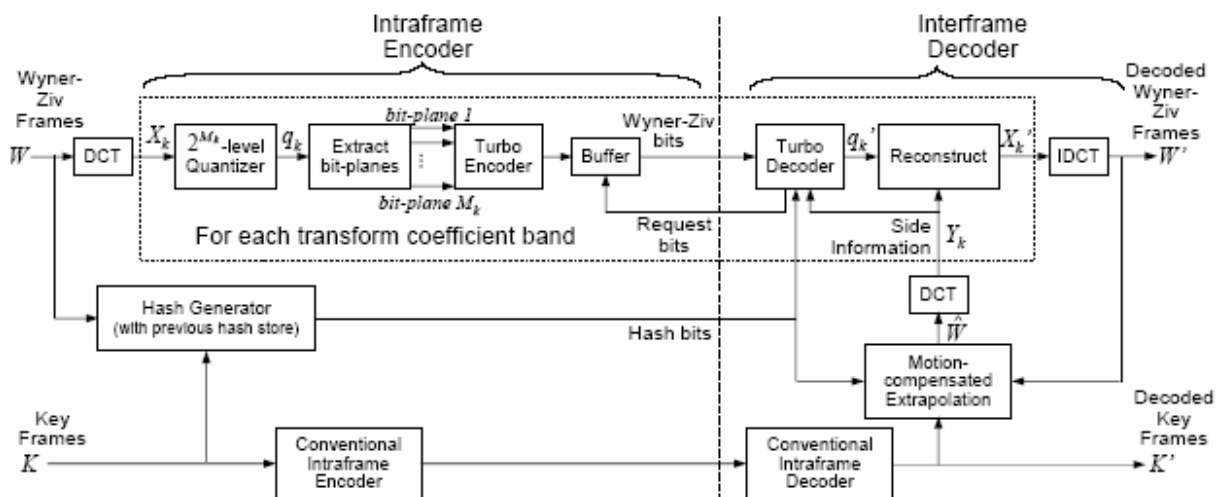


Figure 2.3 – Stanford Wyner-Ziv video codec architecture [19].

In the proposed solution [19], the frames of a video sequence are organized into Groups Of Pictures (GOPs). Each GOP is constituted by a key frame (the first frame of the GOP) and by Wyner-Ziv frames (the remaining frames until the next key frame). The number of Wyner-Ziv frames between two consecutive key frames defines the GOP length. The key frames, represented in Figure 2.3 by  $K$ , are intraframe encoded and decoded using a traditional video coding standard (in this case the H.263+ standard is used).

The Wyner-Ziv frames,  $W$ , are intraframe encoded using the tools mentioned in Section 2.1: transform coding (represented in Figure 2.3 by the DCT module), quantization and turbo coding (as Slepian-Wolf coding). After applying a  $4 \times 4$  discrete cosine transform (DCT) to the  $W$  frame, each transform coefficient band  $X_k$  is independently encoded: the transform coefficients of the  $X_k$  band are quantized and bitplane extraction is performed over the resulting quantized symbol stream  $q_k$ ; each bitplane is then independently turbo encoded. For the Wyner-Ziv frame  $W$ , beside the bits resulting from the intraframe encoding, the Wyner-Ziv bits, the encoder sends additional bits about the current  $W$  frame called hash bits; the hash information associated

to a  $4 \times 4$  block of  $X$  consists of a small number of the quantized transform coefficients. These hash bits help the decoder in the motion estimation task in order to attain more accurate side information.

For the transform coefficient band  $X_k$ , the decoder performs joint decoding of the  $M_k$  bitplane using the Wyner-Ziv bits together with the side information  $Y_k$ ; for the current frame  $W$ , the corresponding side information  $\hat{W}$  is generated through motion extrapolation techniques using the previously decoded frame (both key frame or Wyner-Ziv frame) and the received hash bits.

The side information  $Y_k$  corresponding to  $X_k$  is obtained by applying a DCT transform over  $\hat{W}$ . The feedback channel in Figure 2.3 architecture is used by the decoder to request for more Wyner-Ziv bits; for the  $M_k$  bitplane decoding, a request for more Wyner-Ziv bits is made if the current bitplane error probability is higher than pre-assigned bitplane error probability threshold, typically  $10^{-3}$ .

Once all the  $M_k$  bitplanes are decoded, the decoded quantized symbol stream  $q_k'$  can be obtained. The side information  $Y_k$  together with the decoded quantized symbol stream  $q_k'$  are used to reconstruct the transform coefficient band  $X_k$ . When all the transform coefficient bands are decoded, the decoded Wyner-Ziv frame  $W'$  can be obtained by applying the Inverse Discrete Cosine Transform (IDCT).

The periodical transmission of the key frames provides some resynchronization relatively to the error propagation (distortion) from a reconstructed frame to the side information of the next temporally adjacent frame  $W$ .

Comparing the architecture in Figure 2.3 with the architecture in Figure 2.1:

- The Wyner-Ziv frames  $W$  correspond to the **main information** (sequence  $X$  in Figure 2.1).
- The information resulting from the motion-compensated extrapolation module,  $\hat{W}$ , is the **side information** (sequence  $Y$  in Figure 2.1) associated to  $W$ . In turn,  $Y_k$  ( $k^{\text{th}}$  transform coefficient band of  $\hat{W}$ ) refers to the side information associated to  $X_k$  ( $k^{\text{th}}$  transform coefficient band of  $W$ ).
- Regarding the Figure 2.1 architecture, an additional bitstream (corresponding to the hash bits) is transmitted to the decoder in Figure 2.3 architecture in order to generate better side information.

In the following, a more detailed description of the solution depicted in Figure 2.3 will be presented.

### 2.2.1.1 Encoding Procedure

Two encoding procedures can be distinguished in Figure 2.3 architecture: the encoding of the key frames and of the Wyner-Ziv frames. If the current frame is a key frame,  $K$ , it is traditionally intraframe encoded using the H.263+ standard. For each  $4 \times 4$  block of  $K$ , a small

amount of the quantized transform coefficients is stored in a small hash memory; these  $4 \times 4$  block coefficients are then used to help to decide if hash bits must be transmitted for the co-located block in the next temporally adjacent Wyner-Ziv frame. If the current frame is a Wyner-Ziv frame,  $W$ , the encoding process is performed using the following five stages:

- 1) **Transform (DCT):** The first step to encode a Wyner-Ziv frame  $W$  is transform coding (represented in Figure 2.3 by the DCT module); a  $4 \times 4$  block-based discrete cosine transform (DCT) is applied over frame  $W$ . The transform coefficients of the whole frame  $W$  are then grouped together, according to the position occupied by each transform coefficient within the  $4 \times 4$  blocks, forming the so-called transform coefficient bands; in Figure 2.3,  $X_k$  represents the  $k^{\text{th}}$  transform coefficient band of  $W$ .
- 2) **Quantizer:** Each transform coefficient band  $X_k$  is then uniformly quantized with  $2^{M_k}$  levels producing the quantized symbol stream  $q_k$ .
- 3) **Extract Bitplanes:** Over the resulting quantized symbol stream  $q_k$  associated to the transform coefficient band  $X_k$ , bitplane extraction is performed; this means, the  $X_k$  band quantized symbols bits of the same importance (e.g. the most significant bit) are grouped together forming the corresponding bitplane array.
- 4) **Turbo Encoder:** Each bitplane is then independently fed into the turbo encoder (which plays the role of the Slepian-Wolf encoder in Figure 2.1); the Slepian-Wolf codec is built based on a Rate Compatible Punctured Turbo (RCPT) code structure [45]. The turbo encoder generates parity information for each bitplane which is stored in the buffer and transmitted in small amounts upon decoder request via the feedback channel.
- 5) **Hash Generator:** Beside the Wyner-Ziv bits, the encoder also produces and sends additional bits, designated by hash bits in Figure 2.3, to help the decoder in the motion estimation task associated to the creation of the side information. For each  $4 \times 4$  samples block within the Wyner-Ziv frame  $W$ , the hash code, as is called, corresponds to a small amount (not specified in [19]) of the  $W$  quantized transform coefficients; the hash codewords of a Wyner-Ziv frame are stored in a small hash memory in order to be used to help deciding for each  $4 \times 4$  block of the next temporally adjacent Wyner-Ziv frame hash bits must be transmitted. By computing the distance between each  $4 \times 4$  block of the current frame  $W$  and the co-located hash code in the previous frame, it is decided for each  $4 \times 4$  block of the current frame if hash bits must be transmitted. This decision is based on the thresholding of the computed distance for each  $4 \times 4$  block: for a distance smaller than a given threshold, a “no hash bits” codeword is sent; otherwise, the hash bits associated to the  $4 \times 4$  block are transmitted (beside the Wyner-Ziv bits).

Since the hash code generation and storage procedures only require minimal computation and memory, the authors state that the Wyner-Ziv frames encoding complexity is similar to a traditional intraframe encoding complexity.

### 2.2.1.2 Decoding Procedure

As shown in the Figure 2.3 decoding architecture, one of two decoding procedures is performed, in accordance to the encoding procedure describe in Section 2.2.1.1. For key frame  $K$ , a traditional intraframe decoder using the H.263+ standard is employed; the decoded key frame is then used in the next temporally adjacent Wyner-Ziv frame decoding process to generate  $\hat{W}$  (an estimate of the  $W$  frame) by means of motion estimation. For the Wyner-Ziv frame  $W$ , the decoding procedure is described by the following five stages:

- 1) **Motion-Compensated Extrapolation:** The decoder performs frame extrapolation using the received hash bits and the previous reconstructed frame (Wyner-Ziv frame or key frame) to generate an estimate of frame  $W$ , called  $\hat{W}$ . More specifically, for each  $4 \times 4$  block of the current  $W$  frame, two situations may occur:
  - If a “no hash bits” codeword is received, the corresponding block in the  $\hat{W}$  frame is filled with the co-located samples block from the previous reconstructed frame.
  - If the decoder receives hash bits, the corresponding block in the  $\hat{W}$  frame is generated from the previous reconstructed frame through a motion search based on the received hash bits.

A block-based  $4 \times 4$  DCT is then performed over the  $\hat{W}$  frame to obtain the side information transform coefficient bands  $Y_k$  corresponding to the transform coefficient bands  $X_k$ . To make the side information useful to the following stages (turbo decoding and reconstruction), a statistical dependence model between corresponding coefficients in  $X_k$  and  $Y_k$  must be considered. In [19], the authors assume that the difference between the corresponding elements in  $X_k$  and  $Y_k$  is modelled by a Laplacian distribution.

- 2) **Turbo Decoder:** The decoded quantized symbol stream  $q_k'$  associated to the transform coefficient band  $X_k$  is obtained through a turbo decoding procedure. For each transform coefficient band, the turbo decoder starts decoding the most significant bitplane followed by the sequential decoding of the remaining bitplanes. Each transform coefficient band bitplane is decoded using the received Wyner-Ziv bits associated to that bitplane and the side information  $Y_k$ . When the received Wyner-Ziv bits together with  $Y_k$  are not sufficient to provide a reliable decoding of the current bitplane, more bits are requested via the feedback channel by the decoder; the feedback channel is thus necessary to adapt to the changing statistics between the side information and the frame to be encoded. After the additionally requested Wyner-Ziv bits are received, a new attempt to decode the relevant bitplane is performed. The requests and following decoding operations are executed until the current bitplane error probability,  $P_e$ , is lower than  $10^{-3}$ ; in this case, the turbo decoding of a transform coefficient band bitplane is considered to be successful. An ideal error detection capability is assumed at the decoder to determine the current bitplane error probability of a given transform coefficient band, i.e. the turbo decoder is able to measure in a perfect way the transform coefficient band current bitplane error probability.

In general, due to the availability of the side information, the number of Wyner-Ziv bits required to determine in which quantization interval (level) a transform coefficient is mapped to, from the  $2^{M_k}$  possible levels, is lower than  $M_k$  bits and thus compression efficiency achieved. Notice that the more accurate the side information is, the higher is the compression efficiency since fewer Wyner-Ziv bits are required to provide a reliable decoding.

After turbo decoding the  $M_k$  bitplanes associated to the DCT band  $X_k$ , the bitplanes are grouped together to form the decoded quantized symbol stream  $q_k'$ .

- 3) **Reconstruct:** Given the reconstructed quantized symbol stream  $q_k'$  and the side information  $Y_k$ , the reconstruction of each transform coefficient band,  $X_k'$ , is computed through the conditional expectation  $E(X_k | q_k', Y_k)$ .
- 4) **Inverse Transform (IDCT):** After all transform coefficient bands are reconstructed, a block-based  $4 \times 4$  inverse discrete cosine transform (represented in Figure 2.3 by the IDCT module) is performed and the reconstructed  $W$  frame,  $W'$ , is obtained.

The architecture proposed in [19] and described in Section 2.2.1 constitutes the state-of-the-art reference for the transform domain solution described in Chapter 4. The state-of-the-art reference for the pixel domain solution described in Chapter 3 is the solution proposed in [17]; this solution is a simpler version of the architecture proposed in [19] (which generically makes use of a quantizer, a turbo codec, a reconstruction module and an interpolation module) and therefore it is not described in detail in this Thesis.

### 2.2.1.3 Some Experimental Results

In order to evaluate the performance of the system proposed by Aaron *et al.* in [19], the authors considered two QCIF video sequences, *Salesman* and *Hall Monitor*, at 10 frames *per second*. Figure 2.4 and Figure 2.5 show the PSNR results obtained for the test content used. The values in the horizontal and in the vertical axis contemplate both the key frames and the Wyner-Ziv frames. Each figure shows a set of curves corresponding to different GOP lengths: as it can be seen, the rate-distortion performance of the solution in [19] exhibits significant coding gains (up to 9 dB) over traditional DCT-based intraframe coding.

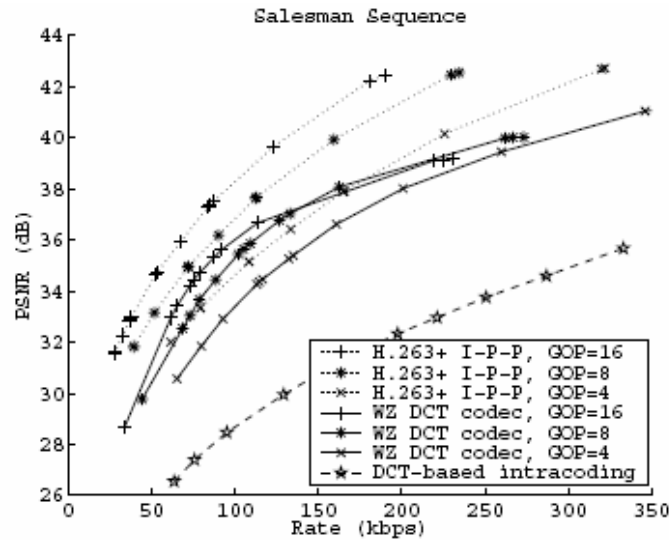


Figure 2.4 – Average PSNR for the Salesman sequence [19].

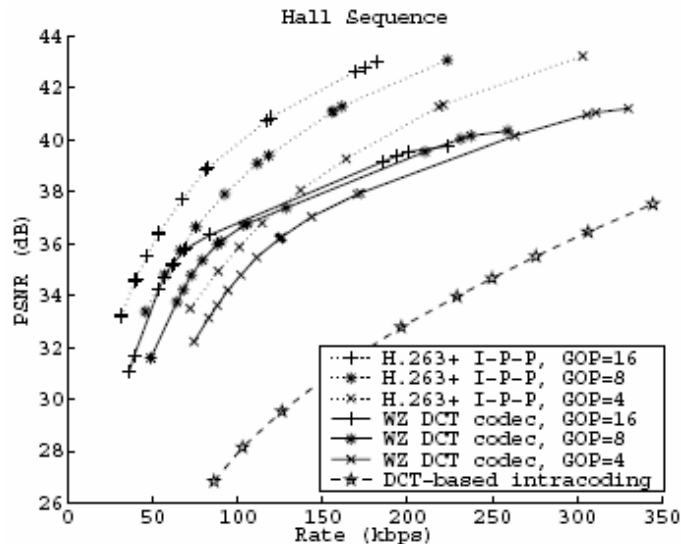


Figure 2.5 – Average PSNR for the Hall Monitor sequence [19].

Comparing the solution in Figure 2.3 with H.263+ interframe coding (with I-P-P structure) for the same GOP size, the results show that the rate-distortion performance of the DVC coding scheme is 1 to 4 dB worse than that obtained with H.263+ interframe coding for the two mentioned sequences. Notice that this difference in the rate-distortion performance is more accentuated for larger GOP sizes. In a traditional interframe coding with I-P-P structure, increasing the number of P frames between two consecutive I frames corresponds to a reduction in the total bitrate for a certain quality since in P frames the temporal correlation between frames is exploited. However when the coding solution proposed in this Section is used, the total bitrate may increase with the increasing of the GOP length for a certain quality. Once the number of  $W$  frames between  $K$  frames increases, the bitrate associated with the  $K$  frames, traditionally intraframe encoded, diminishes. However the error propagation (distortion) from a reconstructed frame to the side information of the next temporally adjacent  $W$  frame grows

since when the key frames are more distant from each other the motion interpolation fails more often; since the motion interpolation fails, the interpolated frame becomes less reliable (with lower quality) and therefore more parity bits are needed to correct the “errors” between  $X_k$  and  $Y_k$ . Thus, to achieve a certain reconstruction quality, an increase of the bitrate may be needed. For more details about this system, the reader should consult [19].

### **2.2.2 Stanford Wyner-Ziv Robust Video Coding Solution**

When a video bitstream is transmitted over an error-prone channel it may become corrupted by the errors introduced by the transmission channel. To achieve a suitable decoded video quality, some of the channel transmission errors are corrected while other are concealed depending on the techniques utilized by the video coding system to deal with transmission errors. Forward Error Correction (FEC) techniques try to solve the problem of error correction by appending error check information to the video bitstream in order to correct (at least some) transmission errors. However, a “cliff” effect is typically observed when bit error probability exceeds the FEC error correction capabilities meaning that error correction suddenly completely fails and, due to interframe error propagation, the quality (measured in terms of PSNR) rapidly drops creating the cliff effect. In alternative to the FEC techniques, Stanford’s group has proposed approaches using pixel domain Wyner-Ziv coding [41], [43] to protect a video bitstream from transmission errors. The results obtained with these approaches show that an additional Wyner-Ziv bitstream can be used to simultaneously achieve strong protection against channel errors and graceful degradation of the video quality.

Figure 2.6 illustrates the architecture of a recent solution proposed in [20] by the Stanford’s group which uses an improved Wyner-Ziv codec compared with previous work. The goal of this new solution is to make a traditionally encoded bitstream becoming error resilient when it is transmitted over an error-prone channel with few or no protection against errors introduced by the channel, for instance by means of channel coding.



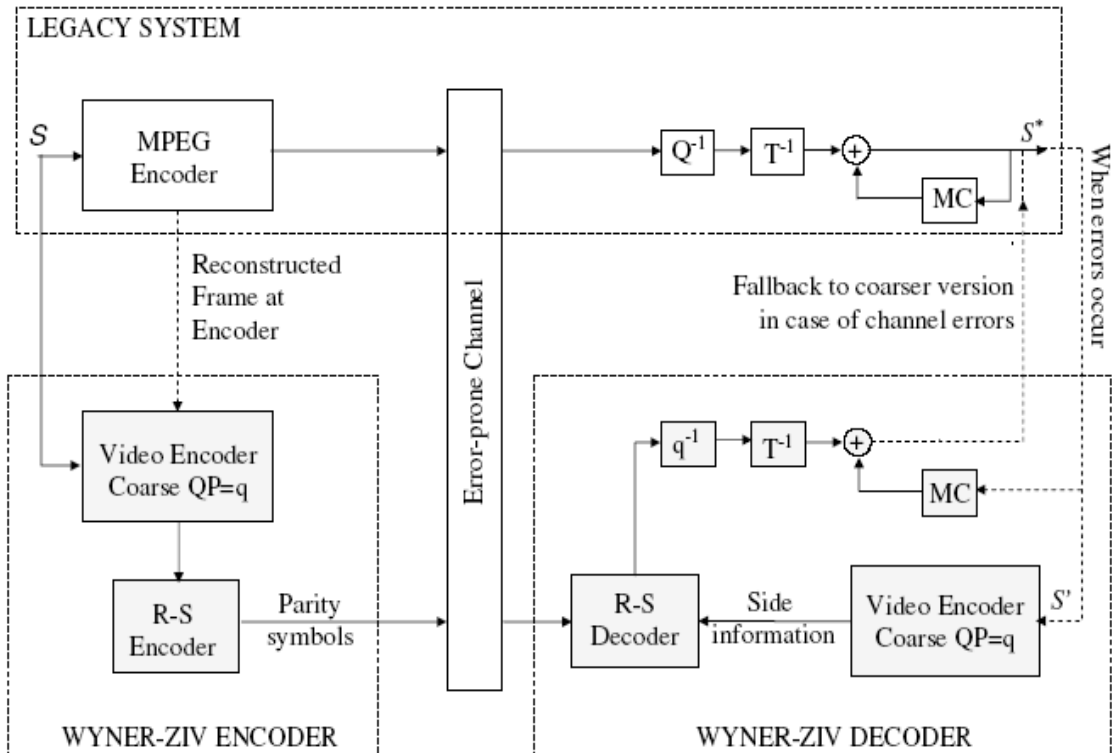


Figure 2.6 – Systematic lossy Forward Error Protection (FEP) system architecture [20].

Systematic lossy Forward Error Protection (FEP) scheme is the terminology used in the literature to refer to the coding architecture in Figure 2.6: In the context of the approach proposed in [20], the concept of “systematic coding” corresponds to the bitstream produced by the legacy system (MPEG encoder) for the video sequence  $S$  and transmitted with minimal or no protection against transmission errors.

Beside this bitstream produced by the legacy system (in this case an MPEG encoder), a supplementary coding framework based on Wyner-Ziv coding is used to produce an additional bitstream for  $S$  with a lower bitrate than the legacy system encoded bitstream. The additional bitstream is used by the decoder to correct transmission errors in the legacy system (MPEG) decoded video frames, thus providing better decoded video quality. The Wyner-Ziv encoder uses a traditional hybrid video encoder to produce a coarsely quantized version of the video sequence  $S$  and a Reed-Solomon (R-S) encoder to generate parity bits; these parity bits, often called Wyner-Ziv bits, are then transmitted over an error-prone channel, together with the bitstream produced by the legacy system (MPEG) encoder. At the Wyner-Ziv decoder, the received Wyner-Ziv bits together with a coarsely quantized version provided by the legacy system,  $S'$ , will provide a decoded video sequence with better visual quality.

Comparing the architecture in Figure 2.6 with the architecture in Figure 2.1:

- The video sequence  $S$  corresponds to the **main information** (sequence  $X$  in Figure 2.1).
- The legacy system decoded video sequence  $S'$  corresponds to the **side information** (sequence  $Y$  in Figure 2.1).

The encoding and decoding procedures of the FEP architecture will be described with more detail in the following sections.

### **2.2.2.1 FEP Encoding Procedure**

In Figure 2.6, two key encoders can be distinguished: the MPEG encoder and the Wyner-Ziv encoder. In both encoders, the video frames to be encoded are divided into slices (a sequence of macroblocks in raster-scan order) in order to provide error resilience. While this does not constitute a novelty for MPEG coding, the same is not true for Wyner-Ziv coding comparatively to other solutions presented in Section 2.2. Since MPEG is a well-known coding technique, the MPEG encoding process will not be detailed here. Instead, a special attention will be given to the Wyner-Ziv encoding procedure.

- 1) **MPEG Encoder:** The video frames  $S$  are traditionally compressed using an MPEG video encoder; the resulting bitstream is then transmitted over a channel that may introduce errors, corrupting the video bitstream. Note that this encoder is to be taken just as an example of traditional encoding not having therefore any mandatory nature, e.g. a H.26x encoder could also be used.
- 2) **Wyner-Ziv Encoder:** The same video frames  $S$  are also encoded with a Wyner-Ziv encoder that comprises a coarse video encoder and a Reed-Solomon encoder.

#### **Coarse Video Encoder**

This video encoder performs the same operations as the MPEG encoder with the exception of the quantization stage. Coarser step sizes are used in this coarse video encoder to encode  $S$  yielding a lower-rate representation than the one produced by the legacy system (MPEG) encoder. Instead of encoding the original frames  $S$ , the coarse video encoder encodes the current locally available MPEG decoded frames in order to use for the predictive coding reference frames as similar as possible to those used by the coarse video encoder at the Wyner-Ziv decoder; thus, mismatch between encoder and decoder is prevented. The current locally available MPEG decoded frame is represented in Figure 2.6 under the name of “Reconstructed Frame at Encoder”.

#### **R-S Encoder**

The bitstream produced by the coarse video encoder constitutes the input to a channel encoder, in this case a Reed-Solomon (R-S) encoder; the channel encoder applies, across the slices of a whole frame, a systematic Reed-Solomon code with byte-long symbols (see Figure 2.7). The parity symbols generated by the R-S encoder constitute the additional bitstream to be transmitted to the decoder; this additional bitstream is called the Wyner-Ziv bitstream.



### 2.2.2.3 Some Experimental Results

In [20], the FEP system PSNR performance is compared with the traditional FEC system PSNR performance when both systems are used for video broadcasting considering an error prone channel scenario. In the traditional FEC system, the MPEG coded video signal is also encoded with a Reed-Solomon code in order to protect the source bitstream against transmission errors. The simulations were performed for two CIF video sequences: *Foreman* and *Coastguard*. In this Section, only the results obtained in [20] for the *Foreman* sequence will be reproduced since those results are sufficient to compare the FEP and the FEC system performances. Table 2.1 provides the main simulation conditions adopted for the *Foreman* video sequence.

Table 2.1 – Main simulation conditions for the *Foreman* video sequence.

<b>Spatial Resolution</b>	CIF
<b>Number of Frames Evaluated</b>	50
<b>Number of Consecutive Macroblocks/Slice</b>	11
<b>Number of Slices/Frame</b>	36
<b>Legacy System Bitrate [Mbps]</b>	1
<b>Wyner-Ziv Bitrate [kbps]</b>	270
<b>R-S Code for FEP System (Wyner-Ziv Coding)</b>	(52, 36)
<b>FEP Parity Information Bitrate [kbps]</b>	120
<b>R-S Code for FEC System</b>	(40, 36)
<b>FEC Parity Information Bitrate [kbps]</b>	120

As it can be observed in Figure 2.8, which shows the PSNR performance for both the FEP and FEC schemes, the “cliff” effect associated to PSNR dropping is equally present in both systems; however it appears later for the FEP scheme. Hence, an acceptable video quality is still assured for higher channel error rates when the FEP scheme is used in comparison with the traditional FEC scheme. It is notorious that, for symbol error rates lower than  $3.5 \times 10^{-4}$ , the proposed FEP architecture exhibits a worse quality performance when comparing to the traditional FEC solution since the quantization errors are propagated to the next temporally adjacent frames through the fallback mechanism; in this context, symbol means one byte-long information packet.

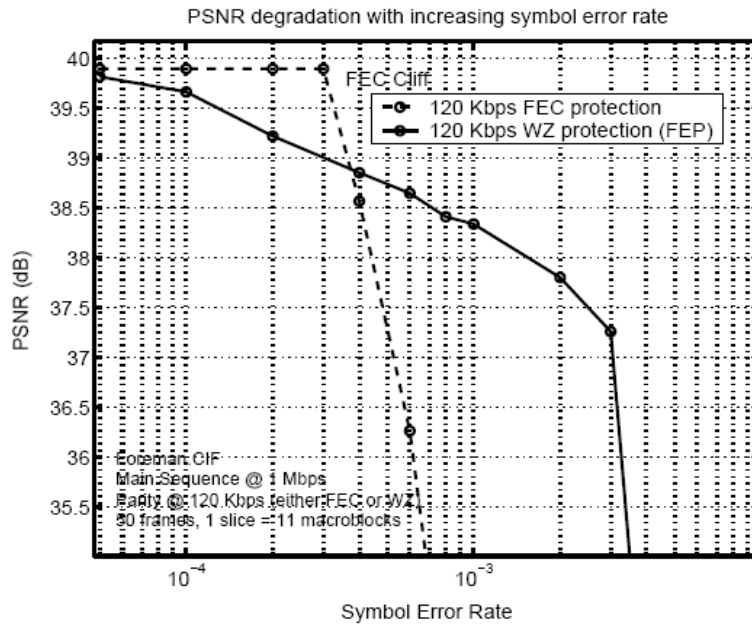


Figure 2.8 – Performance comparison between FEP and FEC systems for the same parity information rate [20].

Figure 2.9 depicts the performance for the FEP and FEC systems in terms of visual quality at a high symbol error rate ( $10^{-3}$ ). For the image on the left, the legacy bitstream (MPEG encoded) was protected against channel transmission errors with a FEC technique, in this case a (40, 36) R-S code, while for the image on the right the legacy system bitstream was protected against channel transmissions errors with an additional bitstream: the Wyner-Ziv bitstream. As it can be noticed, for the conditions at hand, e.g. a symbol error rate of  $10^{-3}$ , the FEP system yields a higher visual quality than the FEC system.

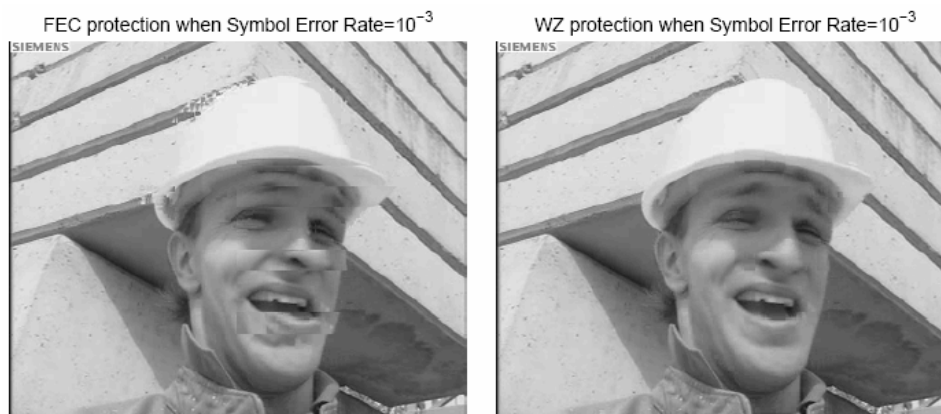


Figure 2.9 – Performance comparison in terms of visual quality between FEP and FEC systems for a symbol error rate of  $10^{-3}$  [20].

Considering the traditional FEC system and the FEP system architecture depicted in Figure 2.6, it is interesting to point out that the FEP system is similar to the FEC system when the quantization parameters in the FEP system are made equal for both the legacy system encoder

and the coarse video encoder (enclosed by the Wyner-Ziv encoder). For more details about this FEP system, the reader should consult [20].

### 2.2.3 Berkeley Wyner-Ziv Robust Video Coding Solution

In traditional interframe predictive coding architectures, the distribution of the computational burden is rather rigid and the sensibility to prediction mismatch drifts between encoder and decoder (e.g. caused by transmission errors) is very high. Using intraframe coding, the coding architecture becomes more robust to channel transmission errors but at the price of decreasing the compression efficiency. This Section will present a coding solution whose major goal is the robustness to interframe propagation of transmission errors. This approach was presented by the Berkeley’s group [40] under the name of “Power-efficient, Robust, hIgh-compression, Syndrome-based Multimedia coding” or PRISM, as it is usually referred in the literature. The PRISM solution aims therefore to combine intraframe coding features (low-complexity encoding and robustness to transmission errors) with interframe coding compression efficiency.

The PRISM solution proposes a new video coding scheme based on Wyner-Ziv coding (see Figure 2.1). However this solution uses the concept of side information differently from the description given in Chapter 1. The ‘single’ side information that characterizes Wyner-Ziv coding as presented in Chapter 1 is here substituted by several side information candidates [40]. This innovation will be opportunely explained later while describing the PRISM decoding procedure. Figure 2.10 and Figure 2.11 illustrate the architectures of the PRISM encoder and decoder, respectively.

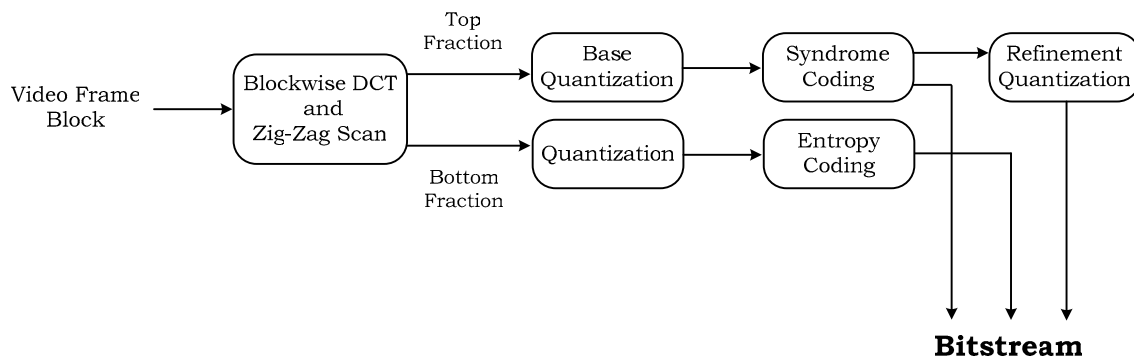


Figure 2.10 – PRISM encoder architecture [40].

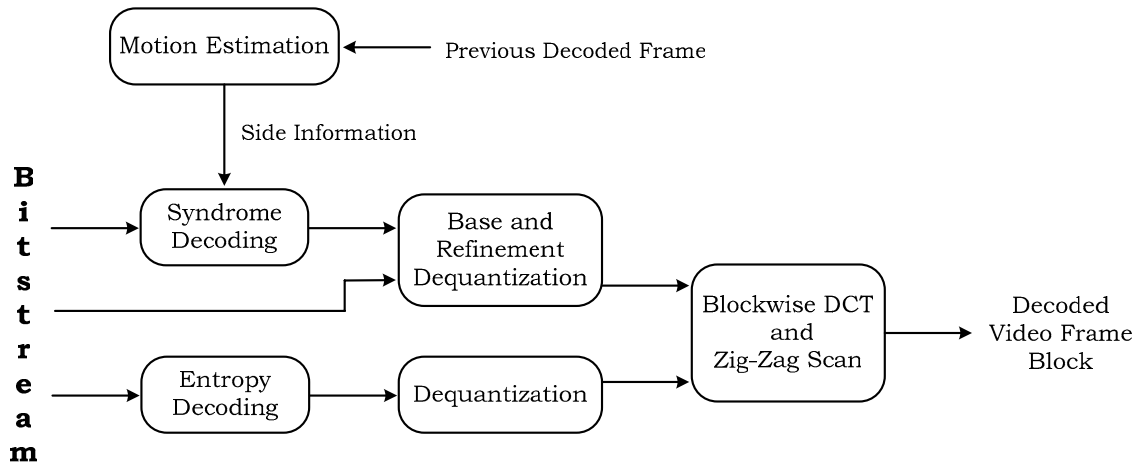


Figure 2.11 – PRISM decoder architecture [40].

In a nutshell, each video frame is divided into  $n \times n$  samples blocks and a blockwise discrete cosine transform (DCT) is applied over each  $n \times n$  samples block; the transform coefficients are then zig-zag scanned. A small number of transform coefficients is coarsely quantized and syndrome encoded while the remaining transform coefficients are traditionally encoded (quantized and entropy encoded). The coarsely quantized coefficients can be further refined in order to achieve a higher decoded quality (low quantization step size). Beside the streams depicted in Figure 2.10, the PRISM encoder bitstream at the  $n \times n$  block level also encloses a Cyclic Redundancy Check (CRC) stream of the base quantized transform coefficients; the CRC will help determining the best candidate side information block from those  $n \times n$  blocks generated by the decoder. Once the best side information block is known, syndrome decoding can be performed. After reconstructing the transform coefficients, the best video frame reconstruction can be obtained.

Comparing the architectures in Figure 2.10 and Figure 2.11 with the architecture in Figure 2.1, it can be stated that for an  $n \times n$  block:

- The **main information** corresponds to the quantized transform coefficients top fraction syndrome encoded (see Figure 2.10).
- The **side information** is composed of several candidates to prediction block instead of the ‘single’ side information that characterizes the architecture in Figure 2.1; the candidates to prediction blocks are generated through half-pixel motion search in the previous reconstructed frame.
- The CRC and the transform coefficients bottom fraction (quantized and entropy encoded) are considered additional information transmitted from the encoder to the decoder. The CRC helps the decoder to decide which candidate side information block is more similar to the original block.

The PRISM encoding and decoding procedures will be analysed with more detail in the following.

### **2.2.3.1 PRISM Encoding Procedure**

Consider the PRISM encoding architecture illustrated in Figure 2.10. Each video frame is first divided into  $8 \times 8$  or  $16 \times 16$  samples blocks; in [40] it is not specified in which circumstances it is used one or other block dimension.

Regarding the previous frame, different regions of the current frame may be described by different amounts of motion; different frame regions (e.g. blocks of samples) may therefore be characterized by different correlation intensities. Before carrying out the operations shown in Figure 2.10 architecture, one previous stage is performed; this stage is called classification.

#### **Classification**

In the classification stage, each  $8 \times 8$  or  $16 \times 16$  samples block is classified into one of several pre-defined classes according to the correlation (statistical dependency) between the current frame block and the co-located block in the previous frame (temporal predictor of the current frame block). In [40], the authors model the correlation intensity through the squared error between each current frame block and the co-located one in the previous frame. The classification stage helps to decide what kind of encoding is well-suited for each block of the current frame: no coding (skip class), traditional coding (intra coding class) and syndrome coding (syndrome coding class). Thus, the current frame blocks classified in the skip class are not encoded and the blocks classified in the intra coding class are traditionally encoded. The blocks classified in the syndrome coding class constitute the major PRISM novelty; therefore, a special attention will be given to the encoding/decoding procedures of those blocks. The encoding modes classes selected for the current frame blocks are then transmitted to the decoder as header information.

**1) Blockwise DCT and Zig-Zag Scan:** To each current frame samples block, a blockwise DCT is applied and the resulting transform coefficients are then zig-zag scanned.

Usually, in real (not synthesized) images, most of the block's energy is concentrated in a small number of transform coefficients corresponding to the lower frequency coefficients. Relying on this idea, for the blocks classified in the syndrome coding mode class, the PRISM solution encodes the low-frequency coefficients using syndrome coding while the high-frequency coefficients are traditionally encoded (i.e. quantized and entropy encoded). Typically, many of the high-frequency coefficients have low or near-zero values and therefore entropy coding uses few bits to send those transform coefficients. On the other hand, the low-frequency coefficients have high values and the syndrome encoding will allow reducing the bitrate needed to transmit them; bitrate reduction is achieved since instead of transmitting each individual codeword corresponding to a quantized transform coefficient, syndrome coding only transmits the index of the set containing that codeword.

**2) Quantization:** The zig-zag scanned transform coefficients are then quantized generating quantized codewords. The DC coefficient (the lowest frequency transform coefficient) and a small number of AC coefficients near the DC (in a zig-zag scan order) are quantized in the



base quantization architectural module; these transform coefficients are called Wyner-Ziv coefficients. The choice of the base quantization step size is constrained by the correlation level of each  $8 \times 8$  or  $16 \times 16$  block within the current frame determined in the classification stage. The remaining transform coefficients (high-frequency coefficients) are fed into the other quantization module (Figure 2.10); the quantization step size corresponds to the desired reconstruction quality (distortion). Hence, the base quantization and the quantization architectural modules in Figure 2.10 differ in the quantization step size that typically assumes different values.

- 3) Syndrome Coding:** The quantized codewords space is divided into several groups of codewords called cosets. Each coset has an index label associated; this index label, known as syndrome, points out the coset to which the codeword corresponding to a quantized transform coefficient belongs to. Since the number of syndromes is lower than the number of codewords, the number of bits required to encode a syndrome is inferior to the number of bits needed to encode a codeword. Using syndrome coding, the transmission of individual codewords associated to the quantized transform coefficients is replaced by the transmission of syndromes and therefore compression is achieved.

In the PRISM solution, a trellis-based syndrome code (128-state rate- $\frac{1}{2}$  trellis code) is applied to quantized low-frequency transform coefficients (typically about 20%) of each  $8 \times 8$  or a  $16 \times 16$  block, represented in Figure 2.12 by the pink listed area. The syndrome encoding resulting bits (known as syndrome bits) are incorporated in the bitstream syntax at the block level.

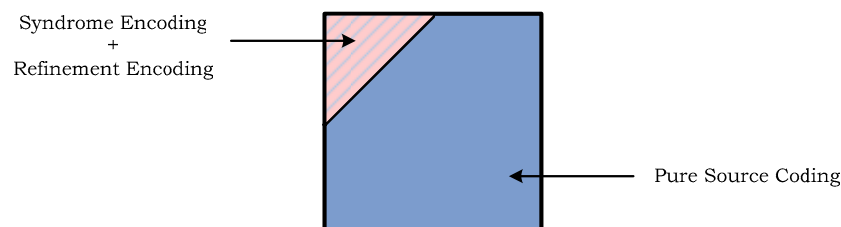


Figure 2.12 – Selective encoding for the various transform coefficients within a block [40].

- 4) Refinement Quantization:** As it is well-known, compressing a signal with different quantization step sizes corresponds to attaining different signal reconstruction quality (distortion) levels at the decoder. Hence, for instance, to attain a desirable block reconstruction quality, a specific quantization step has to be chosen. In Figure 2.10, low-frequency and high-frequency transform coefficients of  $8 \times 8$  or  $16 \times 16$  blocks are quantized with different quantization step sizes, as was mentioned when describing the quantization process. For low-frequency coefficients (syndrome encoded), the choice of the quantization step size depends on the correlation between a block and the co-located one in the previous frame, determined at the classification stage; by doing this, the trellis codes decoding error probability is minimized. In order to attain a global desirable reconstruction quality, a refinement of the base quantization step size is performed. The refinement quantization

process corresponds to sub-partitioning the base quantization interval in order to obtain the quantization step size corresponding to the desirable reconstruction quality. The sub-partitions within the base quantization interval are called refinement intervals; each refinement interval has an index associated to it. The refinement bits associated to the refinement interval index are transmitted to the decoder; these bits are another component of the bitstream syntax at the block level.

- 5) Entropy Coding:** The quantized transform coefficients corresponding to the high-frequency coefficients that have not been syndrome encoded (blue area in Figure 2.12) are then traditionally entropy encoded using run-length Huffman coding. The resulting bits, called pure source coded bits, are incorporated in the bitstream syntax at the block level.

Beyond the five steps performed by the PRISM encoder (DCT, quantization, syndrome coding, refinement quantization, and entropy coding), a cyclic redundancy check (CRC) of the base quantized transform coefficients is also computed and transmitted to help the decoder performing the motion estimation task. The bitstream syntax at the block level encloses therefore syndrome bits, CRC bits, refinement bits and pure source coding bits, as is shown in Figure 2.13.

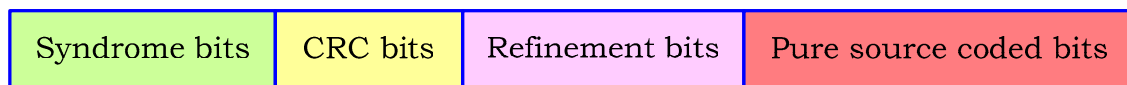


Figure 2.13 – Bitstream syntax at the block level [40].

Considering Figure 2.10, PRISM complexity is essentially related to the transform and the entropy coding modules [40] as in traditional intraframe encoding. Thus, it can be said that the encoding complexity of the PRISM solution is similar to that of a traditional intraframe encoding solution, e.g. JPEG.

### 2.2.3.2 PRISM Decoding Procedure

The PRISM decoder architecture is presented in Figure 2.11. For the frame blocks not encoded (i.e. blocks classified in the skip class), the co-located blocks in the previous reconstructed frame are used as reconstructed blocks; the blocks classified in the intra coding class are decoded using inverse operations to those performed at the encoder: entropy decoding and dequantization. The decoding procedure of the blocks classified in the syndrome coding class is described in the following five steps:

- 1) Motion Estimation:** The most important task performed at the decoder is the motion estimation which provides the information necessary to decode the received syndrome bits. In the PRISM decoding architecture, this information consists of several candidates to prediction block (i.e. several side informations) instead of the single side information that characterizes the Wyner-Ziv coding scenario described in Chapter 1. The candidate

predictors are obtained from the previous reconstructed frame by half-pixel motion estimation. A full motion search motion estimation algorithm is used (half-pixel accuracy), i.e. all neighboring blocks within a search range are used as candidate blocks; this is very similar to what is done at the encoder side in traditional video codecs.

- 2) **Syndrome Decoding:** For each  $8 \times 8$  or  $16 \times 16$  block, the received syndrome bits together with one of the candidate predictors are used to decode the sequence of quantized codewords. From the received syndrome bits, it is possible to obtain several quantized codeword sequences. To find out within this set of quantized codeword sequences which is the closest sequence to the candidate predictor, the Viterbi algorithm is used. If the closest sequence identified does not match the CRC received, the syndrome decoding process is performed again using another candidate predictor (generated by the motion search). The syndrome decoding process stops when the closest sequence identified matches the CRC received. The CRC is used as a reliable and unique signature for each block and allows identifying the best candidate predictor.
- 3) **Base and Refinement Dequantization:** After the quantized coefficients are reconstructed, the base dequantization is performed. In order to achieve better reconstruction quality, the base dequantization is followed by refinement dequantization using the refinement bits transmitted by the encoder. At this stage of the PRISM decoding process, there are two estimates for the Wyner-Ziv coefficients: the coefficients of the prediction block found in the motion estimation stage and the coefficients obtained through syndrome decoding, base dequantization and refinement dequantization. Employing a linear estimation algorithm, the Wyner-Ziv coefficients final estimate is obtained.
- 4) **Entropy Decoding and Dequantization:** The received pure source coded bits, corresponding to the quantized high-frequency transform coefficients, are decoded using inverse operations to those performed at the encoder: entropy decoding and dequantization.
- 5) **Inverse Scan and Inverse DCT:** After inverse zig-zag scanning of the decoded transform coefficients, the inverse discrete cosine transform (IDCT) is then applied completing the PRISM decoding process.

### 2.2.3.3 Some Experimental Results

In order to evaluate the performance of the proposed system, the authors coded the first 15 frames of *Mother and Daughter* ( $352 \times 288$  luminance samples), *Carphone* ( $176 \times 144$ ) and *Football* ( $352 \times 240$ ). The first frame of each video sequence is fully intra mode encoded, i.e. each block of the frame is encoded in intra mode for both the PRISM and H.263+ coders.

Figure 2.14 shows the rate-distortion performance achieved with the PRISM system and the rate-distortion performance of a H.263+ video coder, when no frames are lost. The results are obtained for the three video sequences mentioned above which have different motion characteristics. As it can be noticed from Figure 2.14, independently of the motion content associated to the video sequence, the PRISM rate-distortion performance is between the inter and the intra coding modes of the H.263+ coder.

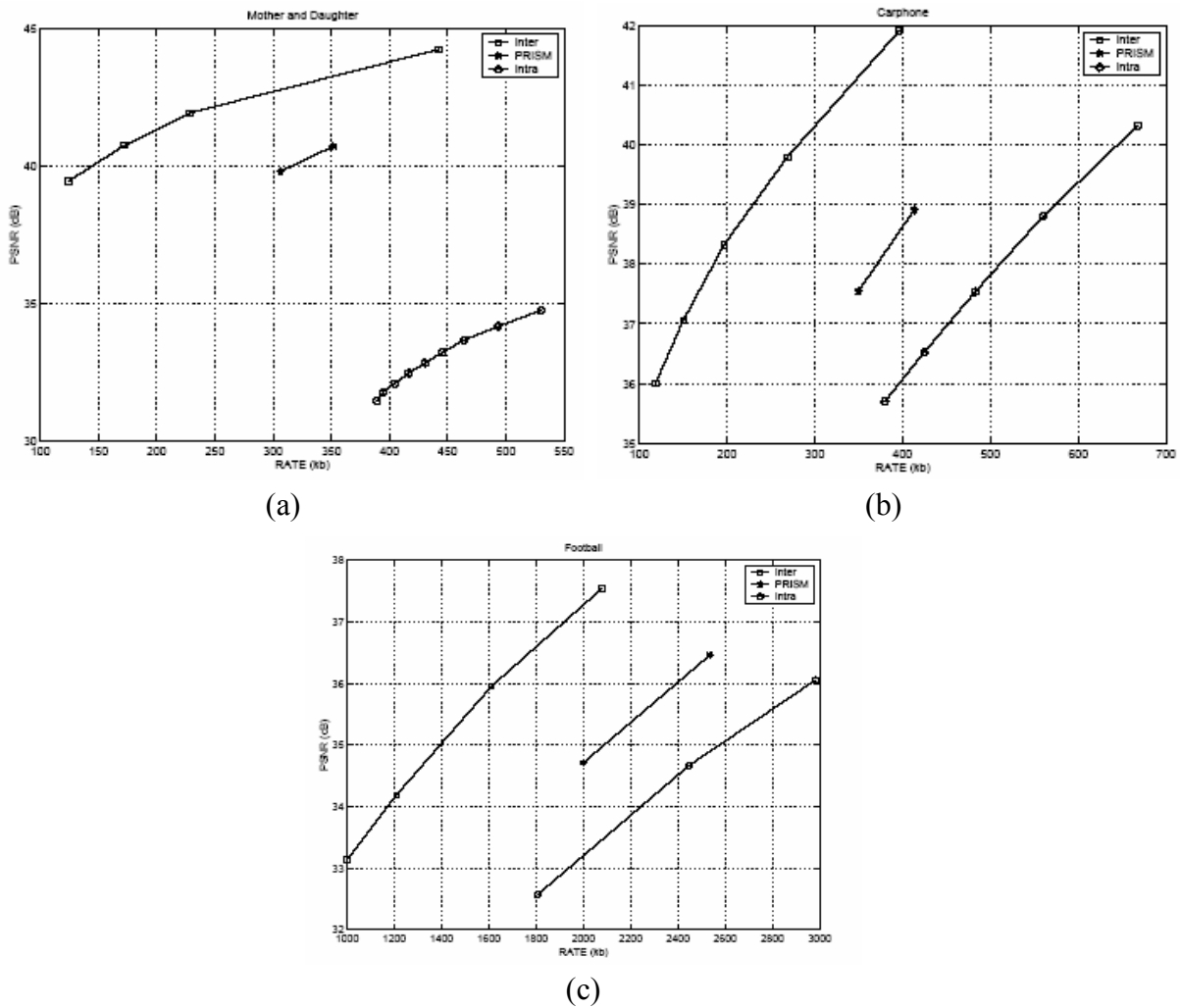


Figure 2.14 – Comparison of the rate-distortion performance for PRISM and H.263+ (both intra and inter coding modes) when no frames are lost [40].

Figure 2.15 shows the impact of one frame loss both for the PRISM and H.263+ coders. The results for the *Football* sequence point out that the robustness of PRISM over H.263+ standard is much higher since annoying visual artefacts due to interframe error propagation are not observed. As can be noticed, the loss of one frame when using PRISM has a negligible effect on the quality of the decoded video, since the error propagation is stopped due to the absent of a prediction loop at the PRISM encoder. However, with the H.263+ coder these artefacts accumulate and propagate to the following frames of the video sequence (observe the evolution of the player with number 57 along both sequences).

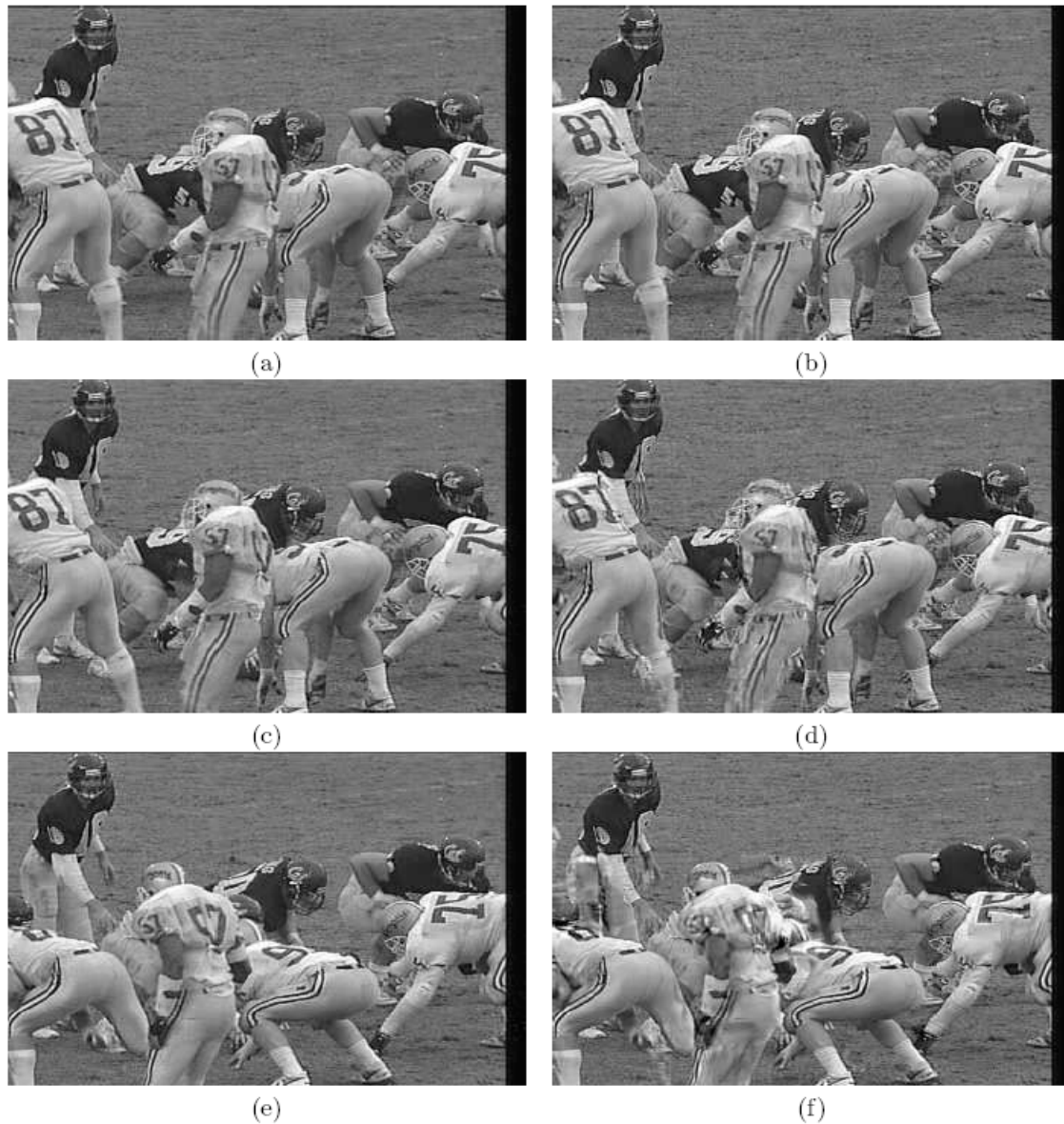


Figure 2.15 - Impact of a frame loss in PRISM (left) and H.263+ (right): each column, from top to bottom, represents the first, third and fourteenth decoded frames for the Football sequence [40].

Hence, the main issue in predictive coding frameworks, this means the drift problem associated to the difference between the prediction frame at the encoder and the prediction frame at the decoder, is fully avoided with the PRISM solution. For more details about the PRISM solution, the reader should consult [40].

### 2.3 Final Remarks

Emerging applications with encoding requirements quite different from those targeted by MPEG-x and H.26x standards, such as low-complexity and low-power consumption at the encoder, have stimulated the development of a new coding paradigm able to satisfy these

requests. This new coding paradigm is built based on distributed source coding which was theoretically studied in the 1970s by Slepian, Wolf, Wyner and Ziv.

A great part of the work that has been performed in distributed video coding refers to Wyner-Ziv video coding – a particular case of distributed video coding which refers to the case where each video frame is encoded independently (intraframe coding), but the same frame is decoded conditionally (i.e. interframe decoding). So far, the results achieved show that Wyner-Ziv video coding can provide interesting coding solutions for some applications where low encoding complexity is the major goal, e.g. multimedia sensor networks. While in traditional video coding, based on hybrid DCT and interframe predictive coding, the most complex encoding operation, the motion estimation task, is performed by the encoder, in the distributed video coding scenario such operation is performed by the decoder. Thus, in Wyner-Ziv video coding, the encoder can exhibit low complexity, similar to traditional intraframe coding, at the expense of higher decoder complexity compared with traditional video coding schemes. On the other hand, the robustness intrinsic to distributed coding, due to channel coding and the absence of prediction loop at the Wyner-Ziv encoder, suggests that a natural application area is joint source-channel coding; in this field, the robustness to channel errors of a bitstream traditionally encoded, either with MPEG-x or H.26x standards, is improved using a Wyner-Ziv encoded bitstream.

The distributed coding overview presented in this Chapter serves as the starting point for chapters 3 and 4, where is described the development and implementation of improved distributed video coding schemes for the pixel and the transform domains. For the development of those distributed video coding solutions, the Stanford Wyner-Ziv Low-Complexity Video Coding Solution (described in Section 2.2.1) is taken as the architectural reference due to its low encoding complexity and versatility in changing from the pixel to the transform domains.

The solution proposed in Chapter 3 (IST-PDWZ) is based on the pixel domain Wyner-Ziv coding architecture described in [17]; the IST-PDWZ solution makes use of a quantizer, a turbo code based Slepian-Wolf codec, a reconstruction module and a frame interpolation module and therefore can be seen as a simplified version of the architecture depicted in Section 2.2.1.

In Chapter 4, the transform domain Wyner-Ziv video codec (IST-TDWZ) proposed is based on the architecture described in [12]. The IST-TDWZ architecture is similar to the IST-PDWZ one; the major difference is that the IST-TDWZ codec includes transform coding to exploit the spatial redundancy within an image. Both solutions show good performance (with significant gains over H.263+ Intra and some gains over the state-of-the-art DVC solutions) and low complexity encoding, providing a possible alternative to traditional video coding for applications where low encoding complexity is a major requirement, e.g. video-camera sensor networks.

## Chapter 3

# IST-Pixel Domain Wyner-Ziv Codec

The Wyner-Ziv video coding – a particular case of Distributed Video Coding (DVC) – is a new video coding paradigm based on two major Information Theory results: Slepian-Wolf and Wyner-Ziv theorems, as was seen in Chapter 2. Generally, this new coding paradigm is characterized by a separate encoding of two correlated sources (for instance, two temporally adjacent frames of a video sequence) and a joint decoding of the sources exploiting the correlation between them. Since the two sources are separately encoded, i.e. associating independent encoders to each of them, independent bitstreams are associated to each of the sources. The decoding procedure of the two encoded sources is performed jointly, exploiting the statistical dependency between the bitstreams.

Although the study of Distributed Source Coding (DSC) dates back to the 1970's, efforts toward practical implementations (feasible solutions) of Wyner-Ziv video coding are recent. The emergence of applications with encoding requirements quite different from those targeted by MPEG-x and H.26x standards (e.g. low-complexity and low-power consumption at the encoder) have stimulated such efforts. In the MPEG-x or H.26x standards, the correlation between two adjacent frames is exploited at the encoder through the complex motion estimation task which leads to a high complexity encoder. Since the exploitation of the correlation between two temporally adjacent (or not even adjacent) frames in the Wyner-Ziv video coding is performed only at the decoder, the encoder can typically exhibit low complexity at the expense of a high decoding complexity. In Chapter 2, the most relevant Wyner-Ziv video coding solutions presented in the literature, e.g. hash-based Wyner-Ziv video coding [19], are described with some detail. The results illustrated in Chapter 2 show that Wyner-Ziv video coding can provide promising coding solutions for some applications where low encoding complexity is a major goal, e.g. multimedia sensor networks.

This Chapter is focused on describing the implementation by the author of this Thesis of a modified version of the approach proposed by Aaron *et al.* in [17]. The solution proposed in that paper constitutes the architectural starting point for some more recent and more advanced solutions by the same authors, like the one proposed in [19]. The solution described in this Chapter will be designated IST-PDWZ from Instituto Superior Técnico-Pixel Domain Wyner-Ziv codec.

### 3.1 IST-Pixel Domain Wyner-Ziv Codec Architecture

Figure 3.1 illustrates the architecture of the IST-PDWZ codec. The general architecture of this solution is similar to the one proposed by Aaron *et al.* in [17]: both make use of a quantizer, a turbo-code based Slepian-Wolf codec, a frame interpolation module and a reconstruction module.

There are however some major differences between the IST-PDWZ solution and the one proposed in [17], namely in the Slepian-Wolf codec and the frame interpolation module. Some of the differences are also motivated by the fact that the codec proposed in [17] is not described with enough detail for all the modules and thus new solutions had to be developed for most of the modules by the author of this Thesis.

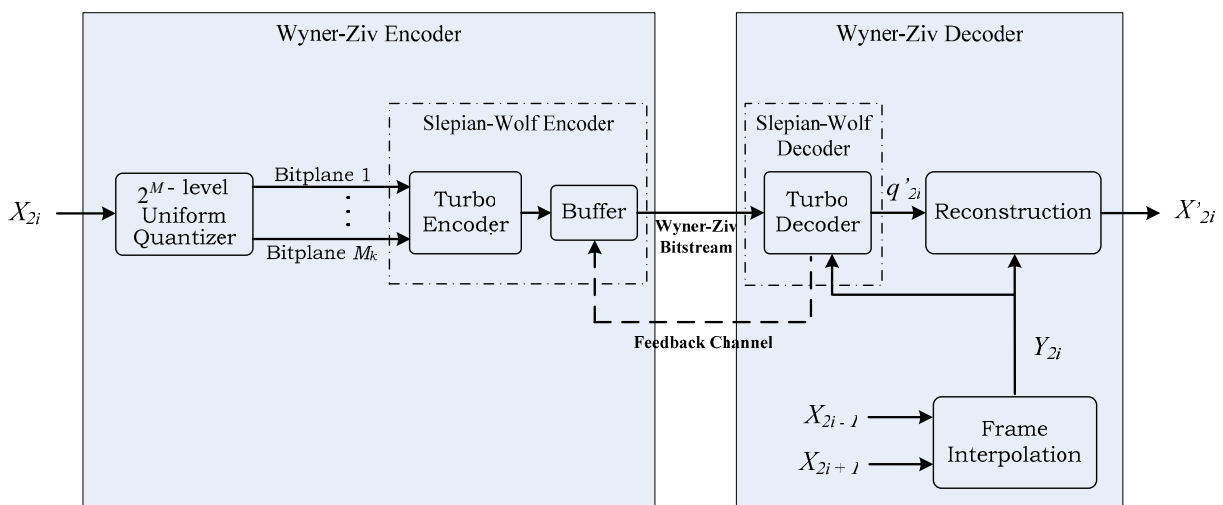


Figure 3.1 – IST-PDWZ codec architecture.

In a nutshell, the coding procedure illustrated in Figure 3.1 is described as follows:

- A video sequence is divided into Wyner-Ziv frames (the even frames of the video sequence) and key frames (the odd frames of the video sequence).
- Each Wyner-Ziv frame of a video sequence,  $X_{2i}$ , is encoded sample by sample, i.e. pixel by pixel.
- The  $X_{2i}$  frame pixels are quantized using a  $2^M$ -level uniform quantizer, generating the quantized symbol stream.



- Over the resulting quantized symbol stream (constituted by all the quantized symbols of  $X_{2i}$  using  $2^M$  levels) bitplane extraction is performed and each bitplane is then independently turbo encoded.
- The redundant (parity) information produced by the turbo encoder for each bitplane is stored in the buffer and transmitted in small amounts upon decoder request via the feedback channel.
- In the IST-PDWZ solution, turbo coding is performed at the bitplane level instead of at the symbol level, as in the approach proposed in [17], since turbo coding at the bitplane level is performed in recently proposed solutions, e.g. [19]. Besides being easier to implement, turbo coding at bitplane level also allows more flexibility in terms of the  $M$  parameter (number of bits required to map a pixel value into one of  $2^M$  quantizer levels) choice. Typically, when the turbo coding operation is performed at symbol level, the  $M$  parameter value has to be a sub-multiple of the turbo encoder input length in order to have an integer number of quantized symbols at the turbo encoder input; each quantized symbol is represented by  $M$  bits when a  $2^M$  levels quantizer is used. In a bitplane level turbo coding scenario, the  $M$  bitplanes are extracted from the quantized symbol stream associated to the whole  $X_{2i}$  frame, as will be described in Section 3.1.1; in the bitplane extraction procedure, each quantized symbol contributes with only one bit for each bitplane (for instance, the most significant bitplane only encloses the most significant bit of all the  $X_{2i}$  frame quantized symbols). Since, independently of  $M$ , only one bit of each quantized symbol is enclosed by a given bitplane, the  $M$  parameter is not therefore restricted to any particularly set of values. The flexibility in the  $M$  value choice is essential to solutions where transform coding is used since any value of  $M$  may be used.

Since the IST-PDWZ solution performs turbo coding at the bitplane level, the IST-PDWZ Rate-Distortion (RD) performance and the performance of the solution proposed by Aaron *et al.* in [17] cannot be directly compared. However, the IST-PDWZ RD performance may be compared with the pixel domain results published by Aaron *et al.* in [12], since those results were obtained with an architecture similar to the one proposed in [17] but performing turbo coding at the bitplane level instead of at the symbol level (for more details the reader should consult [12]). From now on, the IST-PDWZ RD performance will be compared with the pixel domain results published in [12].

- At the decoder, the frame interpolation module is used to generate an estimate of the  $X_{2i}$  frame, called  $Y_{2i}$ , based on two temporally adjacent frames of  $X_{2i}$  (represented by  $X_{2i-1}$  and  $X_{2i+1}$  in Figure 3.1); this estimate is then used by the turbo decoder to obtain the decoded quantized symbol stream  $q'_{2i}$ .
- The  $Y_{2i}$  frame, known as side information, is also used in the reconstruction module, together with the  $q'_{2i}$  stream, to help in the  $X_{2i}$  reconstruction task.

Since  $X_{2i}$  is pixel by pixel encoded, the solution illustrated by Figure 3.1 is named IST pixel domain Wyner-Ziv (IST-PDWZ) codec. In the following sections, the architecture and implementation of each IST-PDWZ module depicted in Figure 3.1 is described in detail.

### 3.1.1 Quantizer in the IST-PDWZ Codec

The first step towards encoding a Wyner-Ziv frame  $X_{2i}$  in the IST-PDWZ architecture, depicted in Figure 3.1, is quantization.

- The pixels of each Wyner-Ziv frame of a video sequence,  $X_{2i}$ , are quantized using a uniform scalar quantizer with  $2^M$  levels; the parameter  $M$  corresponds to the number of bits needed to map a pixel value into one of the  $2^M$  quantizer levels.
- In the IST-PDWZ codec,  $M$  can assume any integer value between 1 and 8 corresponding to a number of quantizer levels ranging from 2 to 256. Varying the  $M$  value, different RD performances can be reached since each  $M$  value has a given rate-distortion point associated. Four rate-distortion points were considered in the IST-PDWZ codec performance evaluation notably in order to allow comparing the RD performance of the IST-PDWZ codec and the pixel domain RD results achieved in [12]; those rate-distortion points correspond to  $M$  values of 1, 2, 3 and 4.
- After quantizing the pixels of the  $X_{2i}$  frame, the quantized symbols (represented by integer values) are converted into a binary stream. The quantized symbols bits of the same importance (e.g. the most significant bit) are grouped together forming the corresponding bitplane array. Since the formation of the bitplane array is performed considering the overall quantized symbol stream, the length of each bitplane array is the frame size,  $N \times M$ .
- Each bitplane is then independently turbo encoded, starting with the most significant bitplane array, which corresponds to the most significant bits of the  $X_{2i}$  frame quantized symbols.

### 3.1.2 Slepian-Wolf Encoder in the IST-PDWZ Codec

After quantizing the  $X_{2i}$  frame pixels and forming the  $M$  bitplane arrays associated to the whole image quantized symbols, each bitplane array is then fed into the Slepian-Wolf encoder, starting with the most significant bitplane array. As Figure 3.1 shows, the Slepian-Wolf encoder typically comprises a turbo encoder and a buffer:

- The turbo encoder produces a sequence of parity bits (redundant bits related to the initial data) for each bitplane array; the amount of parity bits produced for each bitplane depends on the turbo encoder rate, i.e. on the ratio of turbo encoder output bits (parity bits) per turbo encoder input bit.
- The parity bits produced by the turbo encoder are then stored in the buffer, punctured, according to a given puncturing pattern, and transmitted upon decoder request via the feedback channel.

Figure 3.2 illustrates a turbo encoder structure using a parallel concatenation of two identical constituent Recursive Systematic Convolutional (RSC) encoders, as proposed in [46]; in between the RSCs, a random  $L$ -bit interleaver is employed to decorrelate the  $L$ -bit input

sequence ( $\mathbf{X}$ ) between the two RSC encoders [46]. In Figure 3.2, the  $\mathbf{S}_i$  and  $\mathbf{P}_i$  ( $i = 1, 2$ ) symbols represent the sequences produced by the RSC <sub>$i$</sub>  encoder.

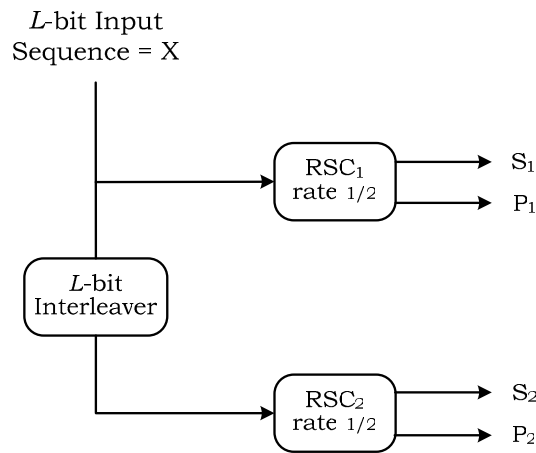


Figure 3.2 – Turbo encoder structure using a parallel concatenation of two identical constituent Recursive Systematic Convolutional encoders (RSC<sub>1</sub> and RSC<sub>2</sub>).

### L-bit Interleaver

One of the turbo encoder architectural modules is the interleaver:

- Generically, the interleaver output sequence is its input sequence rearranged in different order accordingly to a given pattern, the interleaving structure; feeding the interleaver output sequence into a deinterleaver, the order of interleaver input sequence is restored. Figure 3.3 illustrates the interleaving of a 10-bit length input sequence, for sake of simplicity; the input bits are represented in Figure 3.3 by the symbols  $a_i \in \{0, 1\}$  with  $1 \leq i \leq 10$ .

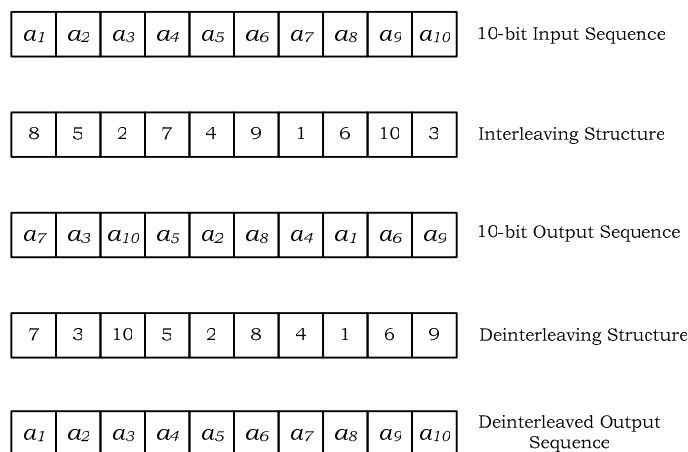


Figure 3.3 – Interleaving and deinterleaving of a 10-bit sequence.

For a 10-bit length input sequence, the interleaving structure is illustrated in Figure 3.3 with an array of numbers, from 1 to 10, with no repetitions. The interleaving structure

maps each index of the input sequence into one index of the output sequence, in a univocal correspondence. According to the interleaving structure depicted in Figure 3.3, the input sequence bit located in the 1<sup>st</sup> position ( $a_1$ ) will be inserted in the 8<sup>th</sup> position of the output sequence; the 2<sup>nd</sup> input bit ( $a_2$ ) will be inserted in the 5<sup>th</sup> position of the output sequence and so on.

- The deinterleaving structure can be obtained from the interleaving structure according to

$$\text{deinterleaver}[\text{interleaver}[j]] = j \quad (3.1)$$

where the interleaving and deinterleaving structures are treated as arrays (interleaver and deinterleaver, respectively); the value of the deinterleaver array in the position (interleaver[  $j$  ]) is equal to  $j$ . The deinterleaving structure “acts” over the interleaved sequence (represented in Figure 3.3 by the 10-bit output sequence) in a similar way to the interleaving structure, restoring the original order of the bits in the 10-bit input sequence (see Figure 3.3).

- There are several types of interleaving structures for turbo codes, such as random interleavers, block interleavers, convolutional interleavers [8]; different interleavers may imply different turbo coding performances [8]. In the IST-PDWZ codec, a pseudo-random interleaver was adopted in order to allow comparing the IST-PDWZ performance with the pixel domain results published in [12]. Note that the  $L$ -bit input sequence cannot be interleaved in a completely random fashion in order to make possible the turbo decoding. That is, the turbo decoder must know the interleaving pattern used by the turbo encoder to be able to perform the decoding task.
- The decorrelation process applied to the turbo encoder input information is important in the context of turbo coding since the performance of turbo coding depends on the randomness between the parity sequences at the output of the two RSC encoders [47].
- The interleaver length  $L$  also affects the turbo coding performance [8]. As shown by Shannon [48], random codes with large block sizes may achieve a transmission rate close to the channel capacity (amount of bits that can be reliably transmitted, i.e. with a bit error probability near zero, over the channel). The interleaver length  $L$  value (more generically, the turbo encoder input length) must therefore be chosen quite large, for instance 1000, since low values of  $L$  may imply a lack of the randomness needed to reach a good performance for the turbo codes.

### **RSC Encoder**

The turbo encoder architecture depicted in Figure 3.2 encloses, beside the interleaver, two recursive systematic convolutional (RSC) encoders. A RSC encoder is typically characterized by a generator matrix  $G$  which allows to obtain the RSC encoder output for a given RSC encoder input; generically, the RSC encoder output  $\text{RSC}_{\text{out}}$  is generated through the RSC encoder input  $\text{RSC}_{\text{in}}$  and the generator matrix  $G$  accordingly to (3.2).

$$\text{RSC}_{\text{out}} = \text{RSC}_{\text{in}} \cdot G \quad (3.2)$$

The rate of the RSC encoder may be expressed by the ratio

$$\frac{\text{number of RSC encoder input bits}}{\text{number of RSC encoder output bits}} \quad (3.3)$$

In the IST-PDWZ solution, RSC encoders of rate  $\frac{1}{2}$  were used since RSC encoders of rate  $\frac{1}{2}$  are employed in recent, more advanced solutions, e.g. in [19]. Each constituent recursive systematic convolutional encoder of rate  $\frac{1}{2}$  enclosed by the turbo encoder may be represented by a generator matrix of the form

$$\begin{bmatrix} 1 & g_2(D) \\ & g_1(D) \end{bmatrix} \quad (3.4)$$

where  $g_1(D)$  and  $g_2(D)$  are two polynomials in  $D$  ( $D$  denotes delay);  $g_1(D)$  and  $g_2(D)$  may be generically expressed by (3.5), for  $i = 1$  and  $i = 2$ , respectively.

$$g_i(D) = g_{i0} + g_{i1} \times D + g_{i2} \times D^2 + \dots + g_{im} \times D^m \quad i = 1, 2 \quad (3.5)$$

The polynomials degree  $m$  indicates the memory of the RSC encoder, i.e. the number of  $D$  elements (shift registers) used in the implementation of the RSC encoder. The coefficients  $g_{ik}$  ( $i = 1, 2$  and  $k = 1, 2, \dots, m$ ) may assume the values 1 or 0 indicating if the value of the shift register  $D^k$  is or is not taken into account in the  $\text{RSC}_i$  encoder output generation. Since only two values are possible for the  $g_{ik}$  coefficients, the number of states of the RSC code is  $2^m$ .

Figure 3.4 illustrates a rate  $\frac{1}{2}$  constituent recursive systematic convolutional encoder with memory  $m = 4$  (16 states) and with a generator matrix given by equation (3.6).

$$\begin{bmatrix} 1 & \frac{1 + D + D^3 + D^4}{1 + D^3 + D^4} \end{bmatrix} \quad (3.6)$$

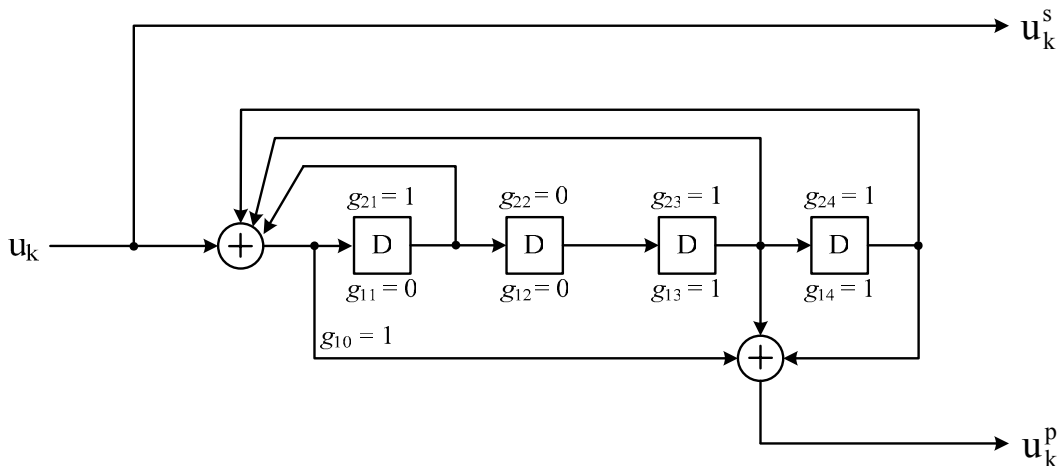


Figure 3.4 – Rate  $\frac{1}{2}$  (one input, two output) constituent recursive systematic convolutional encoder with memory 4 (16 states) and generator matrix given by equation (3.6).

According to (3.3), the RSC encoder is of rate  $\frac{1}{2}$  since for each input bit (represented in Figure 3.4 by the  $u_k$  symbol) there are two output bits ( $u_k^s$  and  $u_k^p$ ).

In fact, in Figure 3.4, the symbol  $u_k$  represents the  $k^{\text{th}}$  bit (or the bit at time  $k$ ) of the  $L$ -bit sequence at the input of the RSC encoder. The symbols  $u_k^s$  and  $u_k^p$  represent the two outputs of the RSC encoder at time  $k$ : the systematic bit and the parity bit, respectively; in this context, the term time refers to the position within an array ( $L$ -bit sequence). The systematic bit  $u_k^s$  is a copy of the input bit  $u_k$ ; since one of the encoder's outputs is a copy of the input bit, the constituent convolutional encoder is called systematic. The parity bit at time  $k$ ,  $u_k^p$ , for a 16-state RSC code is generically given by

$$u_k^p = g_{10} + (s_1^{k-1} \cdot g_{11} + s_2^{k-1} \cdot g_{12} + s_3^{k-1} \cdot g_{13} + s_4^{k-1} \cdot g_{14}) \quad (3.7)$$

where  $g_{10} = u_k + (s_1^{k-1} \cdot g_{21} + s_2^{k-1} \cdot g_{22} + s_3^{k-1} \cdot g_{23} + s_4^{k-1} \cdot g_{24})$  and the operator  $(\cdot)$  is equivalent to the *exclusive* OR operator ( $1.1 = 0.0 = 0$  and  $0.1 = 1.0 = 1$ ). The coefficients  $s_i^{k-1}$  ( $i = 1, 2, 3, 4$ ) correspond, in Figure 3.4, to the shift registers ( $D$ ) values (0 or 1) from left to right before introducing the  $k^{\text{th}}$  input bit into the RSC encoder, i.e. the values of the shift registers  $D$  at time  $k-1$ . The set of values  $(s_1, s_2, s_3, s_4)^{k-1}$  represents the RSC encoder state  $S$  at time  $(k-1)$ , i.e.  $S_{k-1} = (s_1, s_2, s_3, s_4)^{k-1}$ . After introducing the  $k^{\text{th}}$  input bit, the shift registers values change in accordance to the input bit originating a RSC encoder's state transition from  $S_{k-1}$  to  $S_k$  (the RSC code state at time  $k$ ).

For each RSC encoder state  $S_{k-1}$ , there are two possible state transitions when the  $u_k$  bit is shifted into the RSC encoder; each one of the two state transitions is associated to a value of the  $u_k$  bit (value 0 or 1). Besides the input bit  $u_k$ , each one of the two state transitions is also characterized by the RSC encoder output bit (parity bit)  $u_k^p$ .

It is possible to construct a diagram that shows, for each RSC encoder state, the possible state transitions as well as the RSC output sequence (constituted by the  $u_k^s$  and  $u_k^p$  bits) given an RSC encoder input sequence (represented in Figure 3.4 by the  $u_k$  bit); this diagram is often called trellis diagram [8]. Figure 3.5 illustrates the trellis diagram of a RSC encoder with a generator matrix given by equation (3.6) when the RSC encoder initial state is 0000; since the polynomial degree in (3.6) is  $m = 4$ , there are  $2^4 = 16$  RSC encoder states.

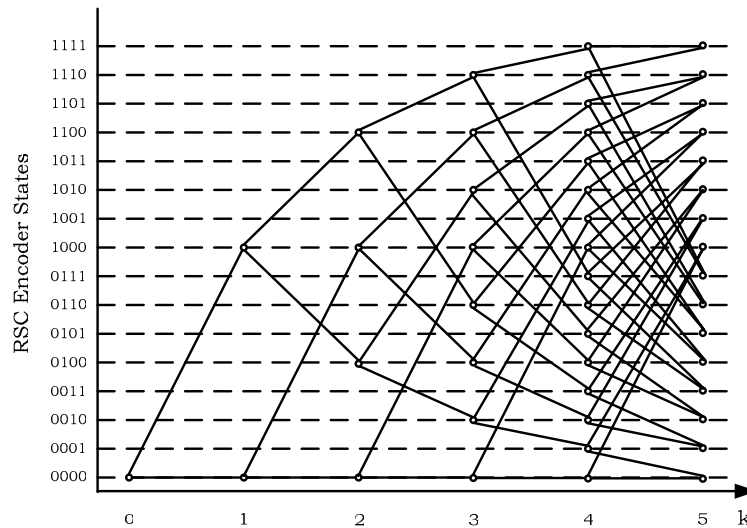


Figure 3.5 – Trellis diagram of a RSC encoder with a generator matrix given by (3.6).

Shifting the first bit of the  $L$ -bit input sequence,  $u_1$ , into the RSC encoder, two state transitions are possible: the state transition associated to  $u_1 = 0$  and the state transition associated to  $u_1 = 1$ ; in Figure 3.5, each state is represented by a circle and each state transition is represented by a line connecting two circles (RSC states). To each  $u_1$  bit value (0, 1) corresponds also an RSC output sequence formed by the  $u_k^s$  and  $u_k^p$  bits (see Figure 3.4). Shifting the  $u_2$  bit into the RSC encoder, two state transitions are possible for each state resulting from shifting  $u_1$  into the RSC encoder, and so on. In order to avoid Figure 3.5 becoming too dense, Table 3.1 contains, for each RSC state, the bits  $u_k$  shifted into the RSC encoder and the corresponding RSC output bits,  $u_k^s$  and  $u_k^p$  (see Figure 3.4).

Table 3.1 – Possible RSC encoder state transitions,  $S_{k-1} \rightarrow S_k$ , and output bits ( $u_k^s$ ,  $u_k^p$ ) given the RSC encoder input bit  $u_k$ .

State $S_{k-1}$	$u_k$	$u_k^s$	$u_k^p$	States $S_k$
<b>0000</b>	0/1	0/1	0/1	0000/1000
<b>0001</b>	0/1	0/1	0/1	1000/0000
<b>0010</b>	0/1	0/1	0/1	1001/0001
<b>0011</b>	0/1	0/1	0/1	0001/1001
<b>0100</b>	0/1	0/1	0/1	0010/1010
<b>0101</b>	0/1	0/1	0/1	1010/0010
<b>0110</b>	0/1	0/1	0/1	1011/0011
<b>0111</b>	0/1	0/1	0/1	0011/1011
<b>1000</b>	0/1	0/1	1/0	1100/0100
<b>1001</b>	0/1	0/1	1/0	0100/1100
<b>1010</b>	0/1	0/1	1/0	0101/1101
<b>1011</b>	0/1	0/1	1/0	1101/0101
<b>1100</b>	0/1	0/1	1/0	1110/0110
<b>1101</b>	0/1	0/1	1/0	0110/1110
<b>1110</b>	0/1	0/1	1/0	0111/1111
<b>1111</b>	0/1	0/1	1/0	1111/0111

The RSC encoder trellis diagram may be useful in the turbo decoding procedure, as it will be seen in Section 3.1.4.

After this brief and generic description of a turbo encoder implementation, consider again Figure 3.1, namely the Slepian-Wolf encoder module:

- Each one of the two RSC encoders enclosed by the turbo encoder computes a parity sequence corresponding to the  $L$ -bit sequence at its input. For the  $RSC_1$  encoder, the  $L$ -bit input sequence is a bitplane extracted from the quantized symbol stream (see Section 3.1.1); for the  $RSC_2$  encoder, the  $L$ -bit input sequence is a pseudo-randomly interleaved version of the bitplane (see Figure 3.2).

Since the convolutional encoders ( $RSC_1$  and  $RSC_2$ ) are systematic, the  $L$ -bit sequence at the input of each RSC encoder is “copied” to the RSC encoder output. This RSC output sequence is known as the systematic sequence and is represented in Figure 3.2 by  $S_1$  and  $S_2$  (for the  $RSC_1$  and the  $RSC_2$  encoders, respectively).

- After turbo encoding a bitplane, the systematic sequences  $S_1$  and  $S_2$  are discarded, as in [12] (in practise, this is the information estimated with the side information at the decoder); on the contrary, the parity sequences,  $P_1$  and  $P_2$ , are stored in the buffer.
- In order to reproduce the results in [12], the rate  $\frac{1}{2}$  constituent recursive systematic convolutional encoders,  $RSC_1$  and  $RSC_2$ , used in the IST-PDWZ codec implementation have a generator matrix given by (3.6). It is also assumed that the two RSC encoders (see Figure 3.2) start at state 0,  $S_0 = 0000$ , but the final state  $S_L$  is considered unknown.

### **Rate Compatible Punctured Turbo (RCPT) Code-Based Slepian-Wolf Codec**

In order to transmit the minimum amount of parity bits needed to successfully decode the quantized symbol stream, the Slepian-Wolf codec is built based on a Rate Compatible Punctured Turbo (RCPT) code structure [45], as in [12]:

- Generically, in a RCPT code structure, after turbo encoding a  $L$ -bit input sequence, the parity sequence generated by each RSC encoder is divided into  $P$  blocks of  $(L/P)$  bits each, where  $L$  is the interleaver length and  $P$  is often called the puncturing period.

Figure 3.6 and Figure 3.7 illustrate a 16-bit parity sequence division process ( $L = 16$ ) considering a puncturing period of 4, for simplicity; the numbers in Figure 3.6 and Figure 3.7 just indicate the position of each parity bit.

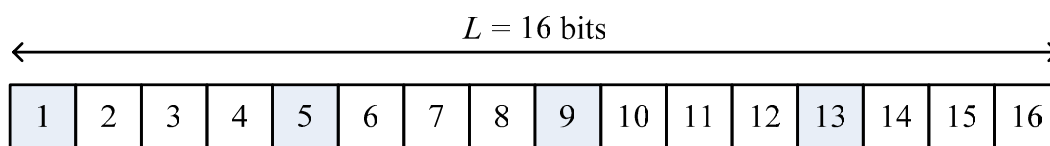


Figure 3.6 – Parity sequence of 16 bit length at the output of a RSC encoder.



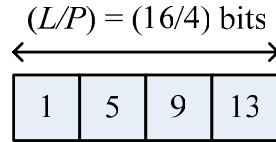


Figure 3.7 – Possible composition of the first block of  $(16/4) = 4$  bits obtained by division of the sequence illustrated in Figure 3.6 into  $P = 4$  blocks.

The first block of  $(16/4)$  bits can, for instance, be formed by the parity bits located at positions 1, 5, 9 and 13 (or in general at positions 1,  $P+1$ ,  $2P+1$ , etc); the bits at positions 2, 6, 10 and 14 (or 2,  $P+2$ ,  $2P+2$ , etc) may constitute the second block of  $(16/4)$  bits and so on. When the decoder requests for more parity bits via the feedback channel, the Wyner-Ziv encoder only transmits a block of  $(16/4)$  bits to the decoder; of course, no block is sent twice. Since, when the decoder requests for more parity bits, only a fraction of the total number of parity bits  $L$  is sent to the decoder, corresponding to a  $(L/P)$ -bit block, puncturing is performed over the parity sequence.

- The RCPT code structure together with the feedback channel present in the IST-PDWZ codec (see Figure 3.1) allows adapting the Wyner-Ziv bitrate according to the similarity between the frame  $X_{2i}$  and the corresponding side information  $Y_{2i}$ . That is, the higher the similarity between the frame to code and its estimation made at the decoder, the fewer the parity bits (Wyner-Ziv bits) that need to be sent to the decoder to reach a certain quality (or to successfully decode the quantized symbol stream).

### **Puncturing Pattern**

The puncturing pattern, i.e. the order by which the blocks of  $(L/P)$  bits (both of  $\mathbf{P}_1$  and  $\mathbf{P}_2$ ) are transmitted to the decoder, is not specified in [12]. Due to the lack of details regarding the puncturing pattern used to obtain the pixel domain results in [12], a pseudo-random puncturing pattern was developed for the IST-PDWZ solution which allows achieving good RD performance, as will be seen in Section 3.2. Thus, within the parity sequence  $\mathbf{P}_i$  ( $i = 1, 2$ ), the blocks transmission is performed in a pseudo-random fashion; the blocks are transmitted alternately from  $\mathbf{P}_1$  and  $\mathbf{P}_2$ .

The puncturing pattern structure is described in the following:

- The Wyner-Ziv encoder starts by sending one  $(L/P)$ -bit block of the parity sequence  $\mathbf{P}_1$ ; no bits are sent from the  $\text{RSC}_2$  parity sequence ( $\mathbf{P}_2$ ). If the decoder requests for more parity bits via the feedback channel, then the encoder transmits one  $(L/P)$ -bit block of  $\mathbf{P}_2$ . If one more parity bits request is made via the feedback channel, the Wyner-Ziv encoder sends another block of  $(L/P)$  bits from  $\mathbf{P}_1$  and so on until no more parity bits requests are made or all the parity bits (from both  $\mathbf{P}_1$  and  $\mathbf{P}_2$ ) have been transmitted.
- In the IST-PDWZ codec, the puncturing period  $P$  ranges from 1 to 32, starting with  $P = 32$ , since this range allows to achieve similar or better PSNR results regarding the pixel

domain results published in [12] (which are the best available in the literature for this type of architecture).

### **3.1.3 Frame Interpolation in the IST-PDWZ Codec**

In Section 3.1, it was mentioned that the Wyner-Ziv decoder requires the side information to be available to reconstruct  $X_{2i}$ . But how is this side information generated at the decoder? Consider Figure 3.1 and assume that the odd frames of a video sequence  $X_{2i-1}$  and  $X_{2i+1}$  (called key frames) are available at the decoder without any compression. There are several techniques that can be employed at the Wyner-Ziv decoder to generate the side information,  $Y_{2i}$ .

The simplest frame interpolation techniques that can be used are to make  $Y_{2i}$  equal to the  $X_{2i}$  previous temporally adjacent frame, illustrated in Figure 3.1 by  $X_{2i-1}$ , – which means to assume that there is no temporal variation – or to perform bilinear (average) interpolation between the key frames  $X_{2i-1}$  and  $X_{2i+1}$ . However, if this technique is employed to generate the side information in video sequences where the similarity between two temporally adjacent frames is low (or even not so low),  $Y_{2i}$  will probably be a rough estimate of  $X_{2i}$ . In this case, it is straightforward that the decoder will need to request more parity bits from the encoder to decode the  $q'_{2i}$  stream when compared to the case where  $Y_{2i}$  is a closer estimate of the  $X_{2i}$  frame. Thus the Wyner-Ziv bitrate will increase for the same PSNR; the Wyner-Ziv bitrate refers to the number of Wyner-Ziv bits (parity bits) needed to decode the  $q'_{2i}$  stream. Subjectively, these simple frame interpolation techniques will introduce in the decoded frame  $X'_{2i}$  artefacts such as “jerkiness” and “ghosting”, especially for low bitrates.

More refined and complex techniques, e.g. techniques based on the motion estimation of the video sequence, are therefore essential to construct high quality side information (close to the original) when the similarity between adjacent frames is low. With more sophisticated techniques than the simple copy of the previous frame, it is possible in those situations to obtain a side information frame more similar to the  $X_{2i}$  frame and thus minimize the Wyner-Ziv bitrate for the same decoded frame  $X'_{2i}$  quality. The accuracy of the frame interpolation module (see Figure 3.1) is therefore a key feature for the IST-PDWZ codec rate-distortion performance.

The motion estimation techniques, used in traditional video coding at the encoder, attempt to choose the best prediction for the current frame in the rate-distortion sense; in other words, for a given block in the current frame, the motion estimation techniques attempt to find the best match in the reference frame independently of the true motion of the block in the scene. For frame interpolation, these techniques are not well-suited since in this case the current frame is not known and it is necessary to estimate the true motion to correctly interpolate the missing frame (current frame), usually by motion compensation between temporally adjacent frames.

Figure 3.8 shows the architecture proposed for the frame interpolation scheme. Besides the low pass filter and the motion compensation modules which are always used, the three modules in the middle are associated to increasingly more powerful motion estimation solutions when 1, 2 or 3 modules are used (always starting from the first module on the left, this means the forward

motion estimation module). In the following, all the frame interpolation framework modules are described in detail.

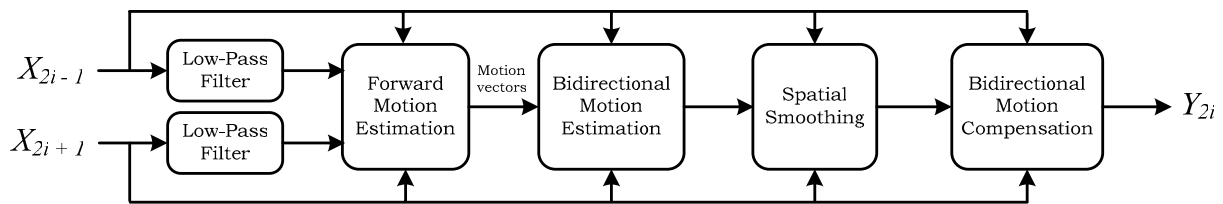


Figure 3.8 – Frame interpolation framework.

### 3.1.3.1 Forward Motion Estimation

The first step to obtain the interpolated frame  $Y_{2i}$  is low-pass filtering.

- Both key frames are low-pass filtered in order to improve the motion vectors reliability and therefore help to estimate motion vectors that are spatially more correlated (or smoother).
- A block matching algorithm is then employed to estimate the motion between the next  $X_{2i+1}$  and previous  $X_{2i-1}$  temporally adjacent key frames. The block-based motion estimation algorithm was chosen due to its low complexity comparing to other algorithms, e.g. dense motion vector fields or motion segmentation with arbitrary regions of support.
- This motion estimation algorithm is characterized by the window size, search range and step size parameters. The window size is the dimension of the square block used as basic unit to perform motion estimation and is fixed for the entire frame. The search range parameter defines the  $X_{2i-1}$  area dimension in which the block most similar to the current block in the  $X_{2i+1}$  frame is searched for. The step size is the distance between pixels in the previous key frame  $X_{2i-1}$  a motion vector is searched for; this parameter enables to reduce the computational complexity of the motion estimation scheme (by increasing the step size) and to provide only a coarse approximation of the true motion field.
- This rigid block-based motion estimation algorithm fails, however, to capture all aspects of the motion field (e.g. occluded areas); also, if frame interpolation is performed, overlapped and uncovered areas will appear in the interpolated frame. This is because the motion vectors obtained do not necessarily intercept the interpolated frame at the center of each non-overlapping block in the interpolated frame. Figure 3.9 illustrates the problem where pixels between two neighboring blocks in the interpolated frame are not interpolated (filled with texture).

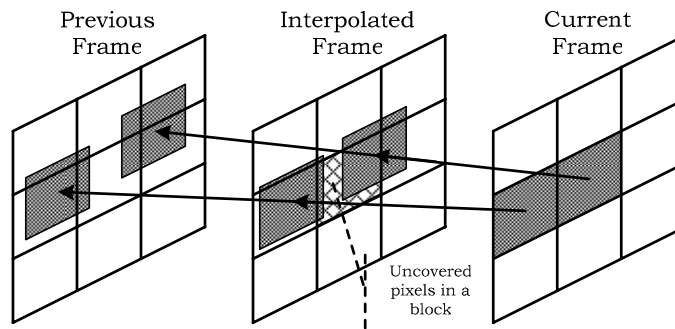


Figure 3.9 – Uncover pixels in the interpolated frame due to the rigid block-based motion estimation algorithm.

- In order to solve this problem, one possible technique is presented in the following: all the motion vectors obtained in the previous stage serve as candidates for each non-overlapping block in the interpolated frame  $Y_{2i}$ . Figure 3.10 illustrates the motion vector selection for a given block of the  $Y_{2i}$  frame: for each block of the interpolated frame  $Y_{2i}$  and from all the candidate vectors, the Motion Vector (MV) that intercepts the interpolated frame  $Y_{2i}$  closer to the center of block under consideration is selected.

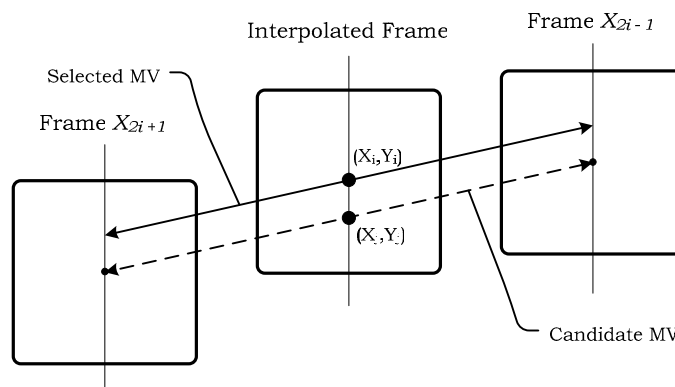


Figure 3.10 – Selection of the motion vector.

- Now that each block in the interpolated frame  $Y_{2i}$  has a motion vector associated to it, bidirectional motion compensation can be performed to obtain the interpolated frame; if a more accurate interpolated frame is desired, further processing may be performed in the next frame interpolation framework module.

### 3.1.3.2 Bidirectional Motion Estimation

The forward motion estimation procedure is followed by the bidirectional motion estimation operation, as illustrated in Figure 3.8.

- A bidirectional motion estimation algorithm, similar to the B-frames coding mode used in current video standards [2], is employed to refine the motion vectors obtained in the forward motion estimation procedure.

- In the Figure 3.1 coding scenario, the interpolated pixels, i.e. the  $Y_{2i}$  pixels, are not known, as opposite to what happens in the B-frames coding mode; so, a different motion estimation technique is therefore used in the IST-PDWZ coding architecture. As Figure 3.11 shows, the adopted bidirectional motion estimation technique selects a linear trajectory between the next and previous key frames,  $X_{2i+1}$  and  $X_{2i-1}$ , respectively, passing at the center of the blocks in the interpolated frame.

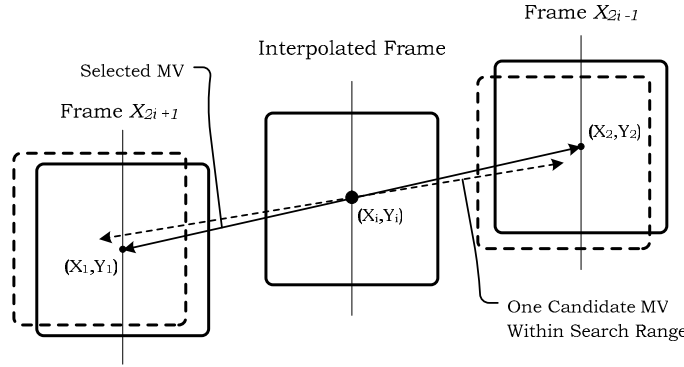


Figure 3.11 – Bidirectional motion estimation.

- The search range is confined to a small displacement around the initial block position and the motion vectors between the interpolated frame and previous and next key frames are symmetric, that is

$$(x_1, y_1) = (x_i, y_i) + MV(B_i) \quad (3.8)$$

$$(x_2, y_2) = (x_i, y_i) - MV(B_i) \quad (3.9)$$

where  $(x_1, y_1)$  are the coordinates of the block in the  $X_{2i-1}$  frame,  $(x_2, y_2)$  are the coordinates of the block in the  $X_{2i+1}$  frame and  $MV(B_i)$  represents the motion vector obtained in the forward motion estimation stage divided by half, since the interpolated frame  $Y_{2i}$  is equally distant to both key frames,  $X_{2i-1}$  and  $X_{2i+1}$ .

- Finally, by taking into account the constraint defined by equations (3.8) and (3.9) bidirectional motion estimation is performed between frames  $X_{2i+1}$  and  $X_{2i-1}$ . The refined motion vectors obtained represent more accurate motion trajectories and some of the errors introduced by the initial motion vector selection operation are corrected.

### 3.1.3.3 Spatial Motion Smoothing Based Estimation

The motion vectors obtained through bidirectional motion estimation may sometimes present low spatial coherence. In order to achieve higher motion field spatial coherence, spatial smoothing algorithms targeting the reduction of the number of false motion vectors, i.e. incorrect motion vectors when compared to the true motion field, may be employed.

- The spatial motion smoothing algorithm employed in the IST-PDWZ codec uses weighted vector median filters [49], commonly applied for noise removal in multi-channel images, since all the components (or channels) of the noisy image are to be taken into consideration. The problem here is similar, since in both cases the main goal is to find the median vector of a set of vectors. While in multi-channel images the vectors correspond to all the components (or channels, e.g. RGB) of the pixel values of the noisy image, here the vectors correspond to both set of coordinates ( $MV_x$  and  $MV_y$ ) of the motion vectors.
- The weighted median vector filter maintains the motion field spatial coherence by looking, at each block, for candidate motion vectors from neighbouring blocks.
- The weighted median vector filter is adjustable by a set of weights which control the filter smoothing strength (or spatial homogeneity of the resulting motion field) depending on the Mean Square Error (MSE) prediction of the block for each candidate motion vector (the MSE is calculated between key frames since the interpolated frame is not available at this moment).
- The spatial motion smoothing algorithm used in the IST-PDWZ codec is both effective at the image boundaries, where abrupt changes of the motion vectors direction occur, as well as in homogenous regions (regions with similar motion) where the outliers are effectively removed. In Figure 3.12, a comparison between an interpolated frame obtained with and without spatial motion smoothing is presented. As Figure 3.12(b) shows, in the region around the nose the motion interpolation fails, since the motion vectors only minimize the MSE between the key frames but do not represent the true motion vector field. The motion interpolation failure in the nose region (Figure 3.12 (a)) can be corrected using a spatial motion smoothing algorithm (Figure 3.12 (a)) which takes into account neighbouring motion vectors to obtain a smoother (without outliers) motion field.



Figure 3.12 – Frame #7 of the Foreman QCIF sequence: (a) with and (b) without spatial motion smoothing.

- The weighted median vector filter proposed is defined as in [49]:

$$\sum_{j=1}^N w_j \|x_{wvmf} - x_j\|_L \leq \sum_{j=1}^N w_j \|x_i - x_j\|_L \quad (3.10)$$

where  $x_1, \dots, x_N$  are the motion vectors of the current block in the previously interpolated frame and the corresponding nearest neighboring blocks; Figure 3.13 illustrates the neighboring blocks  $N_1, \dots, N_8$  for the current block  $B$ .

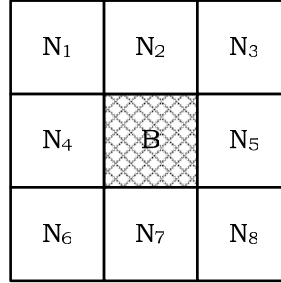


Figure 3.13 – Neighboring blocks of block  $B$  for weighted median vector filter.

In (3.10), the  $w_1, \dots, w_N$  correspond to a set of adaptively-varying weights and  $x_{wvmf}$  represents the motion vector output of the weighted vector median filter: the spatial-temporally smoothed motion vector. The vector  $x_{wvmf}$  is found by searching among all motion vectors the one that minimizes the sum of distances to the other  $N-1$  vectors, in terms of the L-norm.

- The choice of weights is performed according to the prediction error as defined by:

$$w_j = \frac{MSE(x_c, B)}{MSE(x_j, B)} \quad (3.11)$$

where  $x_c$  represents the candidate vector for the current block  $B$  to be smoothed. The MSE (mean square error) represents the matching success between the current block  $B$  in the next key frame and the block in the previous key frame motion compensated with vectors  $x_c$  and  $x_j$ .

- The weights have low values when the MSE for the candidate vector is high, i.e. when there is a high prediction error, and high values when the prediction error for the candidate vector is low. Thus, the choice of weights given by (3.11) takes into account the MSE prediction error of the block with respect to its neighbors. This will provide good adaptation of the weighted median vector filter in comparison to a simpler median vector filter which takes only into account the spatial properties of the motion field.

### 3.1.3.4 Bidirectional Motion Compensation

Once the final motion vectors are obtained, the interpolated frame  $Y_{2i}$  can be filled by simply using bidirectional motion compensation as defined in standard video coding schemes. It is assumed that the time interval between the previous key frame  $X_{2i-1}$  and  $Y_{2i}$  is similar to the time interval between  $Y_{2i}$  and the next key frame  $X_{2i+1}$ , so each key frame has the same weight ( $1/2$ ) when motion compensation is performed.

### 3.1.4 Slepian-Wolf Decoder in the IST-PDWZ Codec

The Slepian-Wolf decoder is another important module of the Wyner-Ziv decoder depicted in Figure 3.1. The main goal of this module is to estimate each bitplane extracted from the quantized symbol stream (see Section 3.1.1) in order to obtain an estimate,  $q'_{2i}$ , of the quantized symbol stream. For that purpose, the Slepian-Wolf decoder uses the parity sequences,  $\mathbf{P}_1$  and  $\mathbf{P}_2$ , sent by the Wyner-Ziv encoder (see Figure 3.2) and the available side information (an estimate or a “noisy” version of the encoded frame  $X_{2i}$  generated at the decoder – see Section 3.1.3).

In the IST-PDWZ architecture depicted in Figure 3.1, the Slepian-Wolf decoding is performed using an iterative turbo decoding procedure, as in [12]. Figure 3.14 illustrates a turbo decoder implementation constituted by two identical Soft-Input, Soft-Output (SISO) decoders (one per each constituent RSC code used in the turbo encoder implementation); since the two SISO decoders exchange information between them, the turbo decoder is called an iterative turbo decoder.

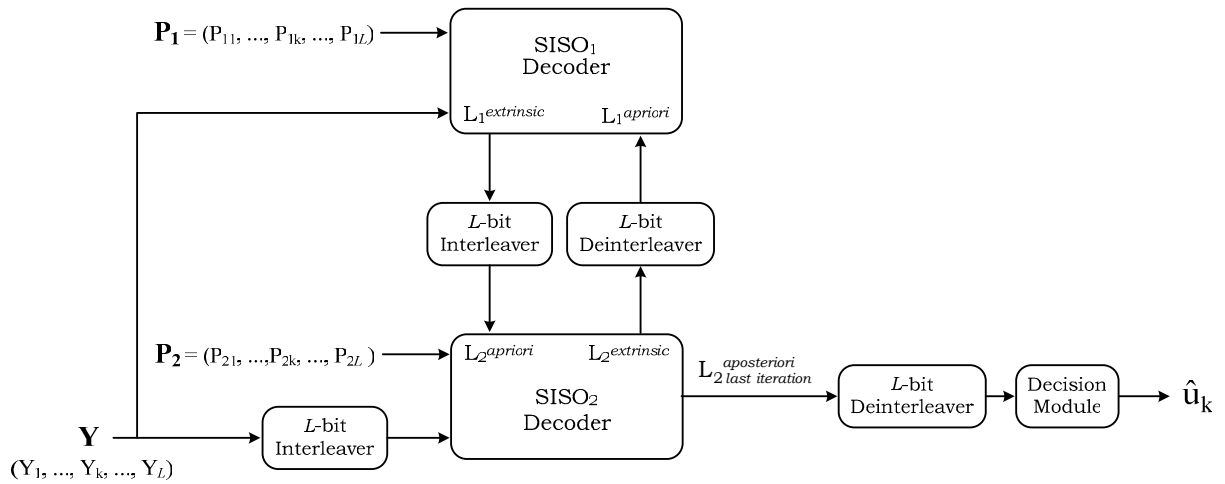


Figure 3.14 – Turbo decoder implementation using two identical soft-input, soft-output (SISO) decoders.

Generally, the SISO decoder input information is called *soft* input information and the information outputted by the SISO decoder is known as *soft* output information. The term *soft* information refers to information from which is possible to make a decision. Consider, for instance, the scenario where the probability of a bit,  $P(\text{bit})$ , being 0 and 1 are given by (3.12) and (3.13), respectively.

$$P(\text{bit} = 0) = 0.32 \tag{3.12}$$

$$P(\text{bit} = 1) = 0.68 \tag{3.13}$$

If a bit value decision has to be performed, taking into account the bit probabilities information, 1 will be the bit value chosen since the probability of the bit being 1,  $P(\text{bit} = 1)$ , is greater than



$P(\text{bit} = 0)$ ; the probabilities are therefore *soft* information. In the Figure 3.14 scenario, the SISO<sub>*i*</sub> decoder's inputs (*soft* inputs) are represented by the symbols  $\mathbf{Y}$ ,  $\mathbf{P}_i$  and  $L_i^{\text{apriori}}$  ( $i = 1, 2$ );  $\mathbf{Y}$  is often known as systematic information,  $\mathbf{P}_i$  as channel information and  $L_i^{\text{apriori}}$  as *a priori* information.

As mentioned in Section 3.1.2, each RSC encoder outputs systematic information (information that is equal to the one at the turbo encoder's input); this information is discarded and therefore no systematic information is sent to the decoder. The Wyner-Ziv decoder has however available the side information  $Y_{2i}$ , generated through frame interpolation techniques (see Section 3.1.3), which is an estimate or a “noisy” version of  $X_{2i}$ . The systematic information (represented by  $\mathbf{Y}$  in Figure 3.14) can therefore be provided by the side information  $Y_{2i}$  to the turbo decoder. Recall that “noise” in the Wyner-Ziv video coding context refers to the differences (‘errors’) between the side information  $Y_{2i}$  and the original Wyner-Ziv frame  $X_{2i}$ ; these differences have nothing to do with channel errors and strongly depend on the quality of the estimation made at the decoder.

The term “channel information” refers generically to the information transmitted by the encoder to the decoder through a channel. In the context of Figure 3.14, the channel information corresponds to the parity information produced by the RSC<sub>1</sub> and RSC<sub>2</sub> encoders and sent to the decoder; thus the channel information is represented by the symbols  $\mathbf{P}_1$  and  $\mathbf{P}_2$  in Figure 3.14.

The SISO<sub>*i*</sub> decoder's output (*soft* output) is often called *extrinsic* or reliability information; in Figure 3.14, the *extrinsic* information is represented by the symbol  $L_i^{\text{extrinsic}}$  ( $i = 1, 2$ ). Generically, the amplitude of the *extrinsic* information, for each  $u_k$ , indicates the reception confidence (reliability) of the  $u_k$  bit and the *extrinsic* information sign corresponds to the  $u_k$  bit value (0 or 1), as will be seen later in this Section; the symbol  $u_k$  represents the  $k^{\text{th}}$  bit of the  $L$ -bit sequence at the input of the turbo encoder (see Figure 3.2).

In the iterative turbo decoder depicted in Figure 3.14, the *a priori* information of a SISO decoder  $L^{\text{apriori}}$  corresponds to the *extrinsic* information computed by the other SISO decoder (represented by  $L_i^{\text{extrinsic}}$   $i = 1, 2$  in Figure 3.14). In an iterative turbo decoder, the *extrinsic* information (also called reliability information) about each  $u_k$  is exchanged between the two SISO decoders in order to improve the *a priori* information at the input of the other SISO decoder. That is, the SISO<sub>1</sub> decoder receives the *extrinsic* information of SISO<sub>2</sub> (represented by  $L_2^{\text{extrinsic}}$  in Figure 3.14) and uses it as SISO<sub>1</sub> *a priori* information (represented by  $L_1^{\text{apriori}}$  in Figure 3.14); in a similar way, the SISO<sub>2</sub> decoder receives the *extrinsic* information of SISO<sub>1</sub> and uses it as SISO<sub>2</sub> *a priori* information. Therefore, the *extrinsic* information sharing process between the two SISO decoders allows an iterative cooperation between them, i.e. an iterative decoding procedure. In fact, it is the iterative decoding scheme that, referencing to the turbo engine principle, is at the origin of the term ‘turbo’ codes. Note that, in the first iteration, the *a priori* information of the SISO<sub>1</sub> decoder is pre-established and not provided by SISO<sub>2</sub> since  $L_2^{\text{extrinsic}}$  is not yet available. It is interesting to point out that the *extrinsic* information provided by SISO<sub>2</sub> to SISO<sub>1</sub> results from information that is not accessible to SISO<sub>1</sub> decoder, i.e. the parity information associated to RSC<sub>2</sub> encoder, and vice-versa.

Since the  $L$ -bit input sequence of  $RSC_2$  is interleaved relatively to the  $L$ -bit input sequence of  $RSC_1$  (see Figure 3.2), it is straightforward that the  $L$ -bit interleaver and deinterleaver between the two SISO decoders arrange the *extrinsic* information in the proper order to each SISO decoder. For the same reason, it is necessary to interleave the  $\mathbf{Y}$  information (provided by the side information  $Y_{2i}$ ) before the SISO<sub>2</sub> decoder can use it. The knowledge of the interleaving and deinterleaving structures is thus essential to allow the turbo decoding procedure. In the IST-PDWZ codec, the same interleaving structure is pre-established for both the Slepian-Wolf encoder and decoder; the deinterleaving structure can be obtained from the interleaving structure, as shown in Section 3.1.2.

The iterative decoding procedure stops when a given convergence criteria is satisfied [8]. After the convergence criteria is fulfilled, an estimate of the  $L$ -bit sequence at the input of the turbo encoder (see Figure 3.2) is obtained based on a decision over the last iteration *a posteriori* information (represented by  $L_2^{aposteriori}$  in Figure 3.14). In the following, the SISO decoding algorithm and the decision operation are described.

#### **3.1.4.1 SISO Decoding Algorithm**

The decoding procedure of turbo codes may be performed using Maximum Likelihood (ML) algorithms (like the Soft-Output Viterbi Algorithm – SOVA) or the Maximum *A Posteriori* (MAP) algorithm [8]. Typically, the ML algorithms are used to estimate the most probable information sequence to have been transmitted and the MAP algorithm is used when the most probable information bit to have been transmitted is required to be estimated.

Since in the IST-PDWZ codec the turbo encoding is performed at the bit level (each bit  $u_k$  is independently turbo encoded – see Section 3.1.2), the decoding algorithm used in the SISO decoder implementation is a modified version of the maximum *a posteriori* (MAP) decoding algorithm proposed by Bahl, Cocke, Jelinek and Raviv in 1974 [50].

Bahl *et al.* showed in [50] that the MAP algorithm minimizes the bit error rate when used in the decoding process of linear block and convolutional codes. In 1993, Berrou, Glavieux and Thitimajshima [46] employed the MAP algorithm to the iterative decoding of turbo codes. However, Berrou *et al.* modified the MAP decoding algorithm in [50] since that algorithm does not consider the recursive nature of the RSC codes used by Berrou *et al.* [46] in the turbo encoder implementation; the Berrou's turbo encoder implementation is similar to the one depicted in Figure 3.2. In the following presentation of the SISO decoding algorithm the logical bit value 0 is represented by the value  $-1$  and the logical bit value 1 is represented by the value  $+1$ .

- The iterative SISO decoding procedure is performed in this Thesis using two modified MAP decoders operating over logarithms of likelihood ratios associated to the systematic and parity bits instead of the bits themselves. The estimate of each bit  $u_k$ , denoted by  $\hat{u}_k$ , is obtained from the Logarithm of the *A Posteriori* Probability (LAPP) ratio defined by (3.14) [51]

$$L(\hat{u}_k) \triangleq \ln \left( \frac{P(u_k = +1 | \mathbf{r})}{P(u_k = -1 | \mathbf{r})} \right) \quad (3.14)$$

where  $P(u_k = \pm 1 | \mathbf{r})$  is the *a posteriori* probability,  $L(\hat{u}_k)$  stands for the LAPP ratio and the operator  $\ln(\cdot)$  is the natural logarithm. The symbol  $\mathbf{r} = (r_1, r_2, \dots, r_L)$  with  $r_k = (r_k^s, r_k^p) = (Y_k, P_{ik})$  represents the information at the turbo decoder for each  $u_k$ : that is, the parity bit  $P_{ik}$  ( $i$  represents the index of the RSC encoder from which  $P_k$  belongs to) and the systematic bit  $Y_k$ .

- Applying the Bayes theorem to (3.14),  $L(\hat{u}_k)$  can be expressed as a sum of two terms where the second term represents the SISO decoder *a priori* information for all the bits  $u_k$ .

$$\begin{aligned} L(\hat{u}_k) &= \ln \left( \frac{p(\mathbf{r} | u_k = +1) \times P(u_k = +1)}{p(\mathbf{r} | u_k = -1) \times P(u_k = -1)} \right) \\ &= \ln \left( \frac{p(\mathbf{r} | u_k = +1)}{p(\mathbf{r} | u_k = -1)} \right) + \ln \left( \frac{P(u_k = +1)}{P(u_k = -1)} \right) \end{aligned} \quad (3.15)$$

In equation (3.15),  $p(\cdot)$  denotes the conditional probability density function (pdf) of  $\mathbf{r}$  given that the symbol  $u_k = \pm 1$  was transmitted.

- As was mentioned in Section 3.1.2, each RSC encoder is characterized by a trellis diagram similar to the one depicted in Figure 3.5; that diagram shows, for each RSC encoder state, the possible state transitions as well as the RSC output bits ( $u_k^s$  and  $u_k^p$ ) given the RSC encoder input  $u_k$ . Based on the trellis diagram knowledge, the *a posteriori* probability  $P(u_k = +1 | \mathbf{r})$  can be written as [51]

$$P(u_k = +1 | \mathbf{r}) = \sum_{S^+} p(S_{k-1}, S_k, \mathbf{r}) / p(\mathbf{r}) \quad (3.16)$$

where  $S^+$  represents all the possible state transitions  $S_{k-1} \rightarrow S_k$  considering the input  $u_k = +1$ . By analogy,

$$P(u_k = -1 | \mathbf{r}) = \sum_{S^-} p(S_{k-1}, S_k, \mathbf{r}) / p(\mathbf{r}). \quad (3.17)$$

where  $S^-$  represents all the possible state transitions  $S_{k-1} \rightarrow S_k$  considering the input  $u_k = -1$ . The trellis diagram must therefore be known by the turbo decoder in order to perform the decoding procedure [8].

- Substituting (3.16) and (3.17) in (3.14) yields

$$L(\hat{u}_k) = \ln \left( \frac{\sum_{S^+} p(S_{k-1}, S_k, \mathbf{r}) / p(\mathbf{r})}{\sum_{S^-} p(S_{k-1}, S_k, \mathbf{r}) / p(\mathbf{r})} \right). \quad (3.18)$$

### Computing the Probabilities $\tilde{\alpha}_k(S_k)$ , $\tilde{\beta}_{k-1}(S_{k-1})$ and $\gamma_k(S_{k-1}, S_k)$

From [51],  $p(S_{k-1}, S_k, \mathbf{r})$  may be expressed as a product of three probabilities

$$p(S_{k-1}, S_k, \mathbf{r}) = \tilde{\alpha}_{k-1}(S_{k-1}) \times \gamma_k(S_{k-1}, S_k) \times \tilde{\beta}_k(S_k). \quad (3.19)$$

- The probability  $\tilde{\alpha}_k(S_k)$  can be computed from (3.20) [51].

$$\tilde{\alpha}_k(S_k) = \frac{\sum_{S_{k-1}} \tilde{\alpha}_{k-1}(S_{k-1}) \times \gamma_k(S_{k-1}, S_k)}{\sum_{S_k} \sum_{S_{k-1}} \tilde{\alpha}_{k-1}(S_{k-1}) \times \gamma_k(S_{k-1}, S_k)} \quad (3.20)$$

Considering that the initial state of the RSC encoder,  $S_0$ , is known, the initialization of  $\tilde{\alpha}_k(S_k)$  (i.e. for  $k = 0$ ) is given by [51]

$$\tilde{\alpha}_0^i(S_k) = \begin{cases} 1, & S_k = S_0 \\ 0, & S_k \neq S_0 \end{cases}, \quad i = 1, 2 \quad (3.21)$$

where  $i$  stands for the RSC encoder index (1 for RSC<sub>1</sub> and 2 for RSC<sub>2</sub>).

- The probability  $\tilde{\beta}_{k-1}(S_{k-1})$  is given by (3.22) [51].

$$\tilde{\beta}_{k-1}(S_{k-1}) = \frac{\sum_{S_k} \tilde{\beta}_k(S_k) \times \gamma_k(S_{k-1}, S_k)}{\sum_{S_k} \sum_{S_{k-1}} \tilde{\alpha}_{k-1}(S_{k-1}) \times \gamma_k(S_{k-1}, S_k)} \quad (3.22)$$

Equation (3.23) expresses the boundary conditions of  $\tilde{\beta}_k^1(S_k)$  (i.e. for  $k = L$ ) assuming that the final state of the RSC<sub>1</sub> encoder  $S_L$  is known [51].

$$\tilde{\beta}_L^1(S_k) = \begin{cases} 1, & S_k = S_L \\ 0, & S_k \neq S_L \end{cases} \quad (3.23)$$

If the RSC<sub>1</sub> encoder's final state  $S_L$  is unknown, the boundary condition is given by [8]

$$\tilde{\beta}_L^1(S_k) = \frac{1}{2^m}, \quad \forall S_k, \quad (3.24)$$

where  $m$  is the memory of the RSC encoder (see Section 3.1.2). The RSC<sub>2</sub> encoder  $\tilde{\beta}_L^2(S_k)$  probability is initialized with the probability  $\tilde{\alpha}_L^2(S_k)$  values for all the states  $S_k$  [51].

- From [51], the probability  $\gamma_k(S_{k-1}, S_k)$  can be written as

$$\gamma_k(S_{k-1}, S_k) = P(u_k) p(\mathbf{r}_k | u_k) \quad (3.25)$$

and the *a priori* probability  $P(u_k)$  [51] can be expressed as

$$P(u_k = \pm 1) = \frac{e^{-\frac{1}{2}L^{apriori}(u_k)}}{1 + e^{-L^{apriori}(u_k)}} \times e^{-\frac{u_k L^{apriori}(u_k)}{2}} = A_k \times e^{-\frac{u_k L^{apriori}(u_k)}{2}} \quad (3.26)$$

where  $L^{apriori}(u_k)$  is defined by equation (3.27).

$$L^{apriori}(u_k) \triangleq \frac{P(u_k = +1)}{P(u_k = -1)} \quad (3.27)$$

In equation (3.25), the  $p(\mathbf{r}_k | u_k)$  factor is the conditional pdf of  $\mathbf{r}_k$  given that  $u_k$  was transmitted; as was already mentioned in this Section,  $\mathbf{r}_k = (r_k^S, r_k^P) = (Y_k, P_{ik})$  with  $P_{ik}$  representing the  $k^{\text{th}}$  parity bit received at the decoder ( $i$  represents the index of the RSC encoder from which  $P_k$  belongs to) and  $Y_k$  the  $k^{\text{th}}$  systematic bit, for each  $u_k$ .

### **Modelling the Residual of the Parity Information**

In order to compute the  $\gamma_k(S_{k-1}, S_k)$  probability, it is necessary to know how to model the conditional pdf  $p(\mathbf{r}_k | u_k)$ . In the IST-PDWZ codec, it is assumed that no errors are introduced in the parity bits (i.e. the Wyner-Ziv bitstream) transmission (as in [12]). As it is well-known, the error-free transmission scenario is the most favourable transmission scenario that can be taken into account and therefore it is in this situation that the best Wyner-Ziv codec RD performance can be achieved. Since no errors are introduced during the parity bitstream transmission, the parity bits transmitted are equal to the parity bits received at the decoder (generically represented in Figure 3.14 by  $\mathbf{P}_1$  and  $\mathbf{P}_2$ ).

As was mentioned in Section 3.1.2, puncturing is used at the encoder and therefore some parity bits are not transmitted to the decoder; those bits are often called deleted bits. Since the decoder knows the puncturing pattern (Section 3.1.2), the deleted bits positions in the bitstreams  $\mathbf{P}_1$  and  $\mathbf{P}_2$  are known. The deleted bits positions are filled with a zero value, indicating that no value was received for that parity bit position [51]; remember that, in the  $\mathbf{P}_1$  and  $\mathbf{P}_2$  bitstreams, the value +1 corresponds to the bit 1 and the value -1 corresponds to the bit 0.

Since, in the IST-PDWZ solution, an error-free parity bits transmission scenario is assumed, the probability of receiving the parity bit  $P_{ik} = \pm 1$  given that  $u_k^p = \pm 1$  was transmitted is one and the probability of receiving  $P_{ik} = \mp 1$  given that  $u_k^p = \pm 1$  was transmitted is zero. Thus, the conditional pdf  $p(r_k^p | u_k) = p(P_{ik} | u_k)$  may be described by the *Dirac delta function*  $\delta(t)$  typically defined by (3.28) and (3.29) [52]; in (3.28),  $\varepsilon$  is an arbitrarily small value.

$$\int_{-\varepsilon}^{\varepsilon} \delta(t) dt = 1 \quad (3.28)$$

$$\delta(t) = 0, \quad t \neq 0 \quad (3.29)$$

Despite these *Dirac delta function* definitions,  $\delta(t)$  is not a function in the strict mathematical sense [52]. Thus, in order to be able to compute  $p(P_{ik} | u_k)$ , it is necessary to approximate the function  $\delta(t)$  in the strict mathematical sense. Consider a Gaussian distribution  $p(x)$  with mean zero and variance  $\sigma^2$  given by (3.30).

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}} \quad (3.30)$$

$$\propto C \times e^{-\frac{x^2}{2\sigma^2}}$$

Consider now the scenario where the Gaussian distribution variance  $\sigma^2$  is arbitrarily small; if the  $x$  value is zero then  $p(x) \sim 1$ , otherwise  $p(x)$  will tend to zero. The Gaussian distribution with mean zero and an arbitrarily small variance  $\sigma^2$  may therefore approximate the *Dirac delta function*.

### **Modelling the Residual of the Systematic Information**

As was already mentioned in this Section, the side information  $Y_{2i}$  (an estimate of the  $X_{2i}$  frame generated at the decoder – see Section 3.1.3), provides the systematic information (represented in Figure 3.14 by  $\mathbf{Y}$ ) to the turbo decoder.

Figure 3.15 depicts the distribution of the residual  $x$ , i.e. the luminance difference between corresponding pixels in  $X_{2i}$  (frame to be coded) and  $Y_{2i}$  (estimation) for the *Foreman* QCIF video sequence. A Laplacian distribution given by (3.31) is also plotted in Figure 3.15, with the parameter alpha equal to 0.58.

$$f(x) = \frac{\alpha}{2} e^{-\alpha|x|} \quad (3.31)$$

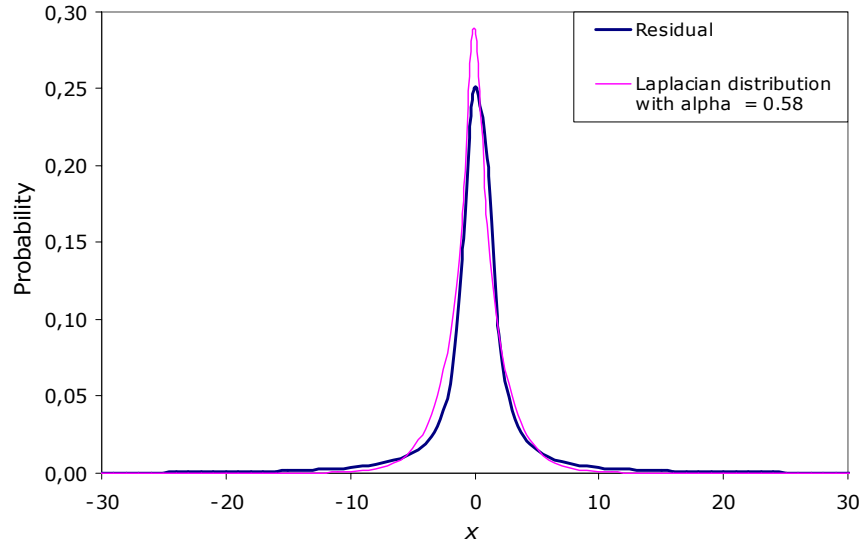


Figure 3.15 – Residual distribution for the Foreman QCIF video sequence.

As it can be noticed, the Laplacian distribution approximates well the residual  $x$  distribution; thus, the residual between  $X_{2i}$  and  $Y_{2i}$  is modelled by a Laplacian distribution in the IST-PDWZ solution (as in [12]).

### Modelling the Conditional pdf $p(\mathbf{r}_k | \mathbf{u}_k)$

From [53], the probability  $p(\mathbf{r}_k | \mathbf{u}_k)$  can be written as

$$\begin{aligned}
 p(\mathbf{r}_k | \mathbf{u}_k) &= p(\mathbf{r}_k^s | \mathbf{u}_k) p(\mathbf{r}_k^p | \mathbf{u}_k) \\
 &\propto e^{-\alpha |r_k^s - u_k|} \times e^{-\frac{(r_k^p - u_k^p)^2}{2\sigma^2}} \\
 &= e^{-\alpha |r_k^s - u_k|} \times e^{-\frac{(r_k^p)^2 + (u_k^p)^2 - 2(r_k^p \times u_k^p)}{2\sigma^2}} \\
 &= \mathbf{B}_k \times e^{-\alpha |r_k^s - u_k|} \times e^{-\frac{(r_k^p \times u_k^p)}{\sigma^2}}
 \end{aligned} \tag{3.32}$$

since the parity information “error” or residual is independent of the systematic information “error”. The “error” or the residual between the turbo encoder  $L$ -bit input sequence (turbo encoder systematic information) and the corresponding information at the turbo decoder ( $\mathbf{Y}$ ) is modelled by a Laplacian distribution; the parity information “error” is modelled by a Gaussian distribution with an arbitrarily small variance.

Substituting (3.26) and (3.32) in (3.25) yields

$$\gamma_k(S_{k-1}, S_k) \propto A_k \times e^{\frac{u_k L^{apriori}(u_k)}{2}} \times B_k \times e^{\frac{r_k^p \times u_k^p}{\sigma^2}} \times e^{-\alpha |r_k^s - u_k|}. \quad (3.33)$$

Then equation (3.33) may be approximated by [51]

$$\gamma_k(S_{k-1}, S_k) \sim e^{\frac{u_k L^{apriori}(u_k)}{2}} \times e^{\frac{r_k^p \times u_k^p}{\sigma^2}} \times e^{-\alpha |r_k^s - u_k|} \quad (3.34)$$

since the term  $\gamma_k(S_{k-1}, S_k)$  appears both in the numerator and denominator of (3.18); the product  $A_k \times B_k$  is independent of  $u_k$  and therefore can be cancelled.

### **Logarithm of the *a Posteriori* Probability (LAPP)**

Thus, equation (3.18) can be rewritten using (3.19) and (3.34) yielding

$$\begin{aligned} L(\hat{u}_k) &= \ln \left( \frac{\sum_{S^+} \tilde{\alpha}_{k-1}(S_{k-1}) \times e^{\frac{u_k L^{apriori}(u_k)}{2}} \times e^{\frac{r_k^p \times u_k^p}{\sigma^2}} \times e^{-\alpha |r_k^s - u_k|} \times \tilde{\beta}_k(S_k)}{\sum_{S^-} \tilde{\alpha}_{k-1}(S_{k-1}) \times e^{\frac{u_k L^{apriori}(u_k)}{2}} \times e^{\frac{r_k^p \times u_k^p}{\sigma^2}} \times e^{-\alpha |r_k^s - u_k|} \times \tilde{\beta}_k(S_k)} \right) \\ &= L^{apriori}(u_k) + \ln \left( \frac{\sum_{S^+} \tilde{\alpha}_{k-1}(S_{k-1}) \times \tilde{\beta}_k(S_k) \times e^{\frac{r_k^p \times u_k^p}{\sigma^2}} \times e^{-\alpha |r_k^s - u_k|}}{\sum_{S^-} \tilde{\alpha}_{k-1}(S_{k-1}) \times \tilde{\beta}_k(S_k) \times e^{\frac{r_k^p \times u_k^p}{\sigma^2}} \times e^{-\alpha |r_k^s - u_k|}} \right). \end{aligned} \quad (3.35)$$

In equation (3.35), the term  $L^{apriori}(u_k)$  is the *a priori* information about each  $u_k$  and the second term represents the *extrinsic* information which is exchanged between the two SISO decoders. Notice that for the first iteration of the iterative decoding, the SISO<sub>1</sub> decoder does not have the *extrinsic* information of SISO<sub>2</sub> available. Thus, assuming the information bits  $u_k = -1$  and  $u_k = +1$  equally probable [51], the SISO<sub>1</sub> *a priori* information (equation (3.27)) is set to zero, i.e.  $L_1^{apriori}(u_k) = 0$ , for  $k = 1$  to  $k = L$ .

The iterative decoding is performed until a maximum number of decoding iterations is reached [51]. In the IST-PDWZ codec the maximum allowed decoding iteration number is 18; through simulations, it was concluded that 18 iterations allows the turbo decoder to converge. In the context of the IST-PDWZ architecture depicted in Figure 3.1, convergence means that the bit error probability decrease ideally to the zero bit error probability scenario.

For more details about the expressions used so far in this Section, the reader should consult [46], [50], [51] and [53].



### 3.1.4.2 Decision Operation

After the iterative decoding stops, the last iteration *a posteriori* information of SISO<sub>2</sub> is computed using equation (3.35).

- The estimate  $\hat{u}_k$  of each turbo encoder input bit  $u_k$  results from thresholding the SISO<sub>2</sub> last iteration *a posteriori* information (this operation is performed in the decision module represented in Figure 3.14). Since the transmitted symbols are +1 and –1, the threshold value is set to zero (central value between +1 and –1).
- After having the estimate  $\hat{u}_k$  for each  $u_k$ , the bit error rate,  $P_e$ , is calculated. As in [12], the bit error rate results from the comparison between each estimate  $\hat{u}_k$  and each information bit  $u_k$ , which is often called ideal error detection. If  $P_e$  is greater than a given bit error rate threshold, the Slepian-Wolf decoder requests more parity bits from the Slepian-Wolf encoder via the feedback channel and the iterative decoding procedure continues; otherwise, the turbo decoding procedure ends and the output of the turbo decoder is the sequence of  $\hat{u}_k$ 's from  $k = 1$  to  $k = L$  which corresponds to an estimate of a given bitplane). In the IST-PDWZ codec, the bit error rate threshold is set to  $1 \times 10^{-3}$  in order to allow comparing the IST-PDWZ results with the ones available in [12].
- After obtaining an estimate of a given bitplane, the turbo decoder starts decoding the following bitplane in terms of significance; the turbo decoder always starts by decoding the most significant bitplane.

### 3.1.5 Reconstruction in the IST-PDWZ Codec

In the IST-PDWZ architecture, illustrated in Figure 3.1, the last stage towards decoding a Wyner-Ziv frame is reconstruction.

- After turbo decoding the  $M$  bitplanes associated to the  $X_{2i}$  quantized pixel values, these  $M$  bitplanes are grouped together to form the decoded quantized symbol stream  $q'_{2i}$  corresponding to the whole Wyner-Ziv frame  $X_{2i}$ . Notice that the decoder knows the  $M$  parameter value used at the encoder side since it is transmitted to the decoder in a bitstream header.
- Once the decoded quantized symbol stream  $q'_{2i}$  is obtained, the reconstruction of the  $X_{2i}$  frame can be performed.
- In the IST-PDWZ codec, the reconstruction procedure for each pixel value can be described by one of three cases listed in the following; Figure 3.16 illustrates the reconstruction function for a 4-level uniform scalar quantizer.

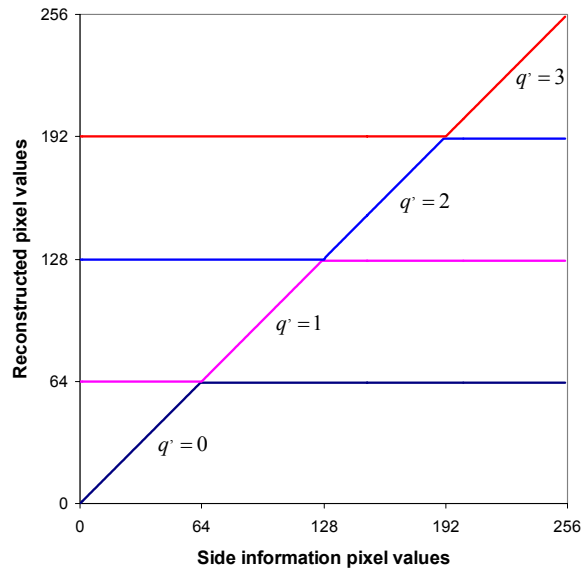


Figure 3.16 – Reconstruction function for a 4-level uniform scalar quantizer.

where  $q'$  stands for the decoded quantized symbol; for a 4-level uniform scalar quantizer  $q'$  can assume any integer value between 0 and 3.

### Case I

If the side information pixel value is within the decoded quantized symbol  $q'$  then the reconstructed pixel value is made equal to the side information pixel value; this corresponds to the  $45^\circ$  zone in Figure 3.16.

### Case II

If the side information pixel value belongs to a quantized symbol lower in magnitude than the decoded one  $q'$ , then the reconstructed pixel value assumes the lowest intensity value within the decoded quantized symbol, i.e. the lower bound of the quantization interval indexed by the decoded quantized symbol  $q'$ .

### Case III

If the side information pixel value belongs to a quantized symbol higher in magnitude than the decoded quantized symbol  $q'$ , then the reconstructed pixel value assumes the highest intensity value within the decoded quantized symbol, i.e. the upper bound of the quantization interval indexed by the decoded quantized symbol.





Since the reconstructed pixel value is always in between the boundaries of the decoded quantized symbol  $q'$ , the error between the frames  $X_{2i}$  and  $X'_{2i}$  (also known as reconstruction distortion) is always limited by the quantizer coarseness; the higher it is  $M$  and thus less coarse the quantization, the lower it is the reconstruction error.

## 3.2 IST-PDWZ Experimental Results

Four video test sequences have been selected for the evaluation of the rate-distortion performance of the IST-PDWZ codec proposed in this Chapter, which architecture is depicted in Figure 3.1.

A brief description of each test sequence main characteristics' is provided in Table 3.2 (fps stands for frames *per* second). A more detailed description of each test sequence can be found in Annex A.

Table 3.2 – Main characteristics of the video test sequences.

Video Sequence Name	<i>Foreman</i>	<i>Mother and Daughter</i>	<i>Coastguard</i>	<i>Stefan</i>
Sample Frame				
Total Number of Frames	400	961	300	300
Number of Frames Evaluated	101	101	299	299
Spatial Resolution	QCIF	QCIF	QCIF	QCIF
Temporal Resolution (fps)	30	30	25	25

In Table 3.2, the *Foreman* and *Mother and Daughter* test sequences are examples of video conference content typically characterized by low and medium amount of movement (activity). The *Coastguard* test sequence is characterized by well defined motion of the objects present in the scene (the boats); the *Stefan* test sequence is an example of sports content characterized by higher activity. As it is well-known, the higher the activity, the more difficult is typically to code the video content. This content variety is important to collect enough representative and meaningful results for the IST-PDWZ performance.

Since some test conditions are common to all the performance evaluation processes performed in the following, those test conditions will be mentioned only once to avoid repetition. The main common test conditions are listed in the following:

- Only the luminance data is considered in the IST-PDWZ rate-distortion performance evaluation in order to allow comparing the results obtained with those available in [12].

- The key frames, represented in Figure 3.1 by  $X_{2i-1}$  and  $X_{2i+1}$ , are considered to be losslessly available at the decoder (to change this condition is a major task for future work).
- The Wyner-Ziv bitstream is assumed to be error-free received, i.e. no errors are introduced during the transmission (as in [12]).
- In the turbo encoder implementation, two recursive systematic convolutional encoders of rate  $\frac{1}{2}$  are employed; each one is represented by the generator matrix  $\begin{bmatrix} 1 & \frac{1+D+D^3+D^4}{1+D^3+D^4} \end{bmatrix}$  (for more details the reader should consult Section 3.1.2).
- The side information is generated through frame interpolation algorithms at the decoder side. Besides forward and bidirectional motion estimation, a spatial motion smoothing algorithm is used to eliminate motion outliers allowing significant improvements in the RD performance (for more details the reader should consult Section 3.1.3).
- As in [12], a Laplacian distribution models the residual between the  $X_{2i}$  frame pixel values and the corresponding pixel values of the  $Y_{2i}$  frame; it is therefore possible to compare the results obtained with the IST-PDWZ codec and those available in [12].
- The Laplacian parameter is estimated over the total number of Wyner-Ziv frames evaluated for a given video sequence. For each video sequence, the Laplacian distribution parameter estimation is performed offline that is before the Wyner-Ziv coding procedure.
- Through simulations, it was concluded that 18 iterations allow the turbo decoder to converge; thus, the maximum allowable number of turbo decoding iterations is 18.
- The bit error rate threshold is assumed to be  $1 \times 10^{-3}$ ; the main reason for this choice has to do with the possibility of comparing the IST-PDWZ results with those available in [12].
- Since what is important at this stage is to evaluate the Wyner-Ziv codec RD performance, the rate-distortion plots only contain the rate and the PSNR values for the even frames, i.e. the Wyner-Ziv coded frames, of a given video sequence (the key frames are lossless anyway).
- For each test sequence, the IST-PDWZ RD performance is compared against H.263+ intraframe coding and H.263+ interframe coding with a I-B-I-B structure. In the later case, only the rate and PSNR of the B frames is shown since the Wyner-Ziv codec performance is here the target.

The results obtained with the IST-PDWZ codec for each one of the four QCIF video sequences, listed in Table 3.2, will be presented and analysed in the following sections.

### **3.2.1 *Foreman* Test Sequence Evaluation**

Although the *Foreman* QCIF sequence has 400 frames, only the first 101 frames of the sequence were considered in the IST-PDWZ codec RD performance evaluation in order to

allow comparing the IST-PDWZ codec performance with the performance achieved by Aaron *et al.* in [12], for the pixel domain scenario, under the same test conditions.

Figure 3.17 shows the IST-PDWZ PSNR results obtained for the *Foreman* QCIF test sequence. As it can be noticed from Figure 3.17 (a), the IST-PDWZ codec provides better rate-distortion results when compared to those available in [12], with coding improvements up to 2.3 dB for the lower bitrates although the improvements are rather constant for all bitrates.

The lack of details about the technical solution used in [12] to plot the pixel domain results and probably the more efficient frame interpolation tools employed in the IST-PDWZ codec may explain the difference between the two curves (correspondent to the IST-PDWZ solution and the pixel domain curve plotted in [12]). This lack of details led the author of this Thesis to develop new tools which may be different from the tools used in [12], making the two coding solutions different although it is not precisely known how much different. As it can be observed from the results depicted in Figure 3.17 (b), the IST-PDWZ codec presents significant gains over H.263+ intraframe coding for all bitrates. There is still however a compression gap when comparing to H.263+ interframe coding with I-B-I-B structure.

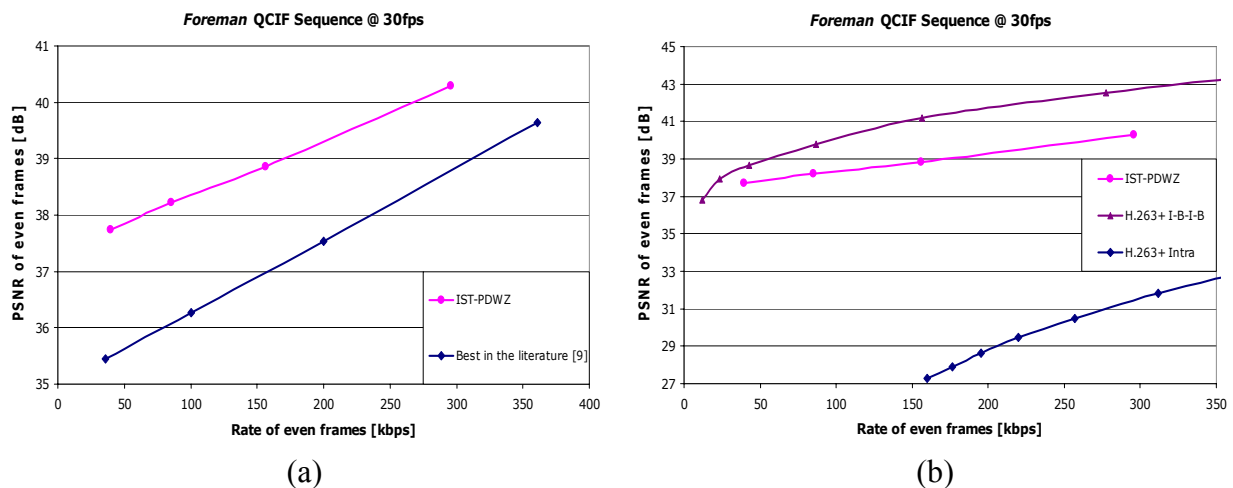


Figure 3.17 – IST-PDWZ rate-distortion performance for the *Foreman* test sequence.

### 3.2.2 Mother and Daughter Test Sequence Evaluation

In order to be able to compare the IST-PDWZ codec RD performance with the performance achieved by Aaron *et al.* in [12] for the *Mother and Daughter* QCIF under the same test conditions, only the first 101 frames of the sequence were considered in the IST-PDWZ codec RD performance evaluation (because this is what is used in [12], although the sequence has 961 frames in total).

Figure 3.18 shows the IST-PDWZ rate-distortion results obtained for the *Mother and Daughter* QCIF test sequence; the rate-distortion results achieved in [12] for the pixel domain scenario are also plotted in Figure 3.18. From the results, it can be observed that the IST-PDWZ codec

provides better results when compared to the pixel domain results achieved in [12], with coding improvements up to 1.3 dB in the lower/middle bitrates.

Once more, the lack of details about the solution used in [12] to plot the pixel domain results may explain the difference between the two curves. From the results depicted in Figure 3.18 (b), it is also possible to notice remarkable gains over H.263+ intraframe coding for all bitrates. However, there is still a compression gap when comparing to H.263+ interframe coding with I-B-I-B structure.

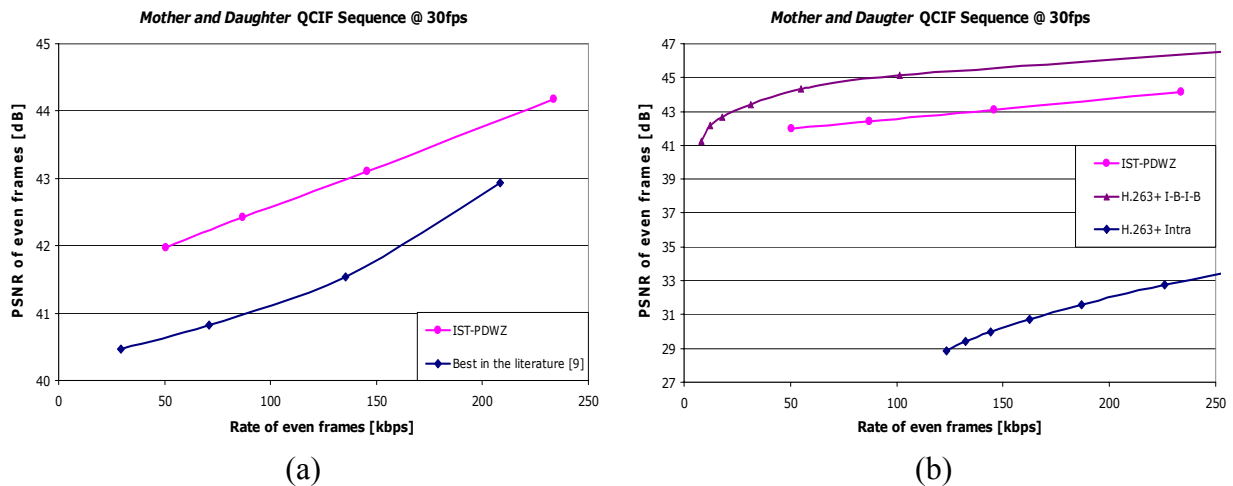


Figure 3.18 – IST-PDWZ rate-distortion performance for the Mother and Daughter test sequence.

### 3.2.3 Coastguard Test Sequence Evaluation

The IST-PDWZ codec RD performance was also evaluated using the *Coastguard* test sequence. For the *Coastguard* QCIF test sequence, 299 frames of the video sequence were taken into account (see Table 3.2)<sup>1</sup>.

Figure 3.19 shows the IST-PDWZ PSNR results achieved for the *Coastguard* QCIF test sequence. From the results depicted in Figure 3.19, it can be observed that the IST-PDWZ solution presents remarkable coding gains over H.263+ intraframe coding for all bitrates. However, there is still a compression gap when compared to H.263+ interframe coding with I-B-I-B structure. No comparison with the solution in [12] is performed since no results are available for this sequence.

<sup>1</sup> As was mentioned in Section 3.1, a video sequence is divided into Wyner-Ziv frames (the even frames of the video sequence) and key frames (the odd frames of the video sequence). Since the side information  $Y_{2i}$  for each  $X_{2i}$  frame is generated through frame interpolation from the previous and the next temporally adjacent frames  $X_{2i-1}$  and  $X_{2i+1}$ , an odd number of frames must be considered.

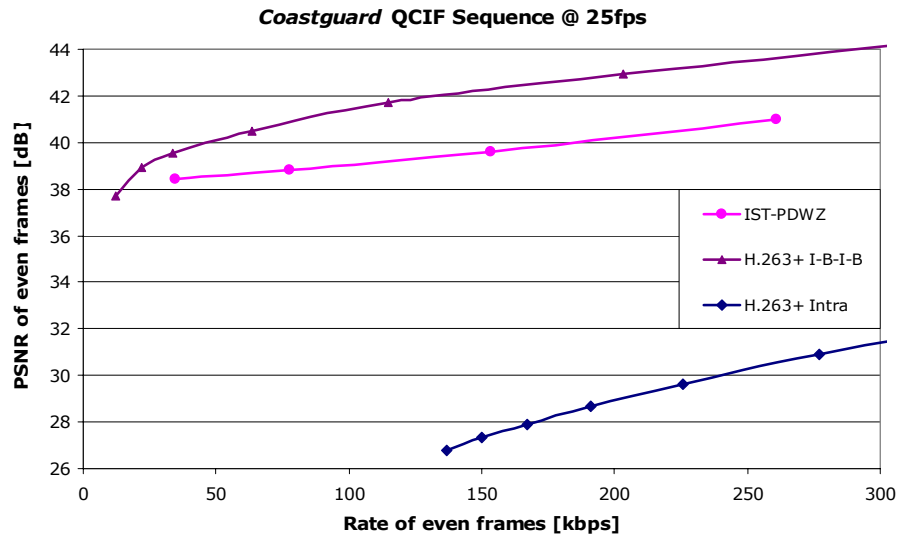


Figure 3.19 – IST-PDWZ rate-distortion performance for the Coastguard test sequence.

### 3.2.4 Stefan Test Sequence Evaluation

The *Stefan* QCIF sequence was the fastest (in terms of amount of motion) test sequence selected for the evaluation of the IST-PDWZ codec RD performance. As Table 3.2 shows, 299 video frames were considered in the IST-PDWZ codec RD performance evaluation.

Figure 3.20 shows the IST-PDWZ rate-distortion results obtained for the *Stefan* QCIF test sequence. As it can be observed from the results depicted in Figure 3.20, the IST-PDWZ solution exhibits coding gains over H.263+ intraframe coding for all bitrates. However, there is still a compression gap when comparing to H.263+ interframe coding with I-B-I-B structure. No comparison with the solution in [12] is performed since no results are available for this sequence.

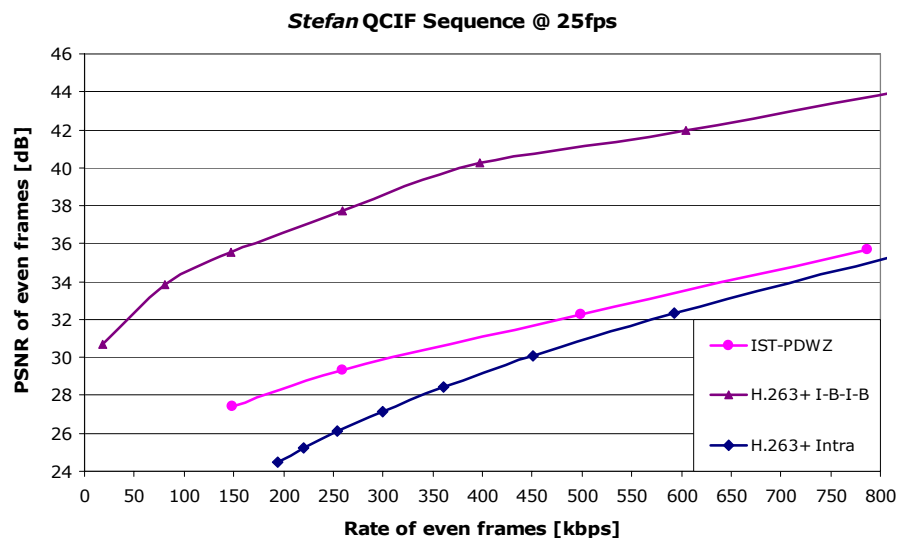


Figure 3.20 – IST-PDWZ rate-distortion performance for the Stefan test sequence.

### 3.2.5 IST-PDWZ versus H.263+ Intraframe Coding Gains

Figure 3.21 depicts the coding gains of the IST-PDWZ solution over H.263+ intraframe coding for the four test sequences evaluated: *Foreman*, *Mother and Daughter*, *Coastguard* and *Stefan* QCIF sequences.

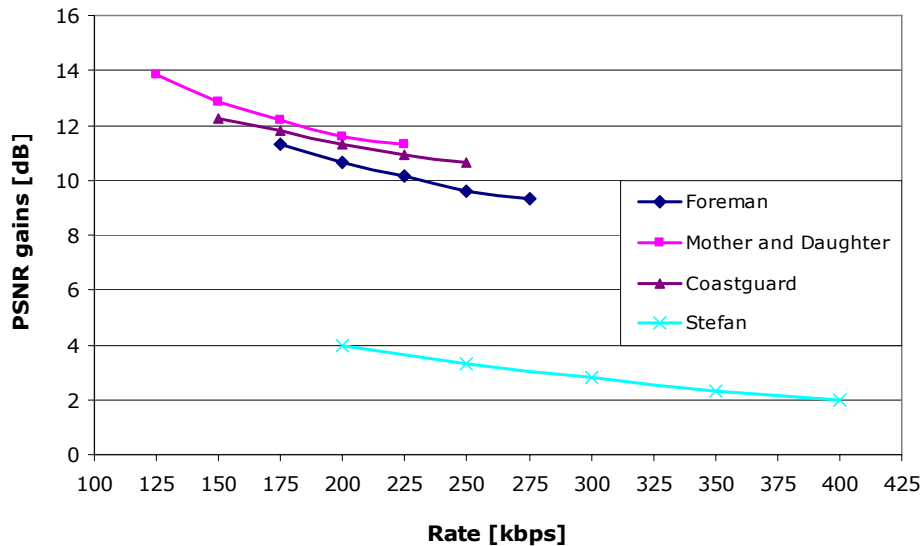


Figure 3.21 – PSNR gains over H.263+ intraframe coding for the *Foreman*, *Mother and Daughter*, *Coastguard* and *Stefan* QCIF video sequences.

The IST-PDWZ rate-distortion performance is above the H.263+ intraframe coding for all bitrates and test sequences, as Figure 3.21 illustrates. For video sequences characterized by a high amount of movement, e.g. the *Stefan* video sequence, the IST-PDWZ codec presents coding gains up to 4 dB regarding the H.263+ intraframe coding. For the video sequences with a lower activity, like the *Coastguard* and the *Mother and Daughter* video sequences, the IST-PDWZ codec shows higher coding gains over H.263+ intraframe coding (the coding gains are up to 13.9 dB for the *Mother and Daughter* sequence and 12.3 dB for the *Coastguard* sequence). For the *Foreman* test sequence, the IST-PDWZ codec coding gains are up to 11.3 dB.

### 3.3 Final Remarks

In this Chapter, an improved Wyner-Ziv video coding solution, named IST-PDWZ, following the same architecture as the one proposed by Aaron *et al.* in [17] has been presented. The proposed IST-PDWZ codec has some differences regarding the codec proposed in [17], notably in the Slepian-Wolf codec and the frame interpolation module. These differences are partly motivated by the lack of details regarding the solution proposed in [17] but also related to improvements explicitly introduced.

The evaluation of the IST-PDWZ RD performance is made in Section 3.2. For the QCIF test sequences *Foreman* and *Mother and Daughter*, the IST-PDWZ codec RD performance can be



directly compared with the more recent pixel domain results published by Aaron *et al.* in [12]. The coding improvements up to 2.3 dB for the IST-PDWZ codec regarding the results published in [12] may be explained by the new solutions developed for the Slepian-Wolf codec and the frame interpolation module.

The results presented in Section 3.2 show that the IST-PDWZ codec RD performance is typically considerably above H.263+ intraframe coding for all bitrates and all the test sequences. For the *Stefan* QCIF test sequence, despite the IST-PDWZ RD performance being above the H.263+ intraframe coding, the coding gain is much lower than for the *Foreman*, *Coastguard* and *Mother and Daughter* QCIF test sequences; this means that the frame interpolation algorithm performed at the decoder does not well model the type of motion in the *Stefan* QCIF sequence. The higher the amount of motion in the video sequence, the lower the coding gain of the IST-PDWZ codec regarding the H.263+ intraframe coding, as expected. Comparing the IST-PDWZ codec RD performance to the H.263+ interframe coding with I-B-I-B structure, there is still a compression gap; this gap is higher for video sequences characterized by a high amount of motion, like the *Stefan* QCIF test sequence, again because the frame interpolation tools employed to generate the side information at the decoder cannot provide a good performance for this kind of video sequences.



## Chapter 4

### IST-Transform Domain Wyner-Ziv Codec

Distributed Video Coding (DVC) is a new video coding paradigm that relies on the Slepian-Wolf and Wyner-Ziv key Information Theory results established in the 1970's; this new video coding paradigm enables to explore the video statistics, partially or totally, at the decoder only, as was seen in previous chapters. A particular case of distributed video coding is the so-called Wyner-Ziv video coding. In the Wyner-Ziv coding scenario, two correlated sources are independently encoded using separate encoders but the encoded streams, associated to each source, are jointly decoded exploiting the correlation between them (for more details see Chapter 2); in the video coding context, the two correlated sources can be two temporally adjacent frames of a video sequence, for example.

Despite the Wyner-Ziv coding theoretical foundations being known for a long time (since the 1970's), practical solutions of Wyner-Ziv coding are much more recent. The emergence of applications characterized by different encoding requirements from those targeted by the traditional delivery systems, e.g. low-complexity and low-power consumption at the encoder, are at the basis of recent efforts towards practical Wyner-Ziv coding solutions. While in traditional video coding schemes, the temporal correlation between adjacent frames of a video sequence is exploited only at the encoder, in the Wyner-Ziv video coding scenario the same correlation can be exploited at the decoder. This means that the high complexity associated with the motion estimation task (performed at the encoder in the traditional video coding) may be shifted to the decoder in a Wyner-Ziv coding scheme, leading to a lower encoding complexity at the expense of a higher decoding complexity.

In Chapter 3, a Wyner-Ziv video coding solution called IST-PDWZ (from Instituto Superior Técnico-Pixel Domain Wyner-Ziv) was proposed, implemented and evaluated in detail; the

IST-PDWZ solution is a much improved version of the approach proposed by Aaron *et al.* in [17] developed at Instituto Superior Técnico by the author of this Thesis. The results in Chapter 3 show that the Rate-Distortion (RD) performance of the IST-PDWZ solution is in between the RD performance of H.263+ intraframe coding and the H.263+ interframe coding with a I-B-I-B structure. Although the IST-PDWZ solution outperforms H.263+ intraframe coding and the comparable Wyner-Ziv codecs available in the literature [17], [12], further work still needs to be done in order to achieve a rate-distortion performance similar to the best available hybrid video coding schemes, at least for similar encoding complexity.

Transform coding is a tool commonly used in traditional image and video coding to explore the spatial correlation within an image or a frame of a video sequence. Generally, the transform is applied over  $n \times n$  sample blocks of a frame, spatially decorrelating the samples inside a block, i.e. converting correlated pixel values into independent transform coefficients. Since, within a block, neighbouring samples are typically strongly correlated, it is possible to represent more efficiently those samples in the frequency domain. The spatial decorrelation within a  $n \times n$  samples block allows block energy to be concentrated in a small number of large valued transform coefficients. The DC coefficient (corresponding to the lowest spatial frequency) and typically the transform coefficients near the DC coefficient enclose most of the  $n \times n$  samples block energy; for that reason, those coefficients are often called low-frequency transform coefficients. The amount of bits necessary to encode a frame can be reduced by considering only those large valued low-frequency coefficients and neglecting the remaining transform coefficients since their amplitude is typically near zero. The impact in the quality of the decoded frame of only transmitting the low-frequency transform coefficients should not be noticed by the Human Visual System (HVS), although this strongly depends on the video content.

As was mentioned in Chapter 2, transform coding is a tool that can also be used in distributed video coding with the same purpose as in traditional video coding, i.e. to exploit spatial correlation between neighbouring sample values. This spatial correlation is not exploited in the IST-PDWZ solution and thus a better rate-distortion performance can be achieved if the transform coding tool is used; however the usage of this tool should not compromise the low-complexity encoding requirement needed for several emerging applications, e.g. wireless low-power surveillance networks.

The starting point for the work presented in this Chapter is the IST-PDWZ codec, already described in Chapter 3. IST-TDWZ (from Instituto Superior Técnico-Transform Domain Wyner-Ziv) is the designation for the new coding solution proposed in this Chapter; its architecture is similar to the one proposed by Aaron *et al.* in [12]. The usage of the transform coding tool in order to achieve a better rate-distortion performance constitutes the main difference between the IST-TDWZ and the IST-PDWZ coding solutions.

## 4.1 IST-Transform Domain Wyner-Ziv Codec Architecture

The IST-TDWZ general architecture presented in Figure 4.1 is similar to the one proposed by Aaron *et al.* in [12] since the solution proposed in [12] represents the starting point, in terms of transform domain architecture, for some recently and more sophisticated proposed solutions, like the one present in [19]. Both architectures make use of transform coding, namely the Discrete Cosine Transform (DCT), and the modules from the pixel domain Wyner-Ziv codec: a uniform quantizer, a turbo-code based Slepian-Wolf codec, a frame interpolation module and a reconstruction module. There are however some differences between the IST-TDWZ solution and the one proposed in [12], namely in the Slepian-Wolf codec, DCT, quantizer and frame interpolation modules. The main reason for these differences is related with the lack of detail in [12] regarding the codec description, which forced the author to develop new solutions for those modules.

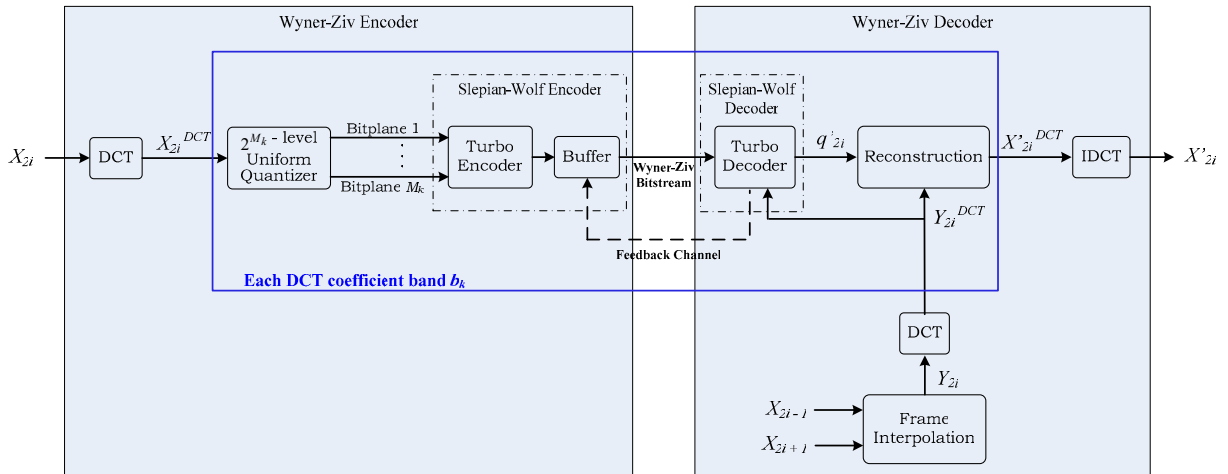


Figure 4.1 – IST-TDWZ codec architecture.

In the following, the coding procedure illustrated in Figure 4.1 will be succinctly described.

- A video sequence is divided into Wyner-Ziv frames (the even frames of the video sequence) and key frames (the odd frames of the video sequence) such as for the IST-PDWZ codec.
- Over each Wyner-Ziv frame  $X_{2i}$ , it is applied a  $4 \times 4$  block-based discrete cosine transform (DCT) as defined by the H.264/MPEG-4 AVC video coding standard [2]; a  $4 \times 4$  DCT transform was chosen in order to allow comparing the results obtained with IST-TDWZ codec and those available in [12].
- The transform coefficients – DCT coefficients – of the entire frame  $X_{2i}$  are then grouped together, according to the position occupied by each DCT coefficient within the  $4 \times 4$  blocks, forming the so-called DCT coefficients bands. Since a  $4 \times 4$  block-based transform is used, there are  $4^2$  possible positions inside a  $4 \times 4$  block and therefore  $4^2$  different DCT coefficients bands can be formed.

- After the transform coding operation, each DCT coefficients band  $b_k$  is uniformly quantized with  $2^{M_k}$  levels (where the number of levels  $2^{M_k}$  varies depending on the DCT coefficients band  $b_k$ ).
- Over the resulting quantized symbol stream (associated to the DCT coefficients band  $b_k$ ), bitplane extraction is performed.
- Each bitplane is then separately turbo encoded. The turbo encoder structure is similar to the one used in the IST-PDWZ implementation (for more details the reader should go to Chapter 3). There are however some differences regarding the interleaver length  $L$  (more generically, the turbo encoder input length) and the puncturing period  $P$ . In the IST-PDWZ solution, the turbo encoder input has the frame size,  $N \times M$ , since in the pixel domain the frame is treated as a whole (see Chapter 3). In the IST-TDWZ solution, the size of each DCT coefficients band is given by the ratio between the frame size  $N \times M$  and the number of different DCT coefficients bands,  $4^2$ ; this ratio corresponds to the turbo encoder input size since each DCT coefficients band is encoded separately.
- The turbo encoder generates redundant (parity) information for each bitplane which is stored in the buffer and sent in chunks (small amounts of parity information) upon decoder request through the feedback channel.
- The parity bits are transmitted according to a pseudo-random puncturing pattern with the same structure as in the IST-PDWZ solution, using however a different puncturing period dynamic range. In the IST-TDWZ implementation, the puncturing period  $P$  ranges from 1 to 48 instead of 1 to 32, as in the IST-PDWZ solution; the higher dynamic range amplitude allows to obtain PSNR values for lower bitrates.
- The decoder performs frame interpolation using the previous and next temporally adjacent frames of  $X_{2i}$  (represented by  $X_{2i-1}$  and  $X_{2i+1}$  in Figure 4.1) to generate an estimate of frame  $X_{2i}$ , called  $Y_{2i}$ . The frame interpolation tools used are equal to the ones employed in the IST-PDWZ implementation (for more details see Chapter 3).
- A block-based  $4 \times 4$  DCT is then carried out over the interpolated frame  $Y_{2i}$  in order to obtain  $Y_{2i}^{DCT}$ , an estimate of  $X_{2i}^{DCT}$ . The residual statistics between correspondent coefficients in  $X_{2i}^{DCT}$  and  $Y_{2i}^{DCT}$  is assumed to be modelled by a Laplacian distribution, as in [12]; the Laplacian parameter is estimated offline for the entire sequence at the DCT band level, i.e. each DCT band has a Laplacian parameter associated.
- Once  $Y_{2i}^{DCT}$  and the residual statistics for a given DCT coefficients band  $b_k$  are known, the decoded quantized symbol stream  $q'_{2i}$  associated to the DCT band  $b_k$  can be obtained through an iterative turbo decoding procedure, similar to the one describe in Chapter 3.
- As in [12], an ideal error detection capability is assumed at the decoder to determine the current bitplane error probability of a given DCT band, i.e. the turbo decoder is able to measure in a perfect way the DCT band current bitplane error probability. The turbo decoding of a DCT band bitplane is considered to be successful if the bitplane error probability is lower than or equal to a given error probability threshold.

- The reconstruction module represented in Figure 4.1 makes use of the  $q'_{2i}$  stream and  $Y_{2i}^{DCT}$  to reconstruct each DCT coefficients band of the  $X_{2i}$  frame.
- After all DCT coefficients bands are reconstructed, a block-based  $4 \times 4$  Inverse Discrete Cosine Transform (represented by the IDCT module in Figure 4.1) is performed and the reconstructed  $X_{2i}$  frame,  $X'_{2i}$ , is obtained.

The main differences between the IST-TDWZ and IST-PDWZ solutions are in the DCT/IDCT, quantization and reconstruction modules. In the following subsections of this Chapter, the implementation of these modules is described in detail.

### 4.1.1 Discrete Cosine Transform in the IST-TDWZ Codec

A block-based transform is employed in the Wyner-Ziv video coding architecture with the same purpose of its usage in traditional video coding schemes: to decorrelate block samples by exploiting the spatial redundancy between neighbouring samples, and to compact the block energy into as few transform coefficients as possible.

The Karhunen-Loève Transform (KLT) is the optimal transform in terms of energy compaction capabilities [54]; however, the KLT transform is signal dependent, i.e. the KLT basis functions are dependent on the signal to be transformed. For a vast set of signals, the discrete cosine transform (DCT) is a close estimate to the KLT transform [55], with the advantage that the DCT basis functions are signal independent. In fact, the DCT transform is widely used by the state-of-the-art traditional video coding standards, from the H.261 to the H.264/MPEG-4 AVC standards [2]).

Generally, the one-dimensional DCT transform converts an  $n \times 1$  samples vector  $\mathbf{x}$  in a new  $n \times 1$  vector  $\mathbf{X}$  of DCT transform coefficients (in the frequency domain) by a linear transformation given by equation (4.1);  $H$  is the  $n \times n$  transformation matrix.

$$\mathbf{X}_{n \times 1} = H_{n \times n} \mathbf{x}_{n \times 1} \quad (4.1)$$

The DCT transform can also be applied to an  $n \times n$  samples matrix; in this case, a two-dimensional DCT transform is employed since there are two dimensions to be considered: the horizontal (along the rows) and the vertical (along the columns) dimensions. The two-dimensional DCT transform is implemented applying a one-dimensional DCT transform twice, one to the horizontal dimension and another to the vertical one [54]. In other words, a  $n \times n$  DCT transform implementation algorithm encloses two steps:

- 1) Each row of the  $n \times n$  samples block is transformed using a one-dimensional DCT transform, characterized by the transformation matrix  $H$ ;
- 2) Each column of the  $n \times n$  block resultant from step 1) is then transformed using a one-dimensional DCT transform characterized by the same transformation matrix.

In the first step, the horizontal correlation within the  $n \times n$  samples block is exploited and in the second step the one-dimensional DCT transform is applied to exploit the vertical correlation.

In the IST-TDWZ architecture, illustrated in Figure 4.1, the first stage towards encoding a Wyner-Ziv frame  $X_{2i}$  is transform coding (represented by the DCT module). The transform employed in the IST-TDWZ solution relies on the  $4 \times 4$  block-based Discrete Cosine Transform, as defined by the H.264/MPEG-4 AVC standard [2], notably in order to allow comparing the IST-TDWZ codec RD performance with the one achieved in [12].

The major characteristics of the  $4 \times 4$  DCT transform used by H.264/MPEG-4 AVC standard are the following:

- It is an integer  $4 \times 4$  block-based transform, i.e. all the operations can be executed using only additions, subtractions and bit-shifts, without accuracy loss.
- All the transform operations can be performed using a 16-bit arithmetic, instead of a 32-bit arithmetic used in a non-integer DCT, reducing the computational complexity.
- The transformation matrix  $H$  is a  $4 \times 4$  matrix defined as [2]:

$$H = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix}. \quad (4.2)$$

- The H.264/MPEG-4 AVC standard specifies completely the inverse transform which ensures that mismatch between different transform implementations does not occur if the specification is followed.
- The inverse transformation matrix  $\tilde{H}_{inv}$  is a  $4 \times 4$  matrix defined as [2]:

$$\tilde{H}_{inv} = \begin{bmatrix} 1 & 1 & 1 & 1/2 \\ 1 & 1/2 & -1 & -1 \\ 1 & -1/2 & -1 & 1 \\ 1 & -1 & 1 & -1/2 \end{bmatrix}. \quad (4.3)$$

Notice that the multiplications by  $\pm 1/2$  in (4.3) can be implemented with 1 bit right-shifts allowing all the decoders to obtain the same results. The relationship between the matrices  $\tilde{H}_{inv}$  and  $H$  is given by equation (4.4), where  $I$  is the Identity matrix.

$$\tilde{H}_{inv} \begin{bmatrix} 1/4 & 0 & 0 & 0 \\ 0 & 1/5 & 0 & 0 \\ 0 & 0 & 1/4 & 0 \\ 0 & 0 & 0 & 1/5 \end{bmatrix} H = I \quad (4.4)$$



In the IST-TDWZ solution, the DCT transform is applied to all  $4 \times 4$  non-overlapping blocks of the  $X_{2i}$  frame, from left to right and top to bottom; the one-dimensional DCT transform is characterized by the transformation matrix given in equation (4.2).

After applying the DCT transform to a  $4 \times 4$  samples block, the 16 correlated samples inside the  $4 \times 4$  block are converted into 16 independent DCT transform coefficients, in the spatial frequency domain. Those DCT coefficients are arranged in a  $4 \times 4$  block called the DCT coefficients block. Figure 4.2 illustrates a  $4 \times 4$  DCT coefficients block; the top-leftmost DCT coefficient is called the DC coefficient and corresponds to the spatial frequency zero. The remaining 15 coefficients are known as AC coefficients and correspond to non-zero spatial frequencies; the AC coefficient located at position 16 corresponds to the highest spatial frequency.

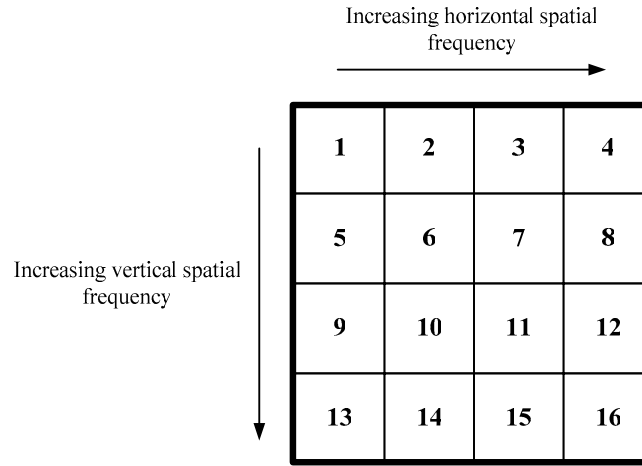


Figure 4.2 – Position ordering inside a  $4 \times 4$  DCT coefficients block.

Once the DCT transform operation has been performed over all the  $4 \times 4$  samples blocks of  $X_{2i}$ , the DCT coefficients are grouped together according to the position occupied by the DCT coefficients within the  $4 \times 4$  DCT coefficients blocks, forming the DCT coefficients bands. In other words, the DCT coefficients band  $b_k$  encloses the transform coefficients of the whole image that occupy the position  $k$  within each  $4 \times 4$  DCT coefficients block. The positions inside a  $4 \times 4$  DCT coefficients block are labelled as shown in Figure 4.2; in fact, any order is possible for the DCT bands numbering since all the DCT bands are encoded and transmitted (which is not always the case, justifying in those situations the usage of zigzag scanning to maximize the subjective impact). The first DCT coefficients band  $b_1$  corresponds to the DC coefficients band and the DCT coefficients band 16 corresponds to the highest AC coefficients band.

At the decoder, an estimate of the  $X_{2i}$  frame, represented in Figure 4.1 by  $Y_{2i}$ , is obtained from the previous and next temporally adjacent frames of  $X_{2i}$  through frame interpolation. The decoder estimate  $Y_{2i}^{DCT}$ , corresponding to  $X_{2i}^{DCT}$ , is then obtained by applying a  $4 \times 4$  DCT transform over the  $Y_{2i}$  frame. The turbo decoder uses then  $Y_{2i}^{DCT}$  to obtain the decoded quantized symbol stream  $q'_{2i}$  associated to the DCT band  $b_k$ .  $Y_{2i}^{DCT}$  is also necessary in the

reconstruction module, together with the  $q'_{2i}$  stream, to help in the DCT coefficients matrix reconstruction task,  $X'_{2i}{}^{\text{DCT}}$ , as will be described in Section 4.1.3.

Since encoded DCT coefficients bands are transmitted to the decoder, the inverse of the DCT operation, known as inverse discrete cosine transform – IDCT, has to be performed at some stage of the Wyner-Ziv decoding procedure in order to obtain the reconstructed  $X_{2i}$  frame,  $X'_{2i}$ . As depicted in Figure 4.1, the IDCT operation is carried out over  $X'_{2i}{}^{\text{DCT}}$ , i.e. the reconstructed matrix of DCT coefficients. The IDCT transform is performed in a similar way to the DCT transform operation, described in this Section; however, instead of using the transformation matrix  $H$ , the IDCT transform employs the inverse transformation matrix  $\tilde{H}_{inv}$  defined by the H.264/MPEG-4 AVC standard and described by equation (4.3).

### 4.1.2 Quantizer in the IST-TDWZ Codec

After the DCT transform operation at the encoder, each DCT coefficients band  $b_k$ , formed as described in Section 4.1.1, is independently encoded.

- Quantization is the first step to encode the DCT coefficients band  $b_k$ , as depicted in Figure 4.1. The IST-TDWZ solution, like the IST-PDWZ one, makes use of a uniform scalar quantizer.
- In the pixel domain Wyner-Ziv video coding (IST-PDWZ) solution, the elements to be quantized are pixel values, namely  $X_{2i}$  frame pixel values; when pixel values are quantized, only positive values are expected at the quantizer input and the dynamic range of such values is fixed and well-known – pixel values vary within the interval  $[0; 255]$  for 8-bit accuracy video data.
- In the IST-TDWZ solution, i.e. a transform domain solution, a DCT coefficients band constitutes the quantizer input.

#### DC Coefficient Quantization

- The DC coefficients band is characterized by high amplitude positive values since each DC transform coefficient expresses the average energy of the corresponding  $4 \times 4$  samples block. Since only positive values are fed into the quantizer, the quantization algorithm for the DC coefficients band can be similar to the one used in the IST-PDWZ implementation. Figure 4.3 illustrates the uniform scalar quantizer used in the DC coefficients quantization procedure;  $\nu$  represents the DC coefficient axis, the numbers 0, 1, 2, 3, ... above the  $\nu$  axis symbolize the quantization intervals index and  $W$  stands for the quantization interval width. Notice that the reconstruction values depend on the reconstruction module at the decoder.

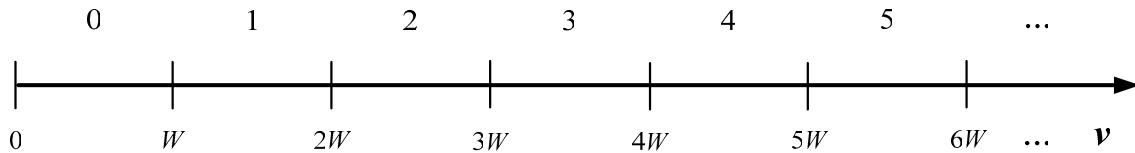


Figure 4.3 – Uniform scalar quantizer with quantization interval width  $W$  for the DC coefficient.

### AC Coefficients Quantization

- For the remaining DCT coefficients bands, called AC coefficients bands, the quantizer input assumes both positive and negative values since the basis functions associated to the AC coefficients present zero mean values [54].
- Figure 4.4 and Figure 4.5 depict the AC coefficients distribution for the bands  $b_2$  and  $b_{16}$ , corresponding to the lowest and the highest spatial frequency AC bands, respectively, of the *Foreman* QCIF video sequence. As it can be observed, the AC coefficients distribution is rather symmetrical around the zero amplitude; this DCT coefficients distribution characteristic occurs not only for the bands  $b_2$  and  $b_{16}$  but also for all the AC bands in between.

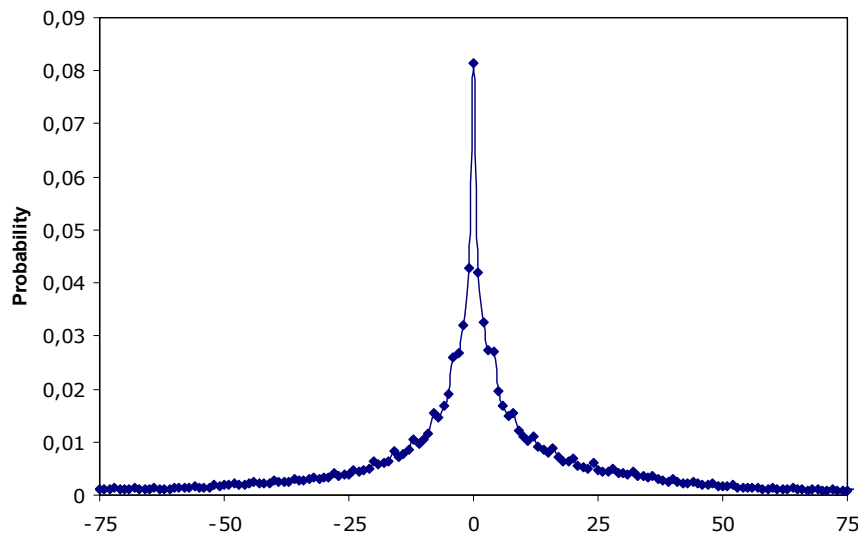


Figure 4.4 – DCT coefficients distribution for the lowest spatial frequency AC band ( $b_2$ ) of the *Foreman* QCIF sequence.

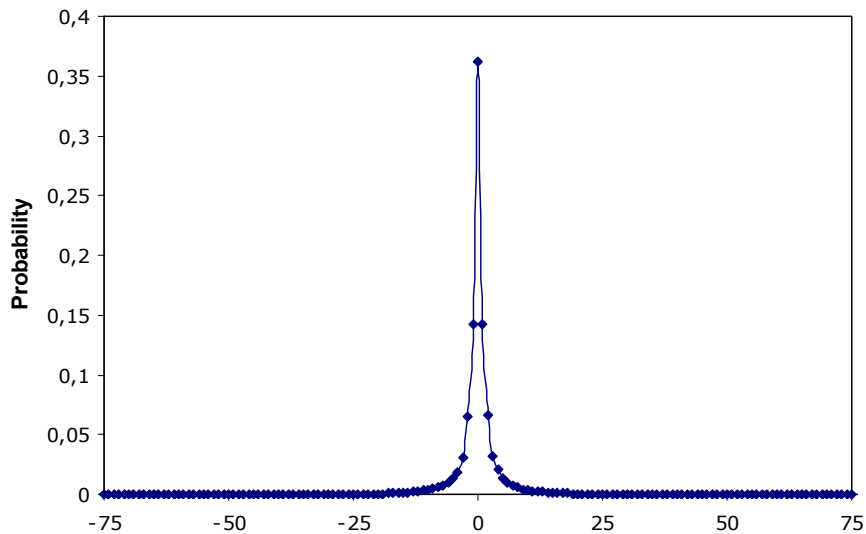


Figure 4.5 – DCT coefficients distribution for the highest spatial frequency AC band ( $b_{16}$ ) of the Foreman QCIF sequence.

- Using a quantizer similar to the one depicted in Figure 4.6 (without a symmetric quantization interval around zero), positive DCT coefficients values are mapped into quantization intervals labelled with indexes greater than or equal to zero and negative DCT coefficients values are mapped into quantization intervals labelled with negative indexes.

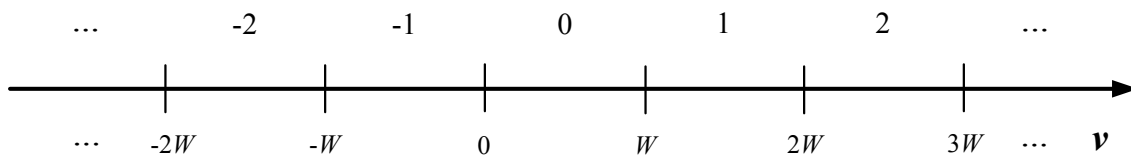


Figure 4.6 – Uniform scalar quantizer without symmetric quantization interval around the zero amplitude.

- As was mentioned in Section 4.1,  $Y_{2i}^{DCT}$  is the decoder best estimate of  $X_{2i}^{DCT}$ ; since  $Y_{2i}^{DCT}$  is an estimate of  $X_{2i}^{DCT}$  not computed from the original, some errors between corresponding coefficients of  $Y_{2i}^{DCT}$  and  $X_{2i}^{DCT}$  may exist.
- If those errors are not corrected through the iterative turbo decoding operation, the annoying block artefact effect becomes visible in the decoded frame  $X'_{2i}$ . In fact, the impact of the errors between  $Y_{2i}^{DCT}$  coefficients and the  $X_{2i}^{DCT}$  corresponding ones is more accentuate for the DCT coefficients around the zero amplitude. In this region, it may happen that a given  $Y_{2i}^{DCT}$  coefficient and the corresponding one in  $X_{2i}^{DCT}$  have different signals, as shown in Figure 4.7; in the Figure 4.7 scenario, the  $Y_{2i}^{DCT}$  coefficient is mapped into the quantization interval -1 and the corresponding coefficient in  $X_{2i}^{DCT}$  is mapped into the quantization interval 0. If after turbo decoding the decoded quantization interval is -1 (the error was not corrected), the block artefact effect will be noticeable in the decoded frame.

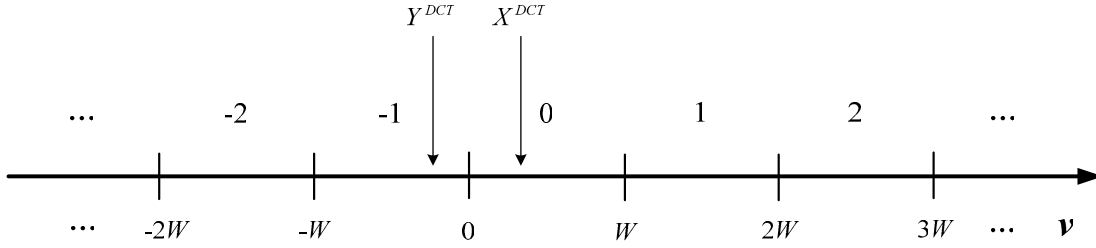


Figure 4.7 – Uniform scalar quantization scenario.

- In order to reduce the block artefacts effect, it may be well-suited to use a quantizer with a quantization interval symmetric around zero, as illustrated in Figure 4.8; low DCT coefficient values around zero are now quantized under the same quantization interval index (independently of its signal) avoiding errors between  $Y_{2i}^{DCT}$  and  $X_{2i}^{DCT}$  corresponding quantized symbols and therefore reducing the annoying block artefact effect.
- Figure 4.8 illustrates a uniform scalar quantizer with a symmetric quantization interval around the zero amplitude;  $W$  is the quantization interval width,  $v$  represents the DCT coefficient values axis and the numbers -2, -1, 0, 1, 2,... above the  $v$  axis indicate the index associated to each quantization interval.

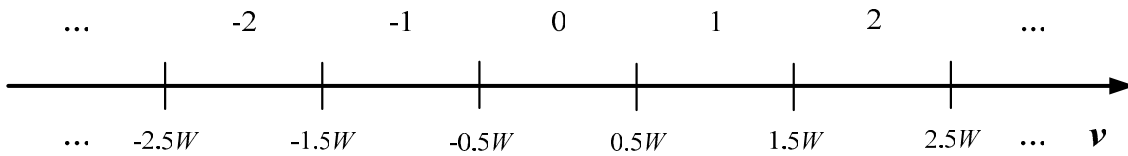


Figure 4.8 – Uniform scalar quantizer with a symmetric quantization interval around the zero amplitude.

- For the case illustrated in Figure 4.8, where all the quantization intervals have the same width  $W$ , the quantization intervals boundaries of the uniform quantizer can be mathematically described by equation (4.5)

$$I_q = \begin{cases} [(q-0.5)W; (q+0.5)W] & q < 0 \wedge q > 0 \\ [-0.5W; 0.5W] & q = 0 \end{cases} \quad (4.5)$$

where  $I_q$  represents the quantization interval boundaries,  $q$  the quantization interval index and  $W$  the quantization interval width; the  $W$  parameter is also known as quantization step size.

### **IST-TDWZ Quantization Approaches**

Thus, two different quantization approaches are followed in the IST-TDWZ codec:

- The DC coefficients band is quantized using a uniform scalar quantizer similar to the one depicted in Figure 4.3.

- The remaining DCT coefficients bands are quantized using the uniform scalar quantizer in Figure 4.8.

### **Number of Quantization Levels**

- The amplitude of the AC coefficients, within a  $4 \times 4$  DCT coefficients block, tends to be higher for the coefficients close to the DC coefficient and to decrease as the coefficients approach the higher spatial frequencies. In terms of AC coefficients bands, this means that AC bands labelled with indexes closer to 1 (index of DC band) enclose values of higher amplitude comparing to AC bands labelled with indexes far away from the DC band. In fact, within a  $4 \times 4$  DCT coefficients block, the lower spatial frequencies enclose more relevant information about the block than the high frequencies, which often correspond to noise or less important details for the human visual system (HVS) [54].
- Since the HVS is more sensitive to lower spatial frequencies, the DCT coefficients representing lower spatial frequencies are quantized using low quantization step sizes, i.e. with a higher number of quantization intervals (levels); the higher spatial frequencies are more coarsely quantized, i.e. with less quantization levels, without significantly decreasing the visual quality of the decoded image. The choice of the number of quantization levels associated to each DCT coefficients band is, therefore, an important way to explore the human visual sensitivity to lower spatial frequencies when compared to higher spatial frequencies.
- The number of quantization levels *per* DCT coefficients band in the IST-TDWZ solution is assumed to be known by the encoder and the decoder.
- To quantize the coefficients of a certain DCT band  $b_k$  it is necessary to have knowledge on its dynamic range, i.e. in which range the DCT coefficients vary, besides the number of quantization levels associated to that band; the quantization step size, necessary to define the quantization intervals bounds, can be computed from these two quantities.
- Letting the decoder know, for each  $X_{2i}$  frame, the dynamic range of each DCT coefficients band instead of using a fixed value allows having quantization interval widths adjusted to the dynamic range of each band. For example, the dynamic range may be lower than the fixed dynamic range selected; since the same number of quantization levels is distributed over a shorter dynamic range, shorter quantization intervals widths can be used. The shorter the quantization step size, the lower is the distortion at the decoder, as will be explained in Section 4.1.3. In the IST-TDWZ codec, the dynamic range for each DCT band is transmitted frame by frame to the decoder and assumed to be error-free received.

### **DC Coefficient Quantization Step Size**

- The quantization step size for the DCT band  $b_k$  results from the division of the  $b_k$  band dynamic range by the  $b_k$  band number of quantization levels  $2^{M_k}$ . The upper bound of the DC coefficient value is given by [56]

$$\sqrt{n^2} I_{\max} \quad (4.6)$$

where  $n^2$  is the number of pixels in a  $n \times n$  pixels block and  $I_{\max}$  stands for the maximum pixel intensity. Thus, for  $4 \times 4$  pixels block and 8-bit accuracy video data, the DC band upper bound, i.e. the DC band dynamic range, is 1024; this value is maintained fixed for all the video frames since  $n$  and  $I_{\max}$  in equation (4.6) are constant throughout the video sequence.

### **AC Coefficients Quantization Step Size**

- To obtain the quantization step size for the AC bands  $b_k$ ,  $k=2, \dots, 16$ , the highest absolute value within each band  $b_k$  is first determined. Since this computation is performed over absolute values, the highest absolute value determined corresponds to half of the total  $b_k$  band dynamic range. The quantization step size  $W$  is then obtained from equation (4.7)

$$W = \frac{2|V_k|_{\max}}{2^{M_k} - 1} \quad (4.7)$$

where  $|V_k|_{\max}$  stands for the highest absolute value within the AC band  $b_k$  and  $2^{M_k}$  represents the  $b_k$  band number of quantization levels.

- The quantization interval index  $q$ , also known as quantized symbol, results from

$$q = \frac{V_k}{W} \quad (4.8)$$

where  $V_k$  is the DCT coefficient value within the DCT coefficients band  $b_k$  and  $W$  is the quantization step size, previously computed, associated to the  $b_k$  band.

In the IST-TDWZ codec, the quantization procedure is as follows:

- Each DCT coefficients band  $b_k$  is quantized using a uniform scalar quantizer with  $2^{M_k}$  levels; the  $M_k$  parameter corresponds to the number of bits required to map each DCT coefficient of band  $b_k$  into one of  $2^{M_k}$  quantizer levels associated to that band.
- Since each  $M_k$  value has a certain rate-distortion point associated to it, different performances can be achieved by changing the  $M_k$  value for the DCT band  $b_k$ .
- In the IST-TDWZ codec performance evaluation, 8 rate-distortion points were considered; each  $4 \times 4$  matrix depicted in Figure 4.9 corresponds to a given rate-distortion point in the IST-TDWZ codec performance.
- The first 7 matrices – (a), (b), ..., (g) – in Figure 4.9 are equal to the ones used in [12] in order to allow comparing the performance of the IST-TDWZ coded and the solution proposed in [12]. The matrix (g), depicted in Figure 4.9, was proposed by the author of this Thesis to verify the IST-TDWZ RD performance for higher bitrates. Matrix (a)

represents the lowest bitrate (and the highest distortion) situation while matrix (h) corresponds to the highest bitrate (and thus lowest distortion) scenario.

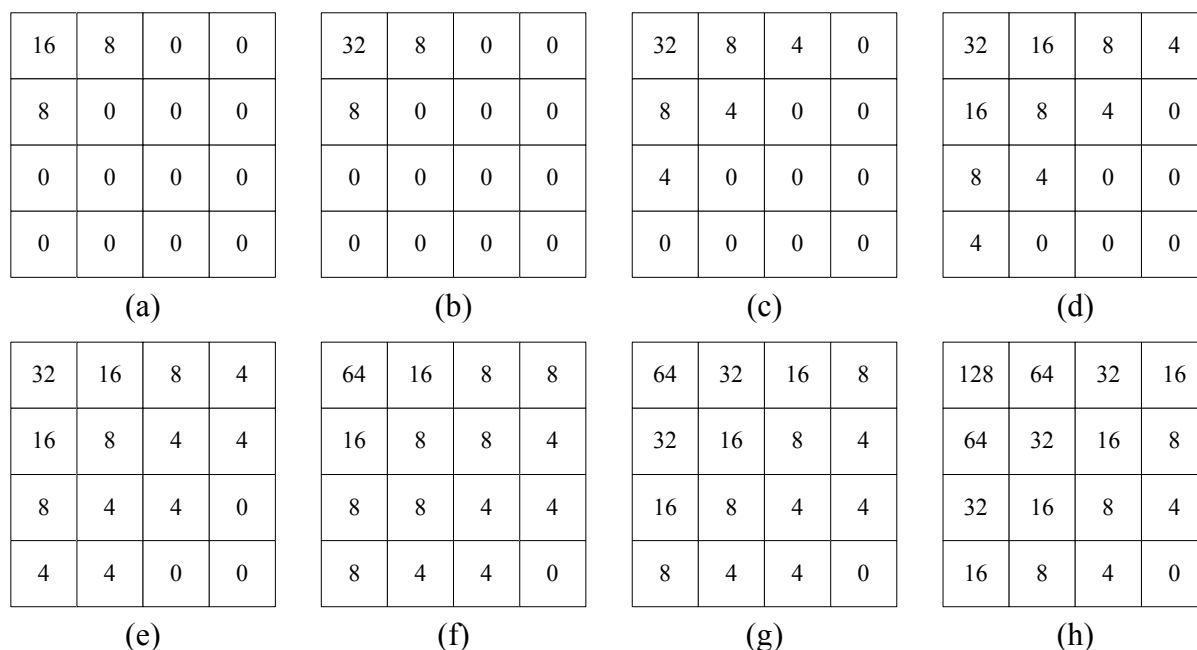


Figure 4.9 – Eight quantization matrices associated to different IST-TDWZ codec rate-distortion performances.

- Within a 4×4 quantization matrix, the value at position  $k$  in Figure 4.9 indicates the number of quantization levels associated to the DCT coefficients band  $b_k$ . The DCT coefficients bands numbering, i.e. the order by which the DCT bands are encoded, is performed as illustrated in Figure 4.2. The quantization matrices depicted in Figure 4.9 are used to determine the rate-distortion performance of the IST-TDWZ codec and are assumed to be known by both the encoder and decoder.
- In Figure 4.9, the value 0 means that no Wyner-Ziv bits are transmitted to the decoder for the corresponding bands; the decoder will replace the DCT bands to which no Wyner-Ziv bits are sent by the corresponding side information DCT coefficients bands determined at the decoder.
- After quantizing the DCT coefficients band  $b_k$ , the quantized symbols (represented by integer values) are converted into a binary stream. The quantized symbols bits of the same significance (e.g. the most significant bit) are grouped together forming the corresponding bitplane array which is then independently turbo encoded. The turbo coding procedure of the DCT coefficients band  $b_k$  starts with the most significant bitplane array, which corresponds to the most significant bits of the  $b_k$  band quantized symbols.
- Since each DCT coefficients band  $b_k$  is independently turbo encoded, the turbo decoding operation is also performed at the DCT coefficients band level. For each DCT coefficients band  $b_k$ , the turbo decoder starts decoding the most significant bitplane array of  $b_k$  band



quantized symbols. For each band  $b_k$ , the number of bitplane arrays to be turbo coded depends on the number of bits needed to map a DCT coefficient value into one of the  $2^{M_k}$  quantization levels associated to the  $b_k$  band.

- The turbo decoder requests for more parity bits, via feedback channel, until the current bitplane error probability is below a given error probability threshold; when this occurs, the turbo decoding operation of the current bitplane array is considered successful.
- After successfully turbo decoding the most significant bitplane array of the  $b_k$  band, the turbo decoder proceeds in an analogous way to the remaining  $M_{k-1}$  bitplanes associated to that band.
- Once all the bitplane arrays of the DCT coefficients band  $b_k$  are successfully turbo decoded, the turbo decoder starts decoding the  $b_{k+1}$  band. This procedure is repeated until all the DCT coefficients bands for which Wyner-Ziv bits are transmitted are turbo decoded.
- Note that, for each Wyner-Ziv frame, the dynamic range of each DCT coefficients band, determined at the quantization stage, is sent to the decoder in order to help in the reconstruction procedure, as will be explained in Section 4.1.3.
- The dynamic range is transmitted in a bitstream header with 16-bits length. In fact, only 10 bits are needed to represent the dynamic range of all DCT bands; however, since the header size is typically a multiple of 8 it has an integer number of bytes; this header marks the beginning of the parity bits transmission for each DCT band. The dynamic range transmission corresponds to a maximum increase of 240 bits per frame, i.e. 15 bands/frame for which the dynamic range value is sent times 16 bits/band (due to packing purposes); no dynamic range value is sent for the DC band.

### 4.1.3 Reconstruction in the IST-TDWZ Codec

After turbo decoding the  $M_k$  bitplanes associated to the DCT band  $b_k$ , the bitplanes are grouped together to form the decoded quantized symbol stream associated to the  $b_k$  band; this procedure is performed over all the DCT coefficients bands to which Wyner-Ziv bits are transmitted.

- Once all the decoded quantized symbol streams are obtained, it is possible to reconstruct the matrix of DCT coefficients,  $X'_{2i}^{\text{DCT}}$  (see Figure 4.1).
- As was mentioned in Section 4.1.2., the DCT coefficients bands to which no Wyner-Ziv bits are sent are replaced by the corresponding DCT bands of the side information,  $Y_{2i}^{\text{DCT}}$ .
- The remaining DCT bands are obtained through turbo decoding procedures, as was described in Section 4.1.2.
- Since it is assumed that the decoder knows the highest absolute value within each AC band (sent by the encoder) and the number of quantization levels for each AC band, the quantization step size for the AC coefficients bands can be easily computed at the

decoder, as described in Section 4.1.2; the highest value within the DC coefficients band is 1024 and is assumed to be fixed (see Section 4.1.2). The quantization step size computed at the decoder is therefore equal to the one at the encoder side, for each DCT coefficients band.

- After the quantization step size is calculated, it is possible to establish the boundaries of the quantization intervals, for a given DCT coefficients band  $b_k$  and reconstruction can be performed.
- The reconstruction procedure for each  $b_k$  band DCT coefficient can be described by one of three cases, as illustrated in Figure 4.10.

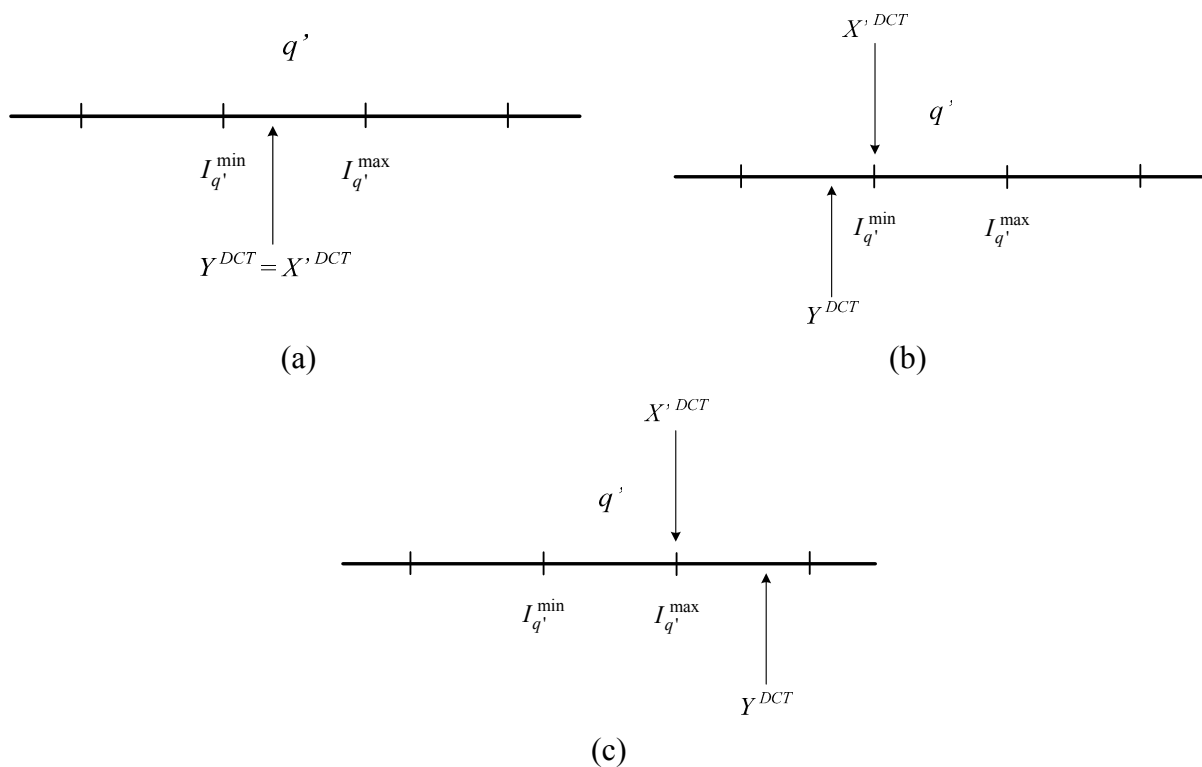


Figure 4.10 – Reconstruction procedure of each  $b_k$  band DCT coefficient: (a) Case I, (b) Case II, (c) Case III.

### Case I

As Figure 4.10 (a) shows, if the side information DCT coefficient  $Y^{DCT}$  is within the turbo decoded quantized symbol  $q'$  then the reconstructed DCT coefficient,  $X', DCT$ , is made equal to the side information DCT coefficient.

### Case II

If the side information DCT coefficient  $Y^{DCT}$  belongs to a quantized symbol lower in magnitude than the turbo decoded one  $q'$ , Figure 4.10 (b), then the reconstructed DCT coefficient  $X', DCT$  assumes the lowest intensity value within the decoded quantized symbol,  $I_{q'}^{\min}$ , i.e. the lower bound of the quantization interval indexed by the decoded quantized symbol.

**Case III**

If side information DCT coefficient  $Y^{DCT}$  belongs to a quantized symbol higher in magnitude than the turbo decoded quantized symbol  $q'$ , Figure 4.10 (c), then the reconstructed DCT coefficient  $X'^{DCT}$  assumes the highest intensity value within the decoded quantized symbol,  $I_q^{\min}$ , i.e. the upper bound of the quantization interval indexed by the decoded quantized symbol.

Since the reconstructed DCT coefficient is between the boundaries of the decoded quantized symbol, the error between DCT coefficients of  $X_{2i}^{DCT}$  and  $X'_{2i}^{DCT}$  (also known as reconstruction distortion) is limited to the quantizer coarseness,  $W$  (see Figure 4.8).





In order to reconstruct the frame  $X'_{2i}$ , the inverse discrete cosine transform (IDCT) must be applied over the reconstructed matrix of DCT coefficients,  $X'_{2i}^{DCT}$  (Figure 4.1), as explained in Section 4.1.1.

## 4.2 IST-TDWZ Experimental Results

In order to evaluate the rate-distortion performance of the IST-TDWZ codec proposed in this Chapter, which architecture is shown in Figure 4.1, four test sequences were considered, which are the same than those considered in the IST-PDWZ performance evaluation.

Table 4.1 provides a brief description of the main characteristics of each test sequence; fps is the abbreviation of frames *per* second.

Table 4.1 – Main characteristics of the video test sequences.

Video Sequence Name	<i>Foreman</i>	<i>Mother and Daughter</i>	<i>Coastguard</i>	<i>Stefan</i>
Sample Frame				
Total Number of Frames	400	961	300	300
Number of Frames Evaluated	101	101	299	299
Spatial Resolution	QCIF	QCIF	QCIF	QCIF
Temporal Resolution (fps)	30	30	25	25

Since the video test sequences evaluated in this Chapter are the same used for the evaluation of the IST-PDWZ codec, the reader should consult Annex A to have a more detailed description of

each test sequence. Some test sequences listed in Table 4.1 are representatives of video conference content, typically characterized by low and medium activity – e.g. amount of movement – e.g. the *Foreman*, the *Mother and Daughter* and the *Coastguard* sequences; the *Stefan* sequence is an example of sports content characterized by higher amount of movement. The higher the activity, the more difficult is to code the video content. This content variety is important to collect enough representative and meaningful results for the IST-TDWZ performance.

Some test conditions are common to all the performance evaluation process; thus, they will be pointed out only once, in order to avoid repetition each time a test sequence is considered. The main common test conditions are listed in the following:

- Only the luminance data is considered in the IST-TDWZ rate-distortion performance evaluation in order to allow comparing the results obtained with the IST-TDWZ codec and those available in [12].
- The key frames, represented in Figure 4.1 by  $X_{2i-1}$  and  $X_{2i+1}$ , are considered to be losslessly available at the decoder.
- The Wyner-Ziv bitstream is assumed to be error-free received, i.e. no errors are introduced during the transmission.
- For the  $X_{2i}$  frame, the dynamic range of each DCT band is assumed to be losslessly available at the decoder.
- The 8 quantization matrices depicted in Figure 4.9 are used to obtain 8 different RD points for the IST-TDWZ solution.
- The turbo encoder implementation is similar to the one used in the IST-PDWZ solution: two recursive systematic convolutional encoders of rate  $\frac{1}{2}$  are employed; each one is represented by the generator matrix  $\begin{bmatrix} 1 & \frac{1+D+D^3+D^4}{1+D^3+D^4} \end{bmatrix}$  (for more details the reader should check Chapter 3).
- The side information is generated using motion compensated frame interpolation algorithms at the decoder. Besides forward and bidirectional motion estimation, a spatial motion smoothing algorithm is used to eliminate motion outliers allowing significant improvements in the RD performance (for more details the reader should check Chapter 3).
- As in [12], a Laplacian distribution models the residual between the  $X_{2i}$  frame DCT coefficients and the corresponding DCT coefficients of the  $Y_{2i}$  frame; thus, it is possible to compare the results obtained with the IST-TDWZ codec and those available in [12].
- Since the Laplacian distribution is at the DCT coefficients band level (see Section 4.1), each DCT coefficients band is characterized by a Laplacian parameter estimated over the total Wyner-Ziv frames number evaluated for a given video sequence. For each sequence, the estimation of the Laplacian distribution parameter is performed offline, i.e. before the Wyner-Ziv coding procedure.

- The maximum allowable turbo decoding iterations number is 18, as in the IST-PDWZ solution; through simulations, it was concluded that 18 iterations allow the turbo decoder to converge.
- The bit error rate threshold, mentioned in Section 4.1, is assumed to be  $1 \times 10^{-3}$ ; the main reason for the choice of this value is related with the possibility of comparing the IST-TDWZ results with those available in [12].
- Since the Wyner-Ziv codec performance is to be evaluated, the rate-distortion plots only contain the rate and the PSNR values for the even frames, i.e. the Wyner-Ziv coded frames, of a given video sequence.
- For each test sequence, the IST-TDWZ rate-distortion performance is compared against H.263+ intraframe coding and H.263+ interframe coding with a I-B-I-B structure. In the last case, only the rate and PSNR of the B frames is shown since, in this Thesis, the Wyner-Ziv codec performance is the target.

In the following subsections, the results obtained with the IST-TDWZ codec for each one of the four QCIF video test sequences, listed in Table 4.1, will be presented and analysed.

#### 4.2.1 *Foreman* Test Sequence Evaluation

In order to be able to compare the IST-TDWZ codec performance with the performance achieved by Aaron *et al.* in [12] for the *Foreman* QCIF under the same conditions, only the first 101 frames of the sequence were considered in the IST-TDWZ codec RD performance evaluation (because this is what is used in [12] although the sequence is longer).

Figure 4.11 shows the IST-TDWZ rate-distortion results obtained for the *Foreman* QCIF test sequence. From Figure 4.11 (a), it is possible to observe that the usage of the transform coding tool in the IST-TDWZ solution provides coding improvements up to 0.6 dB when compared to the IST-PDWZ solution, for the same test conditions. In Figure 4.11 (a), the rate-distortion results achieved in [12] are also plotted for comparison purposes; from the results, it is possible to conclude that the IST-TDWZ codec provides better results when compared to those available in [12], with coding improvements up to 2.1 dB.

The lack of details about the solution proposed in [12] may explain the difference between the two curves (correspondent to the IST-TDWZ solution and the one proposed in [12]); this lack of details led the author of this Thesis to develop tools which may be different from the tools used in [12], making the two coding solutions different although it is not known how much different. From the results depicted in Figure 4.11 (b), it is possible to observe remarkable gains over H.263+ intraframe coding for all bitrates. However, there is still a compression gap when comparing to H.263+ interframe coding with I-B-I-B structure.

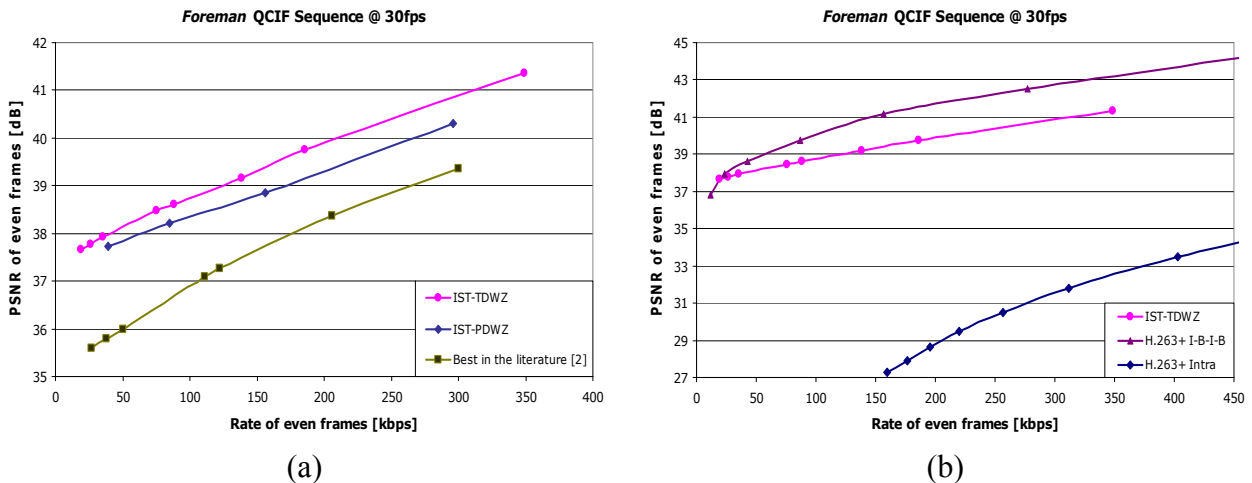


Figure 4.11 – IST-TDWZ rate-distortion performance for the Foreman test sequence.

## 4.2.2 Mother and Daughter Test Sequence Evaluation

The IST-TDWZ codec RD performance was also evaluated using the *Mother and Daughter* test sequence. To be able to compare the IST-TDWZ results with those achieved by Aaron et al. in [12], only the first 101 frames of the *Mother and Daughter* QCIF sequence were considered.

Figure 4.12 shows the IST-TDWZ PSNR results obtained for the *Mother and Daughter* QCIF test sequence. From Figure 4.12 (a), it can be observed that the IST-TDWZ solution provides coding improvements up to 1.3 dB compared to the IST-PDWZ solution, assuming the same test conditions. The rate-distortion results achieved in [12] are also plotted in Figure 4.12. From the results, it can be noticed that the IST-TDWZ codec provides better results when compared to the solution proposed in [12], with coding improvements up to 2 dB.

Again the lack of details about the solution proposed in [12] may explain the difference between the two curves. From the results depicted in Figure 4.12 (b), it is also possible to observe significant gains over H.263+ intraframe coding for all bitrates. However, there is still a compression gap when comparing to H.263+ interframe coding with I-B-I-B structure.

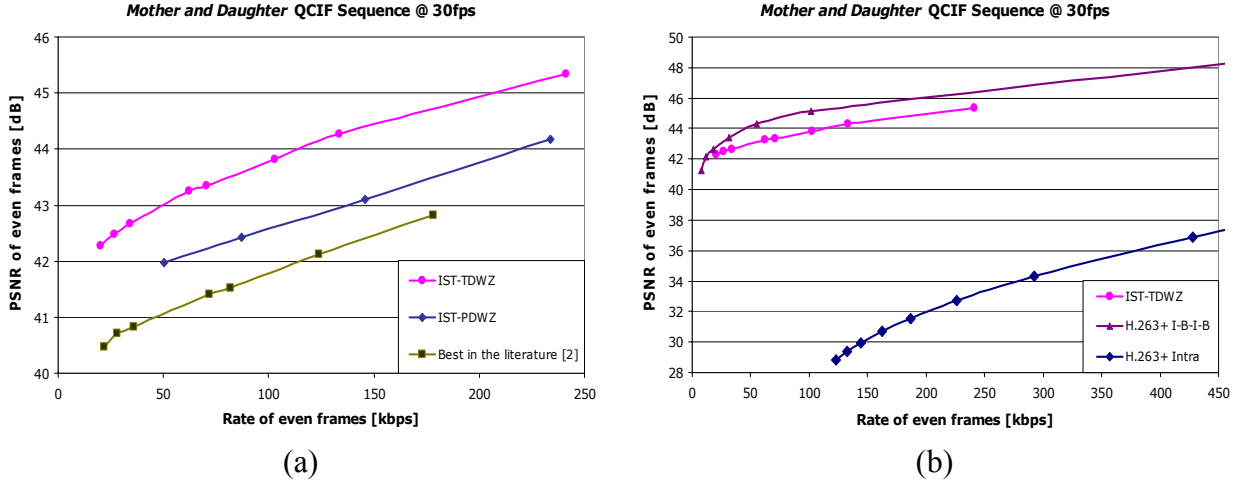


Figure 4.12 – IST-TDWZ rate-distortion performance for the Mother and Daughter test sequence.

### 4.2.3 Coastguard Test Sequence Evaluation

The *Coastguard* test sequence was another sequence for which the IST-TDWZ RD performance was evaluated; in this case, the first 299 frames of the video sequence were considered (see Table 4.1)<sup>1</sup>.

Figure 4.13 shows the IST-TDWZ rate-distortion results obtained for the *Coastguard* QCIF test sequence. As it can be observed from Figure 4.13, the IST-TDWZ solution presents coding improvements up to 1.8 dB comparing to the IST-PDWZ solution, for the same test conditions. From the results depicted in Figure 4.13 (b), it is also possible to observe significant gains over H.263+ intraframe coding for all bitrates. However, there is still a compression gap when compared to H.263+ interframe coding (I-B-I-B structure) for medium and high bitrates. For bitrates lower than 40 kbps, the IST-TDWZ codec RD performance is above the H.263+ with I-B-I-B structure. The *Coastguard* test sequence is characterized by well defined camera motion, basically a panning (see Annex A), which allows to generate, at the decoder, a good estimate of the original frame through the frame interpolation tools. Thus, the amount of parity information needed to correct the “errors” between the original frame and the side information is lower. For this case, for H.263+ interframe coding (I-B-I-B structure), the bitrate cost to transmit error predictions, motion vectors and headers is higher comparing to the amount of parity information required to be sent in the IST-TDWZ solution, in order to achieve the same reconstructed quality.

<sup>1</sup> As was mentioned in Section 4.1, a video sequence is divided into Wyner-Ziv frames (the even frames of the video sequence) and key frames (the odd frames of the video sequence). Since the side information  $Y_{2i}$  for each  $X_{2i}$  frame is generated through frame interpolation from the previous and the next temporally adjacent frames  $X_{2i-1}$  and  $X_{2i+1}$ , an odd number of frames must be considered.

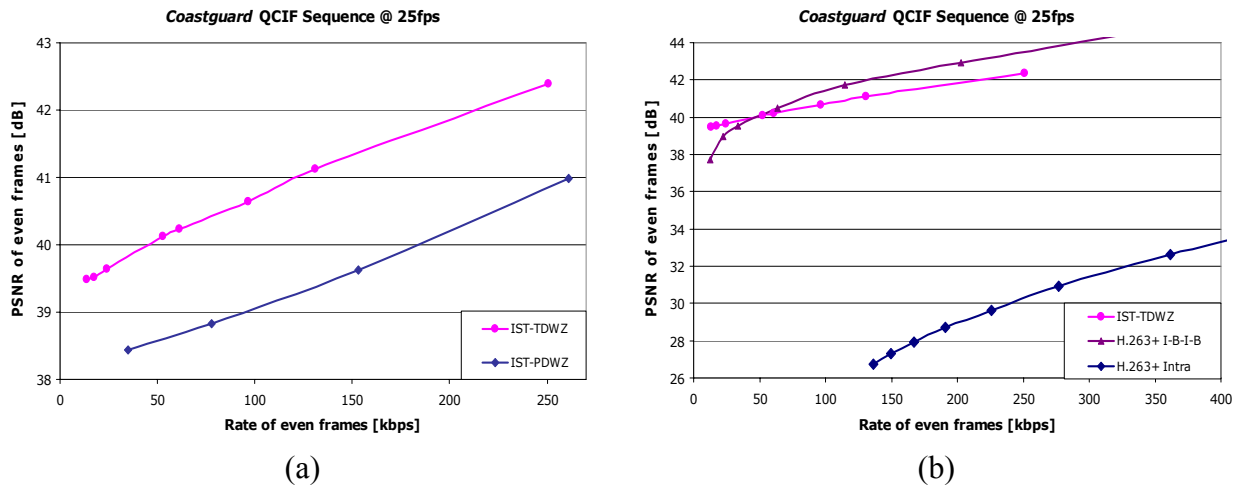


Figure 4.13 – IST-TDWZ rate-distortion performance for the Coastguard test sequence.

#### 4.2.4 Stefan Test Sequence Evaluation

In terms of activity, the *Stefan* QCIF sequence is the fastest test sequence for which the IST-TDWZ codec RD performance was evaluated. As Table 4.1 shows, the first 299 frames of the *Stefan* video sequence were taken into account in the IST-TDWZ codec RD performance evaluation for the same reason mentioned in Section 4.2.3.

Figure 4.14 shows the IST-TDWZ rate-distortion results obtained for the *Stefan* QCIF test sequence. As it can be observed from Figure 4.14, the IST-TDWZ solution presents coding improvements up to 1.7 dB comparing to the IST-PDWZ solution, assuming the same test conditions. From the results depicted in Figure 4.14 (b), it is also possible to notice significant gains over H.263+ intraframe coding for all bitrates. However, there is still a compression gap when comparing to H.263+ interframe coding with I-B-I-B structure.

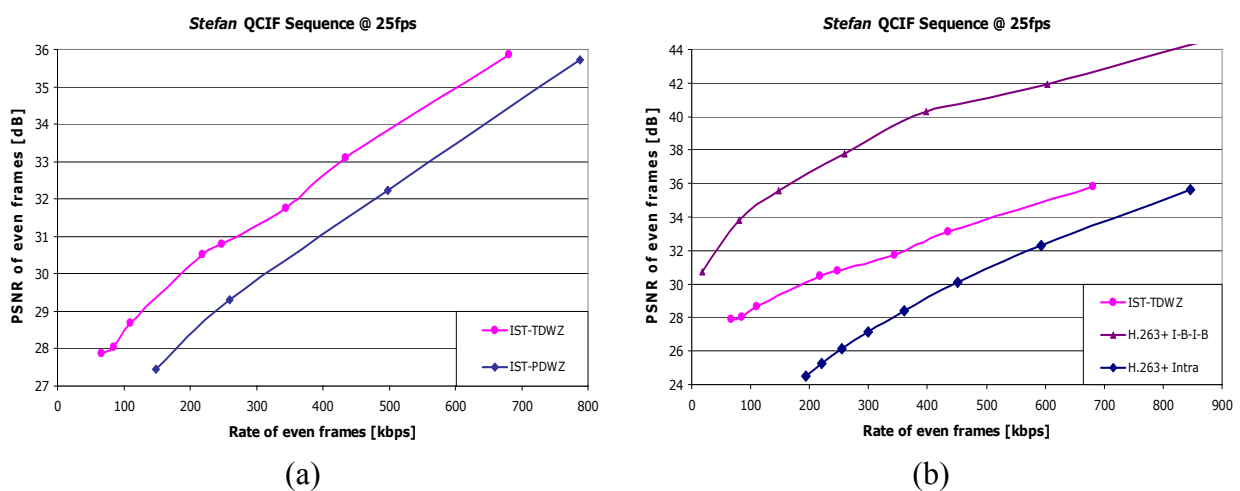


Figure 4.14 – IST-TDWZ rate-distortion performance for the Stefan test sequence.



### 4.2.5 IST-TDWZ versus H.263+ Intraframe Coding Gains

Figure 4.15 shows the IST-TDWZ solution coding gains over H.263+ intraframe coding for the four test sequences evaluated: the *Foreman*, the *Mother and Daughter*, the *Coastguard* and the *Stefan* QCIF video sequences.

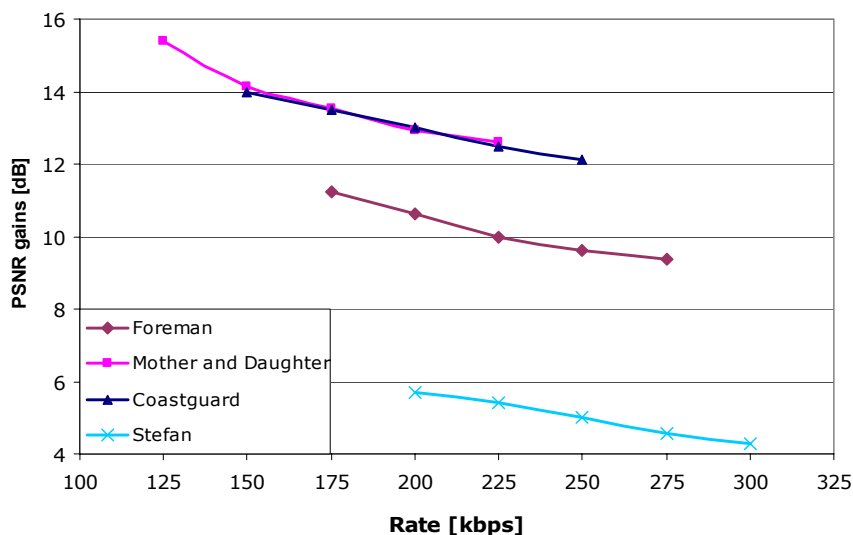


Figure 4.15 – PSNR gains over H.263+ intraframe coding for the *Foreman*, *Mother and Daughter*, *Coastguard* and *Stefan* QCIF video sequences.

As it can be noticed, the IST-TDWZ RD performance is above the H.263+ intraframe coding for all bitrates and test sequences. For fast sequences in terms of activity, like the *Stefan* test sequence, the IST-TDWZ codec presents coding gains up to 5.7 dB regarding the H.263+ intraframe coding. For the video sequences characterized by a lower amount of motion, such as the *Coastguard* and the *Mother and Daughter* test sequences, the IST-TDWZ codec exhibits higher coding gains over the H.263+ intraframe coding (coding gains up to 15.2 dB for the *Mother and Daughter* sequence and 14.1 dB for the *Coastguard* sequence). For the *Foreman* test sequence the IST-TDWZ codec coding gains are up to 11.9 dB.

## 4.3 Final Remarks

In this Chapter, an improved Wyner-Ziv video coding solution, named IST-TDWZ, following the same architecture as the one proposed by Aaron *et al.* in [12] has been presented. The proposed codec has some major differences regarding the codec by Aaron, notably in the Slepian-Wolf codec, quantizer, DCT and frame interpolation modules. These differences are partly motivated by the lack of details regarding the description of the solution proposed in [12] but also related to improvements explicitly introduced.

Section 4.2 presented the RD performance for the IST-TDWZ solution previously proposed. As was mentioned in Section 4.2, the usage of the transform coding tool provides coding improvements up to 1.8 dB when compared to the IST-PDWZ solution. Nevertheless, since a

block-based DCT is used in the IST-TDWZ solution, visually annoying blocking artefacts, especially for low compression factors or lower bitrates, may occur.

For the *Foreman* and *Mother and Daughter* QCIF test sequences, the IST-TDWZ codec RD performance can be directly compared with the solution proposed in [12] (because there are results available). Despite the architectures of the IST-TDWZ codec and the one proposed in [12] being the same, the lack of detail in [12] about some of the tools used forced the author of this Thesis to develop new solutions for some of the architectural modules such as the Slepian-Wolf codec, DCT, quantizer and frame interpolation modules. The new solutions developed for those modules may explain the coding improvements up to 2.1 dB of the IST-TDWZ codec regarding the solution proposed in [12].

From the results presented in Section 4.2, it can be concluded that the IST-TDWZ rate-distortion performance is typically significantly above H.263+ intraframe coding for all bitrates and test sequences. As expected, the higher the amount of motion in the video sequence, the lower the coding gain of the IST-TDWZ codec regarding the H.263+ intraframe coding. There is still a compression gap when the IST-TDWZ codec RD performance is compared to H.263+ interframe coding with I-B-I-B structure; this gap is smaller for sequences with well-defined camera motion like the *Coastguard* sequence, since the interpolation tools can provide better performance for this type of sequences.

## Chapter 5

### Conclusions and Future Work

Distributed Video Coding (DVC) is a new video coding paradigm based on two key Information Theory results for long well-known in the literature: the Slepian-Wolf (1973) [4] and Wyner-Ziv (1976) [6] theorems. This new video coding paradigm enables to explore the signal statistics, partially or totally, at the decoder; in other words, DVC enables to shift complexity from the encoder to the decoder. This shift of complexity should not compromise the coding efficiency, i.e. a Rate-Distortion (RD) performance similar to traditional video coding schemes (where the signal statistics – temporal correlation between adjacent frames – is exploited at the encoder) should be reached. Thus, the new DVC coding paradigm may be described by a configuration where the encoder has low-complexity at the expense of a higher decoder complexity. Since in DVC signal statistics are explored at the decoder, no prediction loop is needed at the encoder side avoiding the interframe error propagation, typical of traditional video coding systems. As a consequence of the prediction loop absence at the encoder, improved error resilience can be achieved in DVC systems in a natural way, i.e. without sending additional information to increase the bitstream error robustness, since there is no error propagation; in traditional video coding schemes, channel coding techniques are employed to make the source encoded bitstream more robust to channel errors.

Typically, in the traditional video coding paradigm Forward Error Correction (FEC) techniques are employed to make the source encoded bitstream more robust to channel errors. In the DVC schemes, no prediction loop is used at the encoder and therefore improved error resilience can be achieved in a more natural way, i.e. without sending additional information to increase the bitstream error robustness. These features make DVC a promising coding solution for some emerging applications such as:

- Wireless low-power surveillance networks.
- Wireless mobile video.
- Multi-view acquisition.
- Video-based sensor networks.

The theory behind this new video coding paradigm and the usage of DVC in the context of those emerging applications were described in Chapter 1.

Efforts towards practical DVC solutions are quite recent; a great part of the work that was developed in the distributed video coding field refers to Wyner-Ziv video coding – a particular case of distributed video coding. In Chapter 2, the most relevant distributed video coding solutions nowadays available in the literature were analyzed; those solutions were developed at the University of Stanford (namely in the Bernd Girod's group) and at the University of California at Berkeley (Kannan Ramchandran's group). The results achieved show that Wyner-Ziv video coding may provide interesting coding solutions for applications where low encoding complexity or robustness to channel transmission errors are the major goals.

Chapter 3 described the IST-PDWZ (from Instituto Superior Técnico-Pixel Domain Wyner-Ziv) solution which follows the architecture of the state-of-the-art pixel domain solution proposed in [17]. The IST-PDWZ is the simplest solution implemented in terms of complexity given that the encoder performs all the processing in the spatial domain (pixel by pixel encoding). Several experiments were performed to evaluate the IST-PDWZ coding efficiency. The results obtained showed coding improvements up to 2.3 dB regarding the more recent pixel domain results for the same type of architecture, published in [12]. Significant coding gains are observed in terms of RD performance regarding H.263+ intraframe coding (for all bitrates and all the test sequences); however there is still a compression gap regarding the H.263+ interframe coding performance with I-B-I-B structure.

The IST-TDWZ (from Instituto Superior Técnico-Transform Domain Wyner-Ziv) solution described in Chapter 4 is an extension of the IST-PDWZ codec and follows the architecture of the state-of-the-art transform domain solution presented in [12]. Comparing to the IST-PDWZ codec, a better rate-distortion performance is achieved (with coding gains up to 1.8 dB) since the H.264/MPEG-4 AVC integer  $4 \times 4$  Discrete Cosine Transform (DCT) is employed to exploit spatial redundancy within each video frame. The performance increase regarding the IST-PDWZ codec comes out at the cost of a higher encoder complexity associated to the transform used; the H.264/MPEG-4 AVC DCT is however one of the most lightweight solutions available today [2]. From the results achieved, coding improvements up to 2.1 dB can be observed regarding the solution proposed in [12] with a similar architecture. The IST-TDWZ codec RD performance is considerably above the H.263+ intraframe coding performance (for all bitrates and all the test sequences); however a compression gap still exists regarding the H.263+ interframe coding performance with I-B-I-B structure.

Despite the IST-PDWZ and IST-TDWZ architectures being based on the state-of-the-art pixel domain and transform domain architectures presented in [17] and [12], respectively, the

implementation of both solutions (encoder/decoder) was entirely performed by the author of this Thesis. The lack of details in [17] and [12] about some architectural modules (essentially the Slepian-Wolf codec and frame interpolation module) forced the author of this Thesis to develop new solutions for those modules. The new tools developed may explain the coding improvements for the IST-PDWZ and IST-TDWZ codecs regarding the state-of-the-art solutions which were taken as inspiration.

The main contributions of this Thesis to the distributed video coding field are the frame interpolation tools at the decoder and the Slepian-Wolf codec. Regarding the frame interpolation tools, a block-based framework was proposed based on forward motion estimation and refinement (bidirectional motion estimation) along with spatial motion smoothing based estimation to correct possible frame interpolation errors; this framework allowed to improve the RD performance of the IST-PDWZ and IST-TDWZ solutions by generating at the decoder a better estimate of the side information from temporally adjacent frames. Notice that the same frame interpolation framework was used both in IST-PDWZ and IST-TDWZ solutions.

In the Slepian-Wolf coding context, the main contributions regard the virtual channel statistics modeling in order to accurately estimate the error distribution between the side information and the original frame. Typically, the mathematical formalism associated with turbo codes is adapted to a Gaussian distribution both for the parity and systematic information. During this work, it was concluded through simulations that the error distribution between the side information and the frame to be encoded is better approximated by a Laplacian distribution. Thus, the mathematical formalism associated with turbo codes was modified to a Laplacian distribution. Moreover, since an error-free transmission channel is assumed both in the IST-PDWZ and IST-TDWZ solutions, the parity information distribution was approximated by a Gaussian distribution with a small variance, for the reasons mentioned in Chapter 3. Puncturing techniques were also developed in order to adjust the turbo codec rate in a flexible way.

The work presented in this Thesis or somehow related to it has been presented in one national conference publication [57] and five international conference publications [58]-[62]. Some of the publications resulted from a joint collaboration with *Politecnico di Milano* (Italy) in the framework of the Network of Excellence VISNET (networked audiovisual media technologies); in fact, that work is not described in this Thesis although it is a direct consequence of it since the software developed here was instrumental to quickly try novel tools and reach promising results. Another result of this joint collaboration was a submission to a highly reputed journal [63].

## 5.1 Future Work

Practical efforts towards distributed video coding solutions are nowadays in its infancy. However there is a rather quickly growing interest by the video coding research community on this topic as may be seen in recent conferences and workshops.

In the context of this Thesis, the new algorithms developed and evaluated were just the first effort by the author to bring some advances to the distributed video coding field. The algorithms developed allow reducing the performance gap of Wyner-Ziv video coding when compared to the traditional video coding systems; however considerably work still needs to be done in order to achieve the compression efficiency of the state-of-the-art traditional video coding standards (e.g. the ITU-T H.264/MPEG-4 AVC standard [2]). In this context, some possible future directions, extending the work described in this Thesis, are presented in the following:

- ◆ **Moving from lossless to lossy key frames:** The solutions developed in this Thesis rely on the assumption that the side information is generated using key frames perfectly available at the decoder, i.e. lossless key frame coding is assumed. In fact, depending on the target quality, a huge bitrate would be needed to provide such perfect key frame reconstruction at the decoder; abrupt variations on the decoded video quality would also be noticed (which is not visually pleasant for the user) since each Wyner-Ziv frame would very likely be encoded with a lower quality when compared with the key frames quality. In this context, the next step is to leave the lossless key frame coding assumption towards the lossy key frame realistic scenario. With lossy key frame coding, it will be necessary to determine the new correlation model between the side information (generated using lossy coded key frames) and the original frame. It will also be necessary to study the impact of the generated side information in the decoded video quality, within this more realistic scenario. Some work in this direction has already been developed in the context of the collaboration with *Politecnico di Milano* (Italy) [61] [62] [63].
- ◆ **More accurate motion interpolation and extrapolation techniques at the decoder:** In a scenario characterized by a very lightweight encoder, the time consuming motion estimation/compensation task needs to be shifted to the decoder. Several frame interpolation techniques can be employed at the Wyner-Ziv decoder to generate the side information. The choice of the technique (or the appropriate set of techniques) used can significantly influence the Wyner-Ziv codec rate-distortion performance; more accurate side information through frame interpolation means that the side information  $Y$  is more similar to the original frame  $X$  and therefore the decoder needs less bits from the encoder (to correct the errors between  $Y$  and  $X$ ) and thus the bitrate is reduced for the same quality. In order to obtain the same performance than the traditional video coding schemes, more powerful motion estimation and compensation techniques are necessary, preferably as efficient as the powerful tools included in the latest H.264/MPEG-4 AVC standard, where multi-frame prediction, variable block size motion compensation, and  $\frac{1}{4}$  pixel motion precision account for most of the performance gains. However, the traditional motion estimation and compensation techniques used at the encoder for hybrid video coding are not fully adequate to perform frame interpolation since they attempt to choose the best prediction for the current (known) frame in the rate-distortion sense. For frame interpolation, it is essential to find an estimate (or a guess) of the current frame

which makes the problem significantly different (the current frame is not available) and therefore new and improved techniques are needed.

Since frame interpolation is performed based on past and future frames, this implies that the decoding order is not equal to the presentation order and introduces an extra delay; the extra amount of delay depends on how far the future available reference is. For some types of applications, this is not acceptable (e.g. videoconferencing) and motion extrapolation techniques have to be used this means no future frames are used. Motion extrapolation is a more challenging task, since it relies only on past decoded frames and is not possible to obtain a motion trajectory between future and past frames (and thus more precise) as occurs with motion interpolation techniques.

- ◆ **Virtual channel statistics modelling:** An estimate (side information) of the original frame is generated at the decoder. In order to make use of the side information, the decoder first needs to know the correlation between the side information and the original frame; this corresponds to a virtual channel where errors occur according to some distribution. However, typical channel models (e.g. Gaussian) are not adequate to the error patterns of the side information and new models are necessary for DVC, along with the techniques to estimate the model parameters. Since the accuracy of the side information may change along time and space, techniques that estimate the error distribution spatially and temporally may improve the accuracy of the model and thus the efficiency of the video codec.
- ◆ **Rate control at the encoder:** Another challenge in the distributed video coding architecture adopted in this Thesis is to perform rate control at the encoder while still maintaining a low encoding complexity. This will avoid the use of the feedback channel in the current architecture and it will open the possibilities for new applications (e.g. broadcasting) where this channel is not physically available. This is however a challenging task, since the encoder does not know the quality of the side information (obtained at the decoder) and therefore can only estimate the bitrate needed to achieve a certain decoded image quality. Since the rate-distortion curve that helps to make this decision it is not known by the encoder, techniques to model the side information quality and perform rate decision are necessary in a practical coding scheme without feedback channel. Also regarding this topic, other encoder control decisions such as mode decision are also needed, e.g. the encoder must decide which frames or blocks are intra encoded, i.e. like in traditional video coding schemes, and which are encoded in a distributed way. If no intra coding mode decision is used when the side information and the frame to be encoded have a weak correlation, e.g. scene cuts, uncovered areas, the encoder needs to send a high amount of bits in order to decode the bitplanes sent. When low temporal correlation exists, intra coding provides better performance since exploiting low temporal correlation will not bring any coding efficiency. The mode decision burden should be minimal in order to not compromise the encoder complexity.

- ◆ **Channel codes:** The channel codes are a very important tool in DVC in order to correct the errors (that change over time) in the side information. Thus it is important to design channel codes adequate to the distributed video coding scenario (source coding). Some of the properties looked for are: i) coding of integer-valued sources with a high dynamic range, e.g. transform coefficients, ii) to work well under high compression ratios, i.e. highly punctured, iii) rate adaptation with minimal complexity when the source correlations change, and iv) performance close to the information theory bound, i.e. the Shannon limit.

Concluding, distributed video coding adopts a completely different coding paradigm by giving the decoder the task to exploit - partly or entirely - the source statistics to achieve efficient compression of the video signal. This new paradigm moves the bulk of the complexity from the encoder to the decoder, allowing the provision of efficient compression solutions with simple encoders and complex decoders. Therefore, it is a strong candidate for some emerging applications, e.g. wireless video, sensor networks, disposable cameras, etc. In this Thesis a major contribution was made to bring the coding efficiency of distributed video coding schemes nearer to hybrid video coding schemes thus paving the way for a breakthrough regarding the next video coding generation.



# Annex A

## Video Test Sequences

This Annex provides a brief description of the test sequences used to evaluate the rate-distortion performance of the IST-PDWZ and IST-TDWZ codecs. The reader can use this Annex as a reference when details on each sequence are needed. Table A.1 summarizes the main characteristics of each test sequence.

*Table A.1 – Main characteristics of the test sequences.*

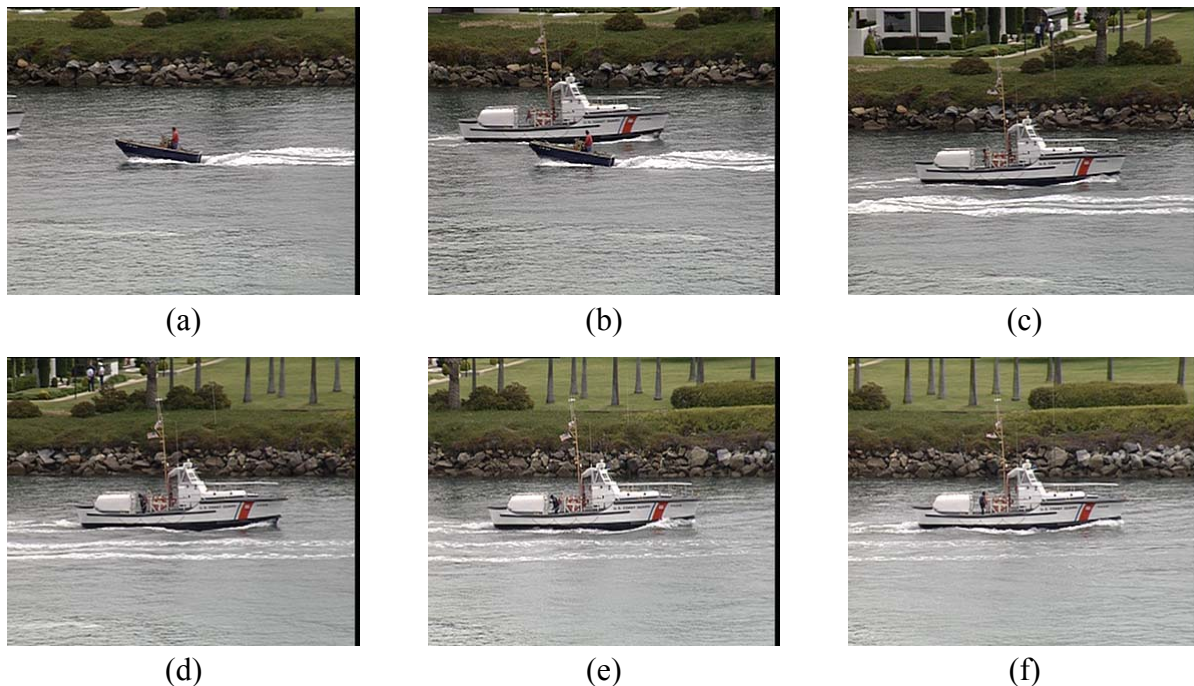
<b>Video Sequence Name</b>	<b><i>Foreman</i></b>	<b><i>Mother and Daughter</i></b>	<b><i>Coastguard</i></b>	<b><i>Stefan</i></b>
<b>Number of Frames</b>	400	961	300	300
<b>Spatial Resolution</b>	QCIF	QCIF	QCIF	QCIF
<b>Temporal Resolution (fps)</b>	30	30	25	25
<b>Source</b>	MPEG	MPEG	MPEG	MPEG

The four test sequences have been used in the context of the Motion Picture Experts Group (MPEG) group of the International Organization for Standardization/International Electrotechnical Commission (ISO/IEC). Note that, in order to be able to compare the results obtained in this Thesis with the ones available in the literature [12], the *Foreman* and the *Mother and Daughter* test sequences had been obtained from [64].

The test sequences had been chosen in order to include low, medium and high activity (amount of movement), highly and poorly textured images, camera pan, solid and non-solid objects, faces, landscapes, water, etc. This content variety corresponds to different coding difficulties. In the following subsections, a more detailed description of each test sequence is presented, in alphabetical order.

## **A.1 Coastguard Sequence**

The *Coastguard* sequence is characterized by a camera following a small boat which is moving to the left direction (pan-left) until a bigger boat appears on the left side. In that point, the camera moves up quickly (fast tilt up) and starts following the bigger boat towards the right direction (pan-right). The objects present in the scene, such as the boats are characterized by a well defined activity (motion). Some frames of this sequence are depicted in Figure A.1, with a temporal spacing of 60 frames.



*Figure A.1 – Coastguard sequence: (a) frame 0; (b) frame 60; (c) frame 120; (d) frame 180; (e) frame 240; (f) frame 299.*

## A.2 Foreman Sequence

This sequence can be clearly divided in two parts: a typical video-telephony scene where the terminal is on the hand of the speaker, followed by a fast change (with horizontal panning) to a scenario with a building under construction. During the first scene, the camera movement is reduced; however, the speaker shakes and moves his head, coming close and moving away from the camera. In the second part of the sequence, the camera movement is considerable, characterized by a pan-left combined with tilt-down. In this part, the objects do not present any movement. Some frames of the *Foreman* sequence are depicted in Figure A.2, with a temporal spacing of 80 frames.

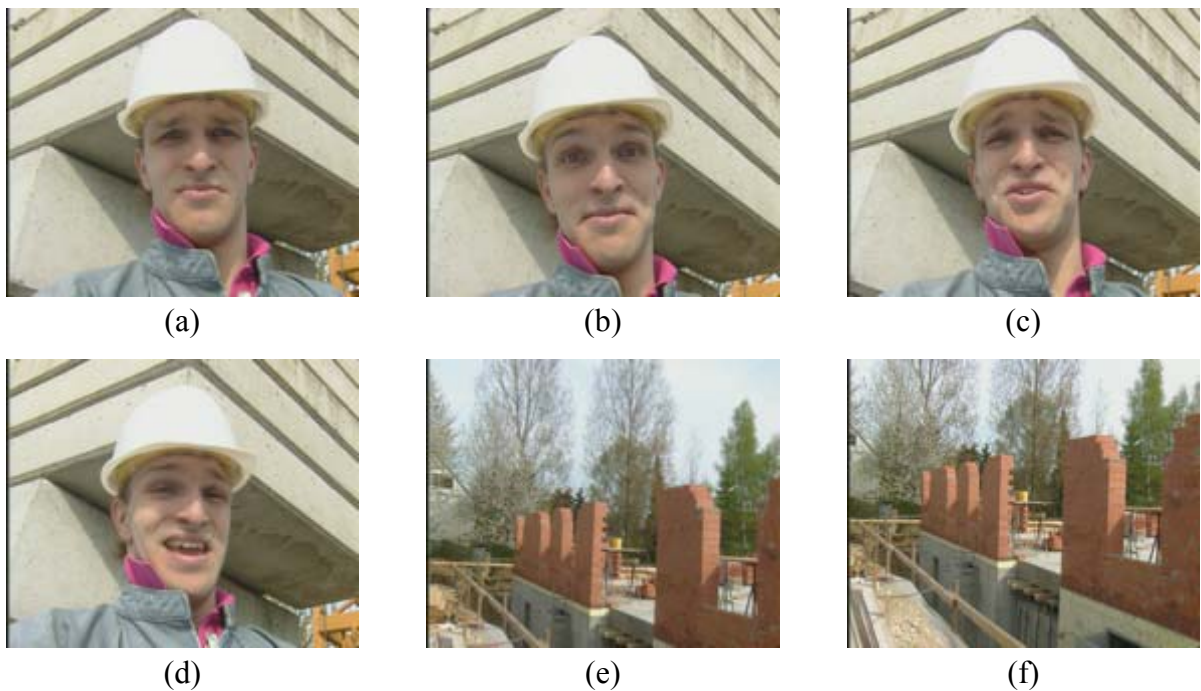
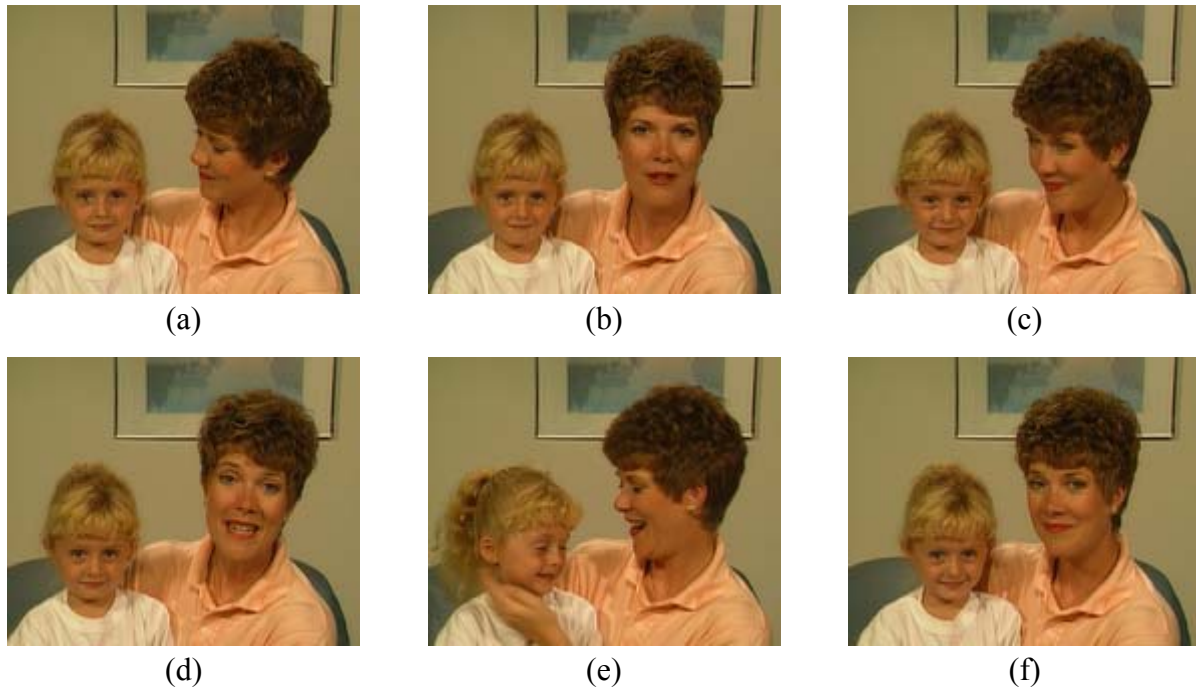


Figure A.2 – Foreman Sequence: (a) frame 0; (b) frame 80; (c) frame 160; (d) frame 240; (e) frame 320; (f) frame 399.

### **A.3 Mother and Daughter Sequence**

In the *Mother and Daughter* sequence, a lady and a child (supposedly her daughter) are speaking to the camera. During the sequence, both the mother and the daughter faces move, however the amount of motion is low. This sequence exhibits some wide homogeneous areas (background) and some textured areas, such as the lady's hair. Some frames of this sequence are depicted in Figure A.3, with a temporal spacing of 192 frames.



*Figure A.3 – Mother and Daughter sequence: (a) frame 0; (b) frame 192; (c) frame 384; (d) frame 576; (e) frame 768; (f) frame 960.*



## A.4 Stefan Sequence

The *Stefan* sequence is a fast sequence, in terms of activity. In this sequence, a camera follows a tennis player moving in the field in all directions. In the background, there is the public which correspond to an area with low activity but highly textured. The camera movements are mainly in the horizontal direction and the tennis player movement is highly complex during the sequence. Some frames of this sequence are depicted in Figure A.4, with a temporal spacing of 60 frames.

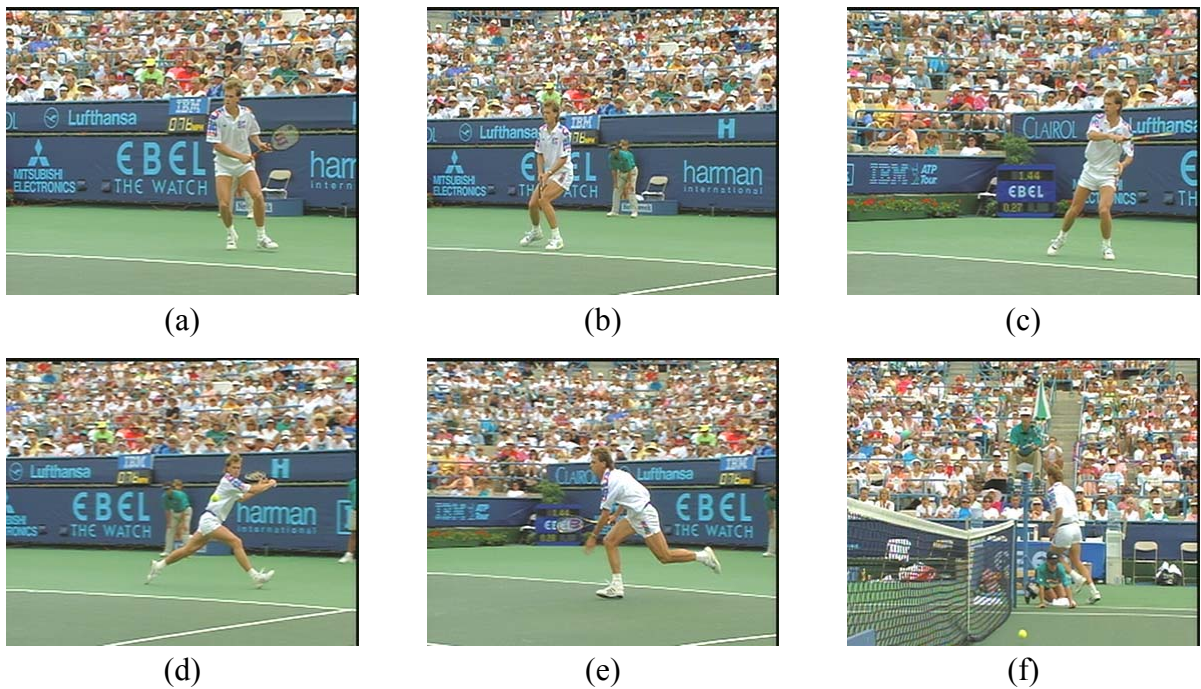


Figure A.4 – Stefan sequence: (a) frame 0; (b) frame 60; (c) frame 120; (d) frame 180; (e) frame 240; (f) frame 299.



# References

## Chapter 1

- [1] T. Wiegand, G. Sullivan, G. Bjøntegaard and A. Luthra, “Overview of the H.264/AVC Video Coding Standard”, *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 13, No. 7, pp. 560-576, July 2003.
- [2] ISO/IEC International Standard 14496-10:2003, “Information Technology – Coding of Audio-visual Objects – Part 10: Advanced Video Coding”.
- [3] J. L. da Silva Jr., J. Schamberger, M. J. Ammer, C. Guo, S. Li, R. Shah, T. Tuan, M. Sheets, J. M. Rabaey, B. Nikolic, A. Sangiovanni-Vincentelli and P. Wright, “Design Methodology for PicoRadio Networks”, *Proceedings of IEEE Design, Automation and Test in Europe*, Munich, Germany, pp. 314-323, March 2001.
- [4] J. Slepian and J. Wolf, “Noiseless Coding of Correlated Information Sources”, *IEEE Transactions on Information Theory*, Vol. 19, No. 4, pp. 471-480, July 1973.
- [5] A. Wyner, “Recent Results in the Shannon Theory”, *IEEE Transactions on Information Theory*, Vol. 20, No. 1, pp. 2-10, January 1974.
- [6] 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, No. 1, pp. 1-10, January 1976.
- [7] R. Zamir, “The Rate Loss in the Wyner-Ziv Problem”, *IEEE Transactions on Information Theory*, Vol. 42, No. 6, pp. 2073-2084, November 1996.
- [8] B. Vucetic and J. Yuan, “Turbo Codes Principles and Applications”, Kluwer Academic Publishers, USA, 2000.

## **Chapter 2**

- [9] J. Bajcsy and P. Mitran, "Coding for the Slepian-Wolf Problem with Turbo Codes", *Proceedings of IEEE Global Telecommunications Conference*, Vol. 2, pp. 1400-1404, San Antonio, Texas, USA, November 2001.
- [10] A. Liveris, Z. Xiong and C. Georghiades, "Compression of Binary Sources with Side Information at the Decoder Using LDPC Codes", *IEEE Communications Letters*, Vol. 6, No. 10, pp. 440-442, October 2002.
- [11] S. Pradhan and K. Ramchandran, "Distributed Source Coding Using Syndromes (DISCUS): Design and Construction", *Proceedings of IEEE Data Compression Conference*, pp. 158-167, Snowbird, Utah, USA, March 1999.
- [12] A. Aaron, S. Rane, E. Setton and B. Girod, "Transform-Domain Wyner-Ziv Codec for Video", *Proceedings of SPIE Visual Communications and Image Processing Conference*, San Jose, California, USA, January 2004.
- [13] S. Pradhan and K. Ramchandran, "Enhancing Analog Image Transmission Systems Using Digital Side Information: A New Wavelet-Based Image Coding Paradigm", *Proceedings of IEEE Data Compression Conference*, pp. 63-72, Snowbird, Utah, USA, March 2001.
- [14] A. Liveris, Z. Xiong and C. Georghiades, "A Distributed Source Coding Technique for Correlated Images Using Turbo Codes", *IEEE Communications Letters*, Vol. 6, No. 9, pp. 379-381, September 2002.
- [15] A. Jagmohan, A. Sehgal and N. Ahuja, "Predictive Encoding Using Coset Codes", *Proceedings of IEEE International Conference on Image Processing*, Vol. 2, pp. 29-32, Rochester, New York, USA, September 2002.
- [16] R. Puri and K. Ramchandran, "PRISM: A New Robust Video Coding Architecture Based on Distributed Compression Principles", *Proceedings of 40<sup>th</sup> Allerton Conference on Communication, Control and Computing*, Allerton, Illinois, USA, October 2002.
- [17] A. Aaron, R. Zhang and B. Girod, "Wyner-Ziv Coding for Motion Video", *Proceedings of Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, California, USA, November 2002.
- [18] X. Zhu, A. Aaron and B. Girod, "Distributed Compression for Large Camera Arrays", *IEEE Workshop on Statistical Signal Processing*, pp. 30-33, St. Louis, Missouri, USA, September 2003.
- [19] A. Aaron, S. Rane and B. Girod, "Wyner-Ziv Video Coding with Hash-Based Motion Compensation at the Receiver", *Proceedings of IEEE International Conference on Image Processing*, Vol. 5, pp. 3097-3100, Singapore, October 2004.
- [20] S. Rane, A. Aaron and B. Girod, "Systematic Lossy Forward Error Protection for Error-Resilient Digital Video Broadcasting", *Proceedings of SPIE Visual Communications and Image Processing Conference*, San Jose, California, USA, January 2004.



- [21] D. Rebollo-Monedero, A. Aaron and B. Girod, "Transforms for High-Rate Distributed Source Coding", *Proceedings of Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, California, USA, November 2003.
- [22] T. Flynn and R. Gray, "Encoding of Correlated Observations", *IEEE Transactions on Information Theory*, Vol. 33, No. 6, pp. 773-787, November 1987.
- [23] S. Shamai, S. Verdú and R. Zamir, "Systematic Lossy Source/Channel Coding", *IEEE Transactions on Information Theory*, Vol. 44, No. 2, pp. 564-579, March 1998.
- [24] S. Servetto, "Lattice Quantization with Side Information", *Proceedings of IEEE Data Compression Conference*, pp. 510-519, Snowbird, Utah, USA, March 2000.
- [25] M. Fleming and M. Effros, "Network Vector Quantization", *Proceedings of IEEE Data Compression Conference*, pp. 13-22, Snowbird, Utah, USA, March 2001.
- [26] D. Muresan and M. Effros, "Quantization as Histogram Segmentation: Globally Optimal Scalar Quantizer Design in Network Systems", *Proceedings of IEEE Data Compression Conference*, pp. 302-311, Snowbird, Utah, USA, April 2002.
- [27] M. Effros and D. Muresan, "Codecell Contiguity in Optimal Fixed-Rate and Entropy-Constrained Network Scalar Quantizers", *Proceedings of IEEE Data Compression Conference*, pp. 312-321, Snowbird, Utah, USA, April 2002.
- [28] D. Rebollo-Monedero, R. Zhang and B. Girod, "Design of Optimal Quantizers for Distributed Source Coding", *Proceedings of IEEE Data Compression Conference*, pp. 13-22, Snowbird, Utah, USA, March 2003.
- [29] X. Wang and M. Orchard, "Design of Trellis Codes for Source Coding with Side Information at the Decoder", *Proceedings of IEEE Data Compression Conference*, pp. 361-370, Snowbird, Utah, USA, March 2001.
- [30] J. García-Frías, "Compression of Correlated Binary Sources Using Turbo Codes", *IEEE Communications Letters*, Vol. 5, No. 10, pp. 417-419, October 2001.
- [31] J. García-Frías and Y. Zhao, "Data Compression of Unknown Single and Correlated Binary Sources Using Punctured Turbo Codes", *Proceedings of 39<sup>th</sup> Allerton Conference on Communication, Control and Computing*, Monticello, Illinois, USA, October 2001.
- [32] P. Mitran and J. Bajcsy, "Near Shannon-Limit Coding for the Slepian-Wolf Problem", *Proceedings of 21<sup>st</sup> Biennial Symposium on Communications*, Kingston, Ontario, Canada, June 2002.
- [33] A. Aaron and B. Girod, "Compression with Side Information Using Turbo Codes", *Proceedings of IEEE Data Compression Conference*, pp. 252-261, Snowbird, Utah, USA, April 2002.
- [34] A. Liveris, Z. Xiong and C. Georghiades, "Compression of Binary Sources with Side Information Using Low-Density Parity-Check Codes", *Proceedings of IEEE Global Telecommunications Conference*, Vol. 2, pp. 1300-1304, Taipei, Taiwan, November 2002.

- [35] A. Liveris, Z. Xiong and C. Georghiades, “Joint Source-Channel Coding of Binary Sources with Side Information at the Decoder Using IRA Codes”, *Proceedings of IEEE Multimedia Signal Processing Workshop*, pp. 53-56, St. Thomas, US Virgin Islands, December 2002.
- [36] A. Liveris, Z. Xiong and C. Georghiades, “Distributed Compression of Binary Sources Using Conventional Parallel and Serial Concatenated Convolutional Codes”, *Proceedings of IEEE Data Compression Conference*, pp. 193-202, Snowbird, Utah, USA, March 2003.
- [37] V. Stankovic, A. Liveris, Z. Xiong and C. Georghiades, “Design of Slepian-Wolf Codes by Channel Code Partitioning”, *Proceedings of IEEE Data Compression Conference*, pp. 302-311, Snowbird, Utah, USA, March 2004.
- [38] D. Schonberg, S. Pradhan and K. Ramchandran, “Distributed Code Constructions for the Entire Slepian-Wolf Rate Region for Arbitrarily Correlated Sources”, *Proceedings of IEEE Data Compression Conference*, pp. 292-301, Snowbird, Utah, USA, March 2004.
- [39] T. Coleman, A. Lee, M. Medard and M. Effros, “On Some New Approaches to Practical Slepian-Wolf Compression Inspired by Channel Coding”, *Proceedings of IEEE Data Compression Conference*, pp. 282-291, Snowbird, Utah, USA, March 2004.
- [40] R. Puri and K. Ramchandran, “PRISM: A Video Coding Architecture Based on Distributed Compression Principles”, ERL Technical Report, University of California, Berkeley, USA, March 2003.
- [41] A. Aaron, S. Rane, R. Zhang and B. Girod, “Wyner-Ziv Coding for Video: Applications to Compression and Error Resilience”, *Proceedings of IEEE Data Compression Conference*, pp. 93-102, Snowbird, Utah, USA, March 2003.
- [42] A. Aaron, E. Setton and B. Girod, “Towards Practical Wyner-Ziv Coding of Video”, *Proceedings of IEEE International Conference on Image Processing*, Vol. 3, pp. 869-872, Barcelona, Spain, September 2003.
- [43] A. Aaron, S. Rane, D. Rebollo-Monedero and B. Girod, “Systematic Lossy Forward Error Protection for Video Waveforms”, *Proceedings of IEEE International Conference on Image Processing*, Vol. 1, pp. 609-612, Barcelona, Spain, September 2003.
- [44] A. Sehgal, A. Jagmohan, N. Ahuja, “A State-Free Causal Video Encoding Paradigm”, *Proceedings of IEEE International Conference on Image Processing*, Vol. 1, pp. 605-608, Barcelona, Spain, September 2003.
- [45] D. Rowitch and L. Milstein, “On the Performance of Hybrid FEC/ARQ Systems using Rate Compatible Punctured Turbo Codes”, *IEEE Transactions on Communications*, Vol. 48, No. 6, pp. 948–959, June 2000.

**Chapter 3**

- [46] C. Berrou, A. Glavieux and P. Thitimajshima, “Near Shannon Limit Error-Correcting Coding and Decoding: Turbo-Codes (1)”, *IEEE International Conference on Communications*, Vol. 2, pp. 1064-1070, Geneva, Switzerland, May 1993.
- [47] L. C. Perez, J. Seghers, and D. J. Costello, “A Distance Spectrum Interpretation of Turbo Codes”, *IEEE Transactions on Information Theory*, Vol. 42, No. 6, pp. 1698-1709, November 1996.
- [48] C. E. Shannon, “A Mathematical Theory of Communication”, *Bell System Technical Journal*, Vol. 27, pp. 379-423, 623-656, July, October, 1948.
- [49] L. Alparone, M. Barni, F. Bartolini and V. Cappellini, “Adaptively Weighted Vector-Median Filters for Motion-Fields Smoothing”, *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, Vol. 4, pp. 2267-2270, Georgia, USA, May 1996.
- [50] L. R. Bahl, J. Cocke, F. Jeinek and J. Raviv, “Optimal Decoding of Linear Codes for Minimizing Symbol Error Rate”, *IEEE Transactions on Information Theory*, Vol. 20, No. 2, pp. 248-287, March 1974.
- [51] W. E. Ryan, “A Turbo Code Tutorial”, New Mexico State University.
- [52] A. B. Carlson, “Communication Systems”, McGraw-Hill, 2000.
- [53] P. A. Regalia, “Iterative Decoding of Concatenated Codes: A Tutorial”, *EURASIP Journal on Applied Signal Processing*, Vol. 2005, No. 6, pp. 762-774, June 2005.

**Chapter 4**

- [54] R. Clarke, “Transform Coding of Images”, Academic Press, San Diego, USA, 1990.
- [55] K. R. Rao and P. Yip, “Discrete Cosine Transform: Algorithms, Advantages, Applications”, Academic Press, Boston, 1990.
- [56] B. Haskell, A. Puri and A. Netravali, “Digital Video: An Introduction to MPEG-2”, Chapman & Hall, New York, USA, 1997.

**Chapter 5**

- [57] C. Brites, F. Pereira; “Distributed Video Coding: Bringing New Applications to Life”, *5<sup>th</sup> Conference on Telecommunications - ConfTele*, Tomar, Portugal, April 2005.
- [58] J. Ascenso, C. Brites, F. Pereira, “Improving Frame Interpolation with Spatial Motion Smoothing for Pixel Domain Distributed Video Coding”, *5<sup>th</sup> EURASIP Conference on*

*Speech and Image Processing, Multimedia Communications and Services*, Slovak Republic, July 2005.

- [59] J. Ascenso, C. Brites, F. Pereira, “Motion Compensated Refinement for Low Complexity Pixel Based Distributed Video Coding”, *IEEE International Conference on Advanced Video and Signal Based Surveillance*, Como, Italy, September 2005.
- [60] L. Natário, C. Brites, J. Ascenso, F. Pereira, “Extrapolating Side Information for Low-Delay Pixel-Domain Distributed Video Coding”, *International Workshop on Very Low Bitrate Video - VLBV*, Sardinia, Italy, September 2005.
- [61] A. Trapanese, M. Tagliasacchi, S. Tubaro, J. Ascenso, C. Brites, F. Pereira, “Embedding a Block-based Intra Mode in Frame-based Pixel Domain Wyner-Ziv Video Coding”, *International Workshop on Very Low Bitrate Video - VLBV*, Sardinia, Italy, September 2005.
- [62] A. Trapanese, M. Tagliasacchi, S. Tubaro, J. Ascenso, C. Brites, F. Pereira, “Improved Correlation Noise Statistics Modeling in Frame-based Pixel Domain Wyner-Ziv Video Coding”, *International Workshop on Very Low Bitrate Video - VLBV*, Sardinia, Italy, September 2005.
- [63] C. Brites, J. Ascenso, A. Trapanese, M. Tagliasacchi, F. Pereira, S. Tubaro, “Advances on Pixel Domain Wyner-Ziv Video Coding”, *IEEE Transactions on Image Processing* (submitted).

## **Annex A**

- [64] [http://ise.stanford.edu/labsite/ise\\_test\\_images\\_videos.html](http://ise.stanford.edu/labsite/ise_test_images_videos.html).